

1. BellaDati User's Documentation	4
1.1 Getting Started	10
1.1.1 Logging in to BellaDati	11
1.1.2 Exploring BellaDati Workspace	12
1.1.3 Video Tutorials	14
1.1.4 Troubleshooting	15
1.1.4.1 Data Sets do not appear	16
1.1.4.2 Strange symbols instead of icons	17
1.2 BellaDati Concepts	20
1.2.1 Detailed Glossary	21
1.2.2 Permissions and Roles	24
1.2.2.1 Report viewers - Control types	25
1.3 Data Sets	27
1.3.1 Creating Data Set	29
1.3.2 Importing Data	30
1.3.2.1 Importing from Clipboard	31
1.3.2.2 Importing from File	32
1.3.2.2.1 Adding custom columns in XML import	35
1.3.2.3 Data Sources	37
1.3.2.3.1 Connecting to SQL Database	40
1.3.2.3.2 Connecting to Microsoft Analysis Services (SSAS)	49
1.3.2.3.3 Connecting to SAP BW	50
1.3.2.3.4 Connecting to SAP HANA	52
1.3.2.3.5 Connecting to URL	54
1.3.2.3.6 Connecting to MongoDB	57
1.3.2.3.7 Connecting to FTP	59
1.3.2.3.8 Connecting to Google Analytics	61
1.3.2.3.9 Connecting to Google Drive	63
1.3.2.3.10 Connecting to Facebook	65
1.3.2.3.11 Connecting to Twitter	69
1.3.2.3.12 Connecting to LinkedIn	74
1.3.2.3.13 Connecting to Zendesk	76
1.3.2.3.14 Connecting to Salesforce	78
1.3.2.3.15 Connecting to Amiando	84
1.3.2.3.16 Connecting to Intuit	85
1.3.2.3.17 Connecting to YouTube	86
1.3.2.3.18 Connecting to Existing Data Set	87
1.3.2.3.19 Connecting to Hadoop	88
1.3.2.3.20 Connecting to MS SharePoint	89
1.3.2.3.21 Data source overview	92
1.3.2.4 Import Settings	94
1.3.2.5 Import Results	103
1.3.2.6 Scheduling Import	105
1.3.2.6.1 On-demand data synchronization	106
1.3.3 Browsing Data	108
1.3.4 Managing Indicators	109
1.3.4.1 Grouping Indicators	111
1.3.4.2 Translating Indicators	112
1.3.4.3 Transforming Indicator values	113
1.3.4.4 Build indicators with formula	114
1.3.4.5 Adding Permissions to Indicators	115
1.3.5 Managing Attributes	116
1.3.5.1 Defining Drill-down Path	117
1.3.5.2 Members Appearance	119
1.3.5.3 Translating Attributes and Members	120
1.3.5.4 Transforming Attribute Values	122
1.3.5.5 Creating Attribute Subsets	123
1.3.5.6 Adding Permissions to Attributes	125
1.3.5.6.1 Setting Permission Filter - Use Case	127
1.3.5.7 Creating attribute custom members	129
1.3.5.8 Editing default GEO location mapper	130
1.3.6 Joining Data Sets	131
1.3.6.1 Changing join point	134
1.3.6.2 Building joined data set	135
1.3.7 Working with SAP HANA Predictions Manager	136
1.3.8 Structure Backup	138
1.3.9 Sharing Data Sets	141
1.3.10 Watching Data Changes	143
1.3.11 Managing GEO Data	144
1.3.12 Database Connections Library	150
1.4 Reports	151
1.4.1 Creating Report	153

1.4.2 Creating View	154
1.4.2.1 Creating Table	157
1.4.2.1.1 Adding Date Intervals	161
1.4.2.1.2 Editing Table Axes Content	164
1.4.2.1.3 Setting table export configuration	166
1.4.2.2 Creating Chart	169
1.4.2.2.1 Creating Candlestick Chart	173
1.4.2.2.2 Creating Gantt Chart	174
1.4.2.2.3 Creating XY chart	175
1.4.2.2.4 Editing Chart Axes Content	178
1.4.2.2.5 Managing chart appearance	180
1.4.2.3 Creating Geo Maps	182
1.4.2.4 Creating KPI labels	188
1.4.2.4.1 Setting appearance of KPI Label	190
1.4.2.5 Adding Custom Content	192
1.4.2.6 Displaying Indicators	194
1.4.2.7 Using Core Statistical Functions	197
1.4.2.8 Exporting View	198
1.4.2.9 Reversing Changes (Undo & Redo)	200
1.4.2.10 Creating Filters	201
1.4.2.11 Conditional formatting	203
1.4.3 Sharing Report	207
1.4.4 Exporting Report	210
1.4.4.1 Export schedule	213
1.4.4.1.1 Adjusting size of the views in the email body	217
1.4.5 Copying Report	219
1.4.6 Managing Layout	220
1.4.7 Saving Reports	221
1.4.8 Using Formulas	222
1.4.8.1 Formula Reference Guide	224
1.4.8.1.1 Datetime Functions	225
1.4.8.1.2 Datetime Constants	229
1.4.8.1.3 Math Functions	231
1.4.8.1.4 Special Functions	233
1.4.8.1.5 Accessing Report Variables	238
1.4.8.1.6 Referencing Data From Another Data Set	239
1.4.8.2 Formula Use Cases	240
1.4.8.2.1 Filtering in Formulas	241
1.4.8.2.2 Calculating Frequencies	243
1.4.8.2.3 Calculating Percentual Share in Drill-Downs	245
1.4.8.2.4 Calculating With Members On Defined Level	246
1.4.8.2.5 Calculating Average Cumulated Values	247
1.4.8.2.6 Calculating Percentiles and Quantils	248
1.4.8.2.7 Getting Last Available Value	250
1.4.8.2.8 Cross-referencing Values from Different Data Set	251
1.4.8.2.9 Handling empty (NULL) values by formulas	252
1.4.8.2.10 Representing numbers in accounting format	253
1.4.8.2.11 Overriding Date Interval with Day Order	254
1.4.8.2.12 Displaying text values in KPI labels	255
1.4.8.2.13 Display Top or Bottom Member Value in KPIs	256
1.4.8.2.14 Calculating average across different drill down levels	257
1.4.8.2.15 Calculate revenue using unit price times quantity and revenue percentage	259
1.4.8.2.16 Calculating with NULL values	260
1.4.8.2.17 Link Relative Ratio and Year-on-year Comparison	261
1.4.9 Setting Date Interval	262
1.4.10 Using Filters	264
1.4.10.1 Using Report Filters	267
1.4.10.2 Using Indicator Filters	268
1.4.11 Report Variables and Filters components	269
1.4.11.1 Custom Filter Components	274
1.4.11.2 Passing Parameters in URL	277
1.4.12 Defining Drill-through URL	281
1.4.13 Adding Comments and Attachments	283
1.4.14 Publishing to Dashboard	285
1.4.15 Searching and Filtering Reports	286
1.4.16 Translating Reports	288
1.4.17 Displaying Source Data	289
1.4.18 Auto update interval	290
1.4.19 Associated views	291
1.4.20 Report appearance	293
1.4.21 Report settings	294
1.4.22 Report navigation	295

1.5 Dashboards	296
1.5.1 Creating Dashboard	298
1.5.2 Managing Dashboard Layout	300
1.5.3 Creating Dashlet	302
1.5.3.1 Adding View	305
1.5.3.2 Adding Navigation	306
1.5.3.3 Adding Filter	307
1.5.3.4 Adding Other Content	308
1.5.4 Sharing Dashboard	309
1.5.5 Adding Attachment	311
1.5.6 Exporting Dashboard	312
1.5.7 Copying dashboard	314
1.5.8 Translating dashboards	315
1.6 Folders	316
1.7 BellaDati Mobile	318
1.7.1 BellaDati Mobile for iOS	319
1.7.2 BellaDati Mobile for Android	326
1.8 BellaApps	334
1.8.1 Creating App	335
1.8.2 Uploading App	337
1.9 Media Gallery	339
1.10 Data Collection Module	341
1.11 Administration	344
1.11.1 Administering Users	345
1.11.2 Importing Users	346
1.11.3 Administering User Groups	347
1.11.4 Sharing console	348
1.11.5 Managing User Profile	350
1.11.6 Managing Configuration	352
1.11.6.1 Setup Active Directory Authentication	354
1.11.6.2 Setup Google login	355
1.11.6.3 Email server configuration	357
1.11.6.4 Twitter Authentication Setup	358
1.11.7 Domain Configuration and Administration	360
1.11.7.1 Color schemes	361
1.11.7.2 Domain Banners (Cover Pictures)	362
1.11.7.3 External Custom Login Page	363
1.11.8 Domain Backup	364
1.11.9 Usage Monitoring	366
1.11.10 Administering domains	367
1.11.11 Mass mailing	368
1.11.12 Report transferring	369

BellaDati User's Documentation

Welcome

The BellaDati User's Guide begins with an overview of the key features of BellaDati, and goes on to describe the use of BellaDati as a smart reporting, analytical and a collaborative tool. If you still have a question that has not been answered, [please ask our support team](#).



You can download the [BellaDati documentation in PDF and HTML](#) formats.

Getting Started

- Logging in to BellaDati
- Exploring BellaDati Workspace
- Video Tutorials
- Troubleshooting
 - Data Sets do not appear
 - Strange symbols instead of icons

BellaDati Concepts

- Detailed Glossary
- Permissions and Roles
 - Report viewers - Control types

Data Sets

- Creating Data Set
- Importing Data
 - Importing from Clipboard
 - Importing from File
 - Adding custom columns in XML import
 - Data Sources
 - Connecting to SQL Database
 - Conditional SQL import
 - Connection parameters examples
 - Upgrading JDBC drivers
 - Using variables in SQL query
 - Connecting to Microsoft Analysis Services (SSAS)
 - Connecting to SAP BW
 - Connecting to SAP HANA
 - Connecting to URL
 - Connecting to MongoDB
 - Connecting to FTP
 - Connecting to Google Analytics
 - Connecting to Google Drive
 - Connecting to Facebook
 - Configuring Facebook connector
 - Connecting to Twitter
 - Configuring Twitter connector
 - Connecting to LinkedIn
 - Connecting to Zendesk
 - Connecting to Salesforce
 - Configuring Salesforce connector
 - Connecting to Amiando
 - Connecting to Intuit
 - Connecting to YouTube
 - Connecting to Existing Data Set
 - Connecting to Hadoop
 - Connecting to MS SharePoint
 - Data source overview
 - Import Settings
 - Import Results
 - Scheduling Import
 - On-demand data synchronization
- Browsing Data

- Managing Indicators
 - Grouping Indicators
 - Translating Indicators
 - Transforming Indicator values
 - Build indicators with formula
 - Adding Permissions to Indicators
- Managing Attributes
 - Defining Drill-down Path
 - Members Appearance
 - Translating Attributes and Members
 - Transforming Attribute Values
 - Creating Attribute Subsets
 - Adding Permissions to Attributes
 - Setting Permission Filter - Use Case
 - Creating attribute custom members
 - Editing default GEO location mapper
- Joining Data Sets
 - Changing join point
 - Building joined data set
- Working with SAP HANA Predictions Manager
- Structure Backup
- Sharing Data Sets
- Watching Data Changes
- Managing GEO Data
- Database Connections Library

Reports

- Creating Report
- Creating View
 - Creating Table
 - Adding Date Intervals
 - Working with Date Intervals
 - Editing Table Axes Content
 - Setting table export configuration
 - Creating Chart
 - Creating Candlestick Chart
 - Creating Gantt Chart
 - Creating XY chart
 - Editing Chart Axes Content
 - Managing chart appearance
 - Creating Geo Maps
 - Creating KPI labels
 - Setting appearance of KPI Label
 - Adding Custom Content
 - Displaying Indicators
 - Using Core Statistical Functions
 - Exporting View
 - Reversing Changes (Undo & Redo)
 - Creating Filters
 - Conditional formatting
- Sharing Report
- Exporting Report
 - Export schedule
 - Adjusting size of the views in the email body
- Copying Report
- Managing Layout
- Saving Reports
- Using Formulas
 - Formula Reference Guide
 - Datetime Functions
 - Datetime Constants
 - Math Functions
 - Special Functions
 - Accessing Report Variables
 - Referencing Data From Another Data Set
 - Formula Use Cases
 - Filtering in Formulas
 - Calculating Frequencies
 - Calculating Percentual Share in Drill-Downs
 - Calculating With Members On Defined Level

- Calculating Average Cumulated Values
- Calculating Percentiles and Quantils
- Getting Last Available Value
- Cross-referencing Values from Different Data Set
- Handling empty (NULL) values by formulas
- Representing numbers in accounting format
- Overriding Date Interval with Day Order
- Displaying text values in KPI labels
- Display Top or Bottom Member Value in KPIs
- Calculating average across different drill down levels
- Calculate revenue using unit price times quantity and revenue percentage
- Calculating with NULL values
- Link Relative Ratio and Year-on-year Comparison
- Setting Date Interval
- Using Filters
 - Using Report Filters
 - Using Indicator Filters
- Report Variables and Filters components
 - Custom Filter Components
 - Passing Parameters in URL
- Defining Drill-through URL
- Adding Comments and Attachments
- Publishing to Dashboard
- Searching and Filtering Reports
- Translating Reports
- Displaying Source Data
- Auto update interval
- Associated views
- Report appearance
- Report settings
- Report navigation

Dashboards

- Creating Dashboard
- Managing Dashboard Layout
- Creating Dashlet
 - Adding View
 - Adding Navigation
 - Adding Filter
 - Adding Other Content
- Sharing Dashboard
- Adding Attachment
- Exporting Dashboard
- Copying dashboard
- Translating dashboards

Folders

BellaDati Mobile

- BellaDati Mobile for iOS
- BellaDati Mobile for Android

BellaApps

- Creating App
- Uploading App

Media Gallery

Data Collection Module

Administration

- Administering Users
- Importing Users
- Administering User Groups
- Sharing console
- Managing User Profile
- Managing Configuration
 - Setup Active Directory Authentication
 - Setup Google login
 - Email server configuration
 - Twitter Authentication Setup
- Domain Configuration and Administration
 - Color schemes
 - Domain Banners (Cover Pictures)
 - External Custom Login Page
- Domain Backup
- Usage Monitoring
- Administering domains
- Mass mailing
- Report transferring

Release notes

- BellaDati 2.7.14.1.1
- BellaDati 2.7.14.1
- BellaDati 2.7.14
- BellaDati 2.7.13.6
- BellaDati 2.7.13.5.1
- BellaDati 2.7.13.5
- BellaDati 2.7.13.4.1
- BellaDati 2.7.13.4
- BellaDati 2.7.13.3
- BellaDati 2.7.13.2
- BellaDati 2.7.13.1.1
- BellaDati 2.7.13.1
- BellaDati 2.7.13
- BellaDati 2.7.12.4
- BellaDati 2.7.12.3
- BellaDati 2.7.12.2
- BellaDati 2.7.12.1.1
- BellaDati 2.7.12.1
- BellaDati 2.7.12
- BellaDati 2.7.11.10.2
- BellaDati 2.7.11.10.1
- BellaDati 2.7.11.10
- BellaDati 2.7.11.9
- BellaDati 2.7.11.8
- BellaDati 2.7.11.7
- BellaDati 2.7.11.6.1
- BellaDati 2.7.11.6
- BellaDati 2.7.11.5.2
- BellaDati 2.7.11.5.1
- BellaDati 2.7.11.5
- BellaDati 2.7.11.4
- BellaDati 2.7.11.3
- BellaDati 2.7.11.2.3
- BellaDati 2.7.11.2.2
- BellaDati 2.7.11.2.1
- BellaDati 2.7.11.2
- BellaDati 2.7.11.1
- BellaDati 2.7.11.0.4
- BellaDati 2.7.11.0.3
- BellaDati 2.7.11.0.2
- BellaDati 2.7.11.0.1
- BellaDati 2.7.11
- BellaDati 2.7.10.0.7
- BellaDati 2.7.10.0.5
- BellaDati 2.7.10.0.4
- BellaDati 2.7.10.0.3

- [BellaDati 2.7.10.0.2](#)
- [BellaDati 2.7.10.0.1](#)
- [BellaDati 2.7.10](#)
- [BellaDati 2.7.9.5.5](#)
- [BellaDati 2.7.9.5.4](#)
- [BellaDati 2.7.9.5.3](#)
- [BellaDati 2.7.9.5.2](#)
- [BellaDati 2.7.9.5.1.1](#)
- [BellaDati 2.7.9.5.1](#)
- [BellaDati 2.7.9.5](#)
- [BellaDati 2.7.9.4.1](#)
- [BellaDati 2.7.9.4](#)
- [BellaDati 2.7.9.3](#)
- [BellaDati 2.7.9.2](#)
- [BellaDati 2.7.9.1.5](#)
- [BellaDati 2.7.9.1.4](#)
- [BellaDati 2.7.9.1.3](#)
- [BellaDati 2.7.9.1.2](#)
- [BellaDati 2.7.9.1.1](#)
- [BellaDati 2.7.9.1](#)
- [BellaDati 2.7.9.0.1](#)
- [BellaDati 2.7.9](#)
- [BellaDati 2.7.8.1.2](#)
- [BellaDati 2.7.8.1.1](#)
- [BellaDati 2.7.8.1](#)
- [BellaDati 2.7.8](#)
- [BellaDati 2.7.7.4](#)
- [BellaDati 2.7.7.3](#)
- [BellaDati 2.7.7.2](#)
- [BellaDati 2.7.7.1](#)
- [BellaDati 2.7.7](#)
- [BellaDati 2.7.6](#)
- [BellaDati 2.7.5.6](#)
- [BellaDati 2.7.5.5](#)
- [BellaDati 2.7.5.4](#)
- [BellaDati 2.7.5.3](#)
- [BellaDati 2.7.5.2](#)
- [BellaDati 2.7.5.1](#)
- [BellaDati 2.7.5.0.1](#)
- [BellaDati 2.7.5](#)
- [BellaDati 2.7.4.5](#)
- [BellaDati 2.7.4.4](#)
- [BellaDati 2.7.4.3](#)
- [BellaDati 2.7.4.2.1](#)
- [BellaDati 2.7.4.2](#)
- [BellaDati 2.7.4.1](#)
- [BellaDati 2.7.4.0.1](#)
- [BellaDati 2.7.4](#)
- [BellaDati 2.7.3.4](#)
- [BellaDati 2.7.3.3](#)
- [BellaDati 2.7.3.2](#)
- [BellaDati 2.7.3.1](#)
- [BellaDati 2.7.3](#)
- [BellaDati 2.7.2.2](#)
- [BellaDati 2.7.2.1](#)
- [BellaDati 2.7.2](#)
- [BellaDati 2.7.1.1](#)
- [BellaDati 2.7.1](#)
- [BellaDati 2.7.0.1](#)
- [BellaDati 2.7](#)
- [BellaDati 2.6.3.2](#)
- [BellaDati 2.6.3.1](#)
- [BellaDati 2.6.3](#)
- [BellaDati 2.6.2.1](#)
- [BellaDati 2.6.2](#)
- [BellaDati 2.6.1.3](#)
- [BellaDati 2.6.1.2](#)
- [BellaDati 2.6.1.1](#)
- [BellaDati 2.6.1](#)
- [BellaDati 2.6.0.3](#)
- [BellaDati 2.6.0.2](#)
- [BellaDati 2.6.0.1](#)

- [BellaDati 2.6](#)
- [BellaDati 2.5.2](#)
- [BellaDati 2.5.1](#)
- [BellaDati 2.5](#)

SvNavigation

Getting Started

The following pages contain information to help you get started using BellaDati:

- [Logging in to BellaDati](#)
- [Exploring BellaDati Workspace](#)
- [Video Tutorials](#)
- [Troubleshooting](#)

Logging in to BellaDati



Anonymous access to BellaDati is not permitted and you are always prompted to log in.

The **Login panel** will be displayed if you are not logged in to BellaDati. You can do following:

1. **Login:** To log in to BellaDati, enter your **Username** and **Password** and click the **Log In** button. You can also use following services to log in:
 - Log in using your **Google** account
 - Log in using your **Facebook** account
 - Log in using your **Twitter** account
 - Log in using your **LinkedIn** account
1. **Lost Password:** Click this link in case you have lost your password or your BellaDati account has been locked (see login policy below). Select the appropriate option and enter your username (usually e-mail address) in the popup. Then click "Submit" button and now you should receive the instructions how to set your new password or unlock your account via e-mail in a few minutes. If you have forgotten your username, you will need to contact your BellaDati administrator for help.
2. **Create a New Account:** If you do not have a user account yet, you can create your own user account by clicking on the link and follow registering procedure.



Security note: When you are using BellaDati Cloud, you can verify BellaDati site identity by clicking on thawte certificate logo here to prevent fraud of your password. Data transfer security is then ensured by HTTPS encryption.

Login and Password Policy

- Password must be at least **8 characters long** and must contain at least **one uppercase [A-Z]**, **one lowercase [a-z]** and **one numeric character [0-9]**.
- Language specific characters (like diacritic) are not supported.
- When the login fails three times, your BellaDati account become locked. An e-mail with verification link will be sent to your registered e-mail. Use this link to unlock your account.



Unlocking link will be valid for 24 hours only! Otherwise you need to create new request on BellaDati login page.



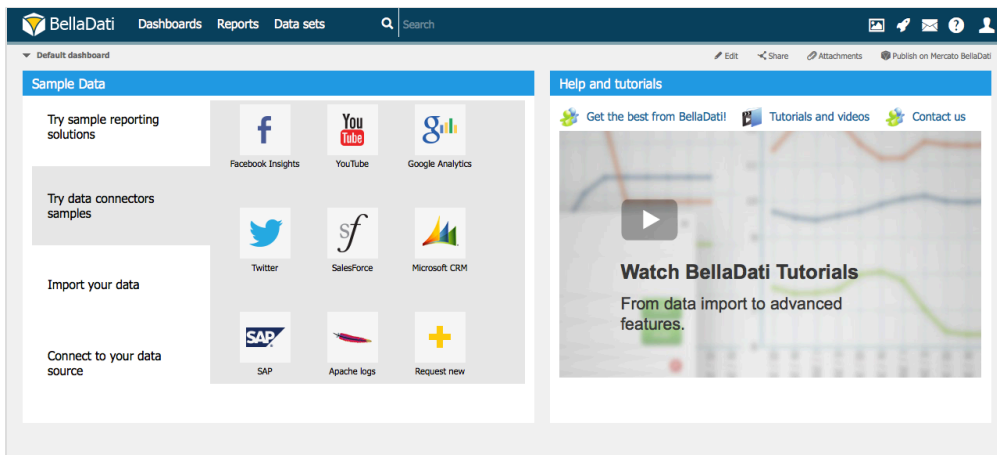
BellaDati On-Premise notes:


1. Maximum failed login attempts count can be set differently by domain administrator.
2. Authentication can be performed using LDAP or Active Directory services.

Exploring BellaDati Workspace

The Dashboard is the first screen you see when you login to BellaDati.

- **Menu** (navigation bar) is the same on every screen in BellaDati. It contains links which give you quick access to many of BellaDati's most useful functions. Menu structure depends on the roles assigned to you by administrator.
- **Main window** content changes according to the context of your current work in BellaDati. Dashboard is the default page. An additional submenu is usually displayed on the left or top of the main window to allow executing context specific functions. This will be displayed in corresponding documentation sections.
- **Footer** provides supplemental function and is also the same on every screen in BellaDati.



 Please note that your BellaDati screen may look slightly different from this screenshot when you are using BellaDati On-Premise. The default content of BellaDati page after login may also differ if you specify particular URL before login.

More detailed description of navigation in BellaDati is following:

Main menu

Following functions are available for all BellaDati users if not stated otherwise:

- **BellaDati icon** - always navigates you to default BellaDati page (Dashboards)
- **Dashboards** - navigates to [dashboards page](#)
- **Reports** - navigates to [reports page](#)
- **Data sets** - navigates to [data sets page](#) and [data source configuration](#); *data manager and domain administrator only*
- **Users** - navigates to [user](#) and [user groups administration](#); *domain and system administrator only*
- **Domains** - navigates to domain management; *system administrator only*
- **Search field** - allows you to search among [reports](#), [data sets](#), [indicators](#) and [attributes](#)
- **Media Gallery** - allows you to manage your [media and images](#)
- **BellaApps** - allows you to import and export [BellaDati apps](#)
- **Contact Us** - get in touch with BellaDati support
- **Help** - navigates you to context help (help opens in separate browser's window)
- **YourName, Logout** - this link navigates you to your profile details, password change and safe logout from BellaDati; domain and system administrators can assign user roles or user groups through this



System administrator role is not required for BellaDati Cloud usage. However this role could be useful for BellaDati On-Premise, especially for large enterprise companies.

Footer

BellaDati footer displays:

- Installed BellaDati version
- Link to **support** (this documentation, bug reporting, video tutorials, blog, etc.)
- Privacy policy and data security conditions
- Terms of use (or EULA for BellaDati On-Premise)

Video Tutorials

Troubleshooting



This section describes steps, which should be followed, when situation like this appears.

- [Strange symbols instead of icons](#)
- [Data Sets do not appear](#)

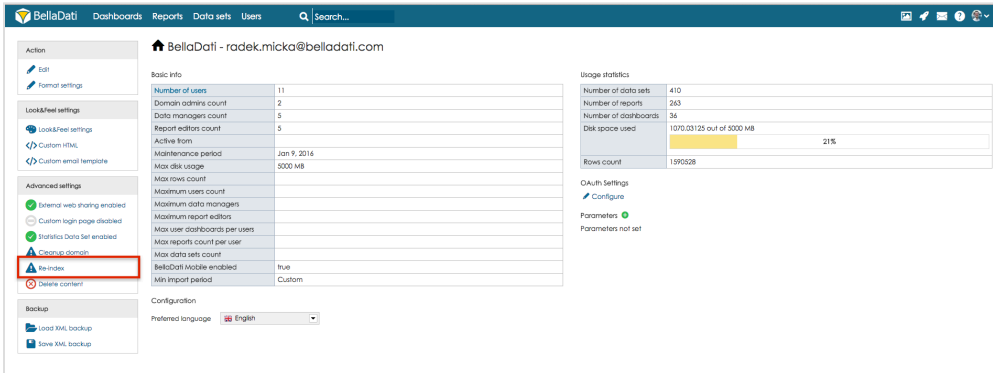
Data Sets do not appear


Description

Previously created data sets do not appear.

Solution

1. Go to domain settings
2. Run the **Re-index** service



 Domain settings page can be accessible only by Domain administrator.

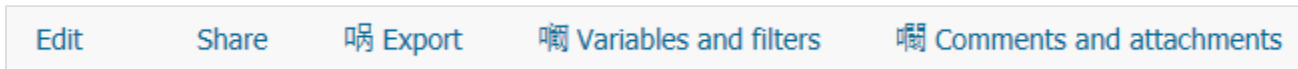
If these steps do not work, use following workaround:

1. **Stop** BellaDati service
2. Delete the folder C:\Program Files\BellaDati\indexes
3. **Start** BellaDati service
4. Run the **Re-index** service

Strange symbols instead of icons

Description

Instead of icons, I'm seeing strange symbols in Internet Explorer.

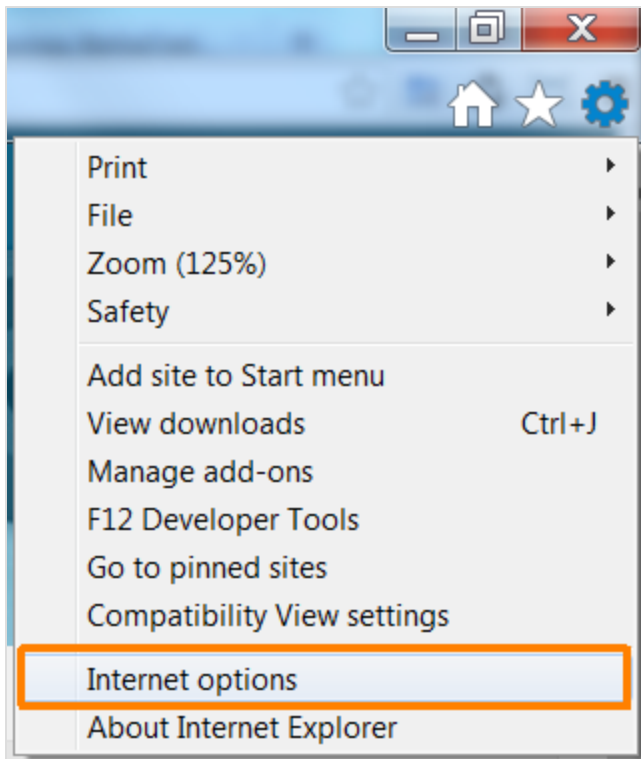


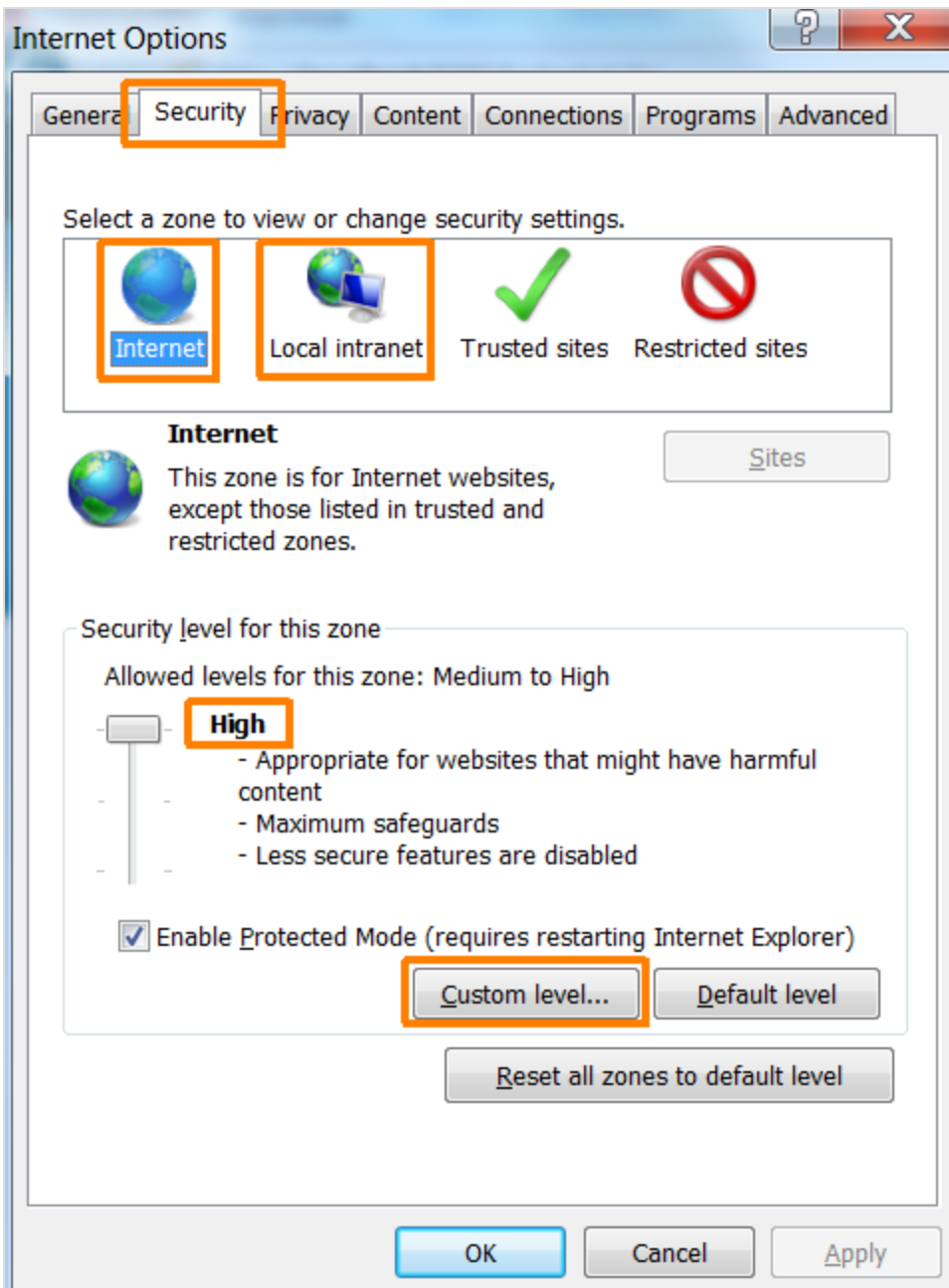
The actual symbols displayed on your computer may be different from the ones shown in the screenshot.

Cause and Solution

BellaDati is using a custom font to display its icons. High security settings in Internet Explorer may disable the use of custom fonts.

To enable the icons, go to **Internet Options** and select the **Security** tab.

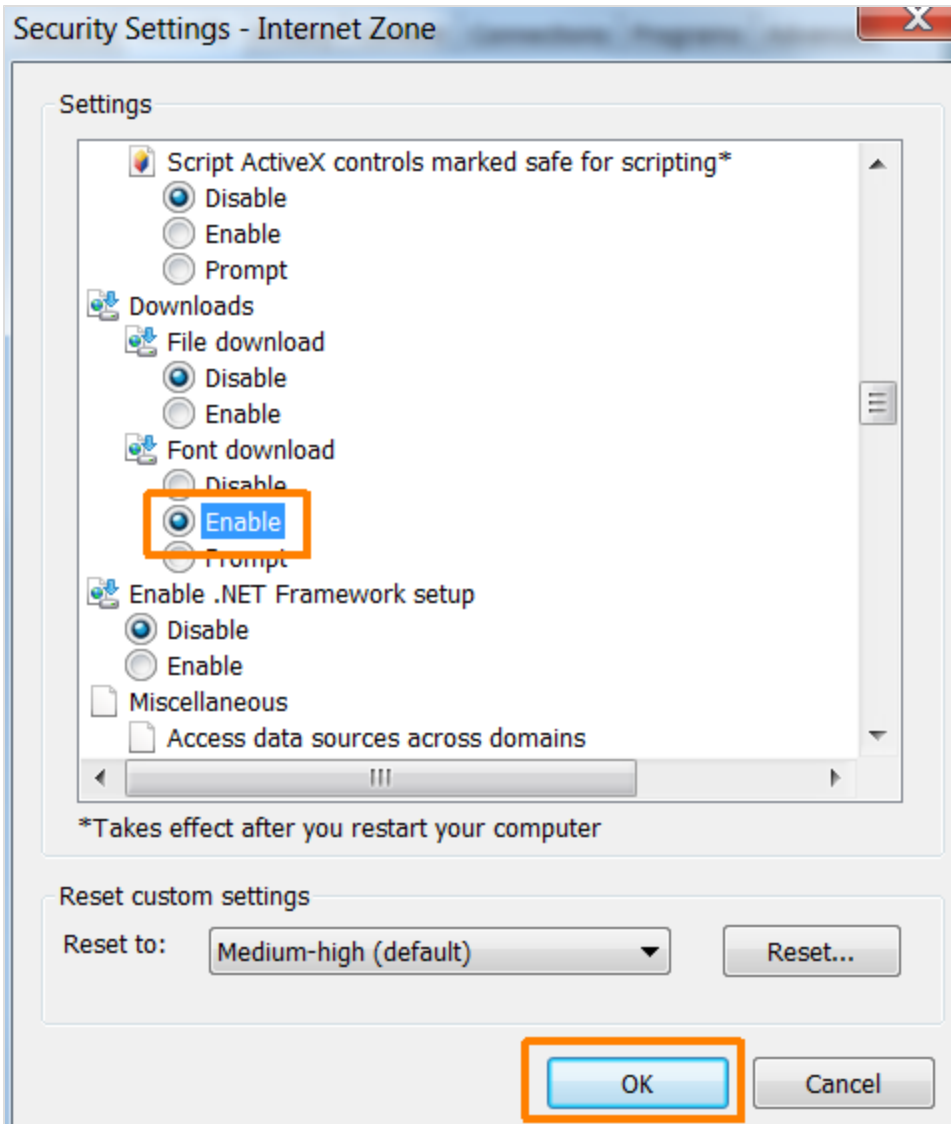




Select **Internet** or **Local intranet** depending on where your BellaDati server is located. If you're unsure, ask your systems administrator or simply apply the following steps for both.

The zone you're modifying should have a security level of either **High** or **Custom**.

Click the **Custom Level** button.



Scroll down until you see **Downloads** and **Font download**. Select **Enable**.

Close all dialogs with **Ok** and press **Ctrl+R** to reload the page. Icons are now displayed.

[Edit](#) [Share](#) [Export](#) [Variables and filters](#) [Comments and attachments](#)

BellaDati Concepts

Key concepts used in BellaDati are described below. See [BellaDati glossary](#) for details.

Data set

BellaDati has its own integrated data warehouse. This warehouse contains virtual databases that represents data with similar characteristics. These virtual databases are called data sets and reports are build on them. Each data set can be connected to multiple [data source](#).

Possible actions: [Importing data](#) | [Browsing data](#) | [Exporting data](#) | [Structure backup](#) | [Joining data sets](#) | [Watching data changes](#) | [Cleaning data](#) | [Sharing data](#) | [Defining attributes](#) | [Defining indicators](#) | [Creating and removing datasets](#)

Data source

Imagine companies has already running their business. Such company has applications and systems installed or use services that generate a lot of data in underlying databases. Moreover employess produces Excel spreadsheets, text or CSV files of data. These databases and files are very suitable data sources for BellaDati analysis. Generally all third party systems BellaDati can import data from are called data sources.

Possible actions: [Connecting to database](#) | [Connecting to URL](#) | [Connecting to FaceBook](#) | [Connecting to Google Services](#) | [Connecting to CRM](#) | [Scheduling automatic upload](#)

Report

Report is basically a set of tables and charts created mainly for detailed overview and analytical purposes on data stored in data sets. Each report can also contain custom content, comments, attachments and can be easily shared with other BellaDati users or published to corporate intranet or public. Permanent export of each report to PDF, Excel, PNG or Power Point is possible.

Possible actions: [Creating report](#) | [Managing Layout](#) | [Creating view](#) | [Creating table](#) | [Creating chart](#) | [Creating geo map](#) | [Creating KPI label](#) | [Copying report](#) | [Sharing report](#) | [Exporting report](#) | [Adding comments](#) | [Publishing report](#) | [Filtering data](#) | [Using Formulas](#) | [Searching and Filtering Reports](#)


Dashboard

Dashboard typically consists of the most important tables or charts from reports and is primarily determined for managers who need quick overview of actual company situation. Each dashboard can be customized by adding special content (dashlets). Dashboards can also be shared for public via web.

Possible actions: [Creating dashboard](#) | [Managing Layout](#) | [Creating Dashlet](#) | [Sharing Dashboard](#) | [Adding Attachment](#)

Detailed Glossary



Base concepts

Attribute	<p>Attributes are describing indicators, usually in a form of general terms such as "country", "department", "product", "employee", etc. Attributes consist of members (attribute values, instances) eg. "Czech Republic", "Sales Department", "Product XY", "John Smith".</p> <p><i>For example, a Sales record could have attributes such as sales person, store, region, date, etc.</i></p>
Dashboard	<p>Dashboard typically consists of the most important tables or charts from reports and is primarily determined for managers who need quick overview of actual company situation. Special custom content can be added to every dashboard as a customization. Dashboards can also be shared for public via web.</p>
Data set	<p>A virtual database of BellaDati's integrated data warehouse. Each data set usually represents data with similar characteristics (e.g. invoices, wages, costs) from one data source. Data set consists of data set dates, indicators, attributes and attribute translations. Each report is build on single data set. More data sets can also be joined together.</p>
Data set indicator	<p>Data set indicators are defined within the data set and are available as musters for report indicators. Another settings (like aggregations, appearance, etc.) are not supported. Simply said, the data set indicators represents a raw numerical value (in the OLAP language it is a <i>fact</i>) with basic attributes - name, unit and rounding mode. Values of these indicators are straightforwardly stored in BellaDati's data warehouse directly from imported data.</p>
Member / Attribute value	<p>"Instance" of one attribute. For example, members of the "employee" attribute may be "Jan Novak", "John Smith" etc.</p> 
Record	<p>Particular data row stored in the data set. Each record consists of indicator values, attribute values and date/time information and has the same structure within single data set</p>
Report	<p>An analytic output, providing answers to user's research according to the gathered data. Each report can contain description, tags and consists of one or more views (charts and pivot tables), comments and possibly attachments.</p>
Report indicator	<p>Report indicators are created in the report from the data set indicators or ad hoc. Unlike the data set indicators, report indicators are supporting wide range of various settings - member aggregation, time aggregation, appearance, conditional formatting, extended formula support with report variables etc.</p>

Detailed glossary

Term	Meaning
Alarm (data watch)	<p>Notifies data manager about changes in indicator's values in a particular data set according fulfillment of one or more predefined conditions.</p>
Chart	<p>Graphic representation of analytic data view. There are a lot of chart types in BellaDati - eg. line chart, pie chart, bar chart, stack bar chart, horizontal bar chart, radar chart, horizontal heat map. Each of them offers a different way how to explore particular dimensions (time, indicators and attributes).</p>
Comments	<p>A short message or explaining information attached to report or table cells.</p>
Conditional formatting	<p>Conditional formatting allows user to adjust appearance of the particular indicators in table according to currently displayed values or their changes.</p>
Copy	<p>A clone of the existing report, table or chart. Any future change of the copy does not affect the original report.</p>

Dashlet	<p>A basic item of the dashboard. Dashlets usually represent your existing views (tables, charts) from reports. Another types of dashlets have informational, supplemental or customizing function.</p>
----------------	---

Data source	Data source is typically a database, Microsoft Excel spreadsheet, text file or another services eg. Google Doc spreadsheet, Facebook or enterprise services like CRMs (SalesForce, Amiamo etc.).
Date Units	Date Units are predefined time aggregations including Day of Week, Day of Month, Day of Year, Week of Year, Month Of Year, Quarter of Year or Seconds in Minute and Minutes in Hour.
Domain	Whole virtual space of the one registered organization. Users of one domain can't access data in another domains. Domains are completely independent. Separate domains can also be suitable for individual divisions or SBUs of large international companies.
Drill-down	<p>An operation which results in displaying more detailed data with higher granularity according to the chosen drill-down path. If you drill-down a member, you will see its child members. Eg. drill-down of particular affiliate can display its employees.</p> 
Drill-down path	<p>A sequence of attributes which influences the results of the drill-down operation. It can be defined in the data set or chosen ad-hoc in the report.</p> 
Fact	Facts are equal to the data set indicators.
Filters	Filters restricts data displayed in views according to selected members only or general member match pattern. Eg. you can set filter to show only the largest cities in a table or chart. General filter settings are also available when sharing data sets.
Formula	Formula is a tool allowing user to define its own new indicators. Despite using common mathematical functions, advanced functions such as regressions or getting different values of existing indicators in time or depending on their aggregation is possible.
Geo data	Geo data are pairs of location identification and its coordinates. In BellaDati, location can be represented as point or region . While point is identified by single pair of longitude and latitude coordinates, region comprises of multiple points creating the polygonal area.
Geo point	Geo point is one of the attribute types. It holds information about latitude and longitude coordinates and can be used in Geo Map view type to plot data into its particular location. Alternative to Geo points are separately maintained Geo Data .
Group of indicators	Group of indicator consists of one or more included indicators which have the same context (eg. wage components). Users can effectively work with the whole group like with a single indicator.
iFrame	iFrame is a feature, which allows users to embed their analytic views into the external website using HTML code.
Joined data set	An abstract data set, which represents data from two or more source data sets. This allows user to analyze mutual dependencies of data from different data sources or can easily assign member IDs to their names (codebook). Data sets can be connected together according specified condition - inner, outer or cross join. Joined data set reflects all the changes in the source data sets.
Member aggregation	Specifies the way, how the repeating data records of one member are aggregated (in one time unit). The aggregation possibilities are: sum, average, minimum, maximum and count (of different members).
Metric	Metrics are equal to the report indicators.
News	Dashlet, which shows actual and former changes mainly in reports and data sets within the domain. This dahlet also displays information about the author of those changes.
Pivot table	An analytic view in a form of flexible rows and columns. Users can adjust, what should be displayed on each axis (various attributes, indicators or time units) and choose the structure of the drill-down path for the current table. It is also possible to analyze indicators in pivot table through the conditional formatting.
Predefined import	Predefined import is a tool, which allows users to easily load columns settings from the first import during repeated imports.
Query Scheduler	A tool, which allows to schedule and control regular data synchronization. It's main purpose is to update data warehouse with actual data from data sources.

Roll-up	An operation, which displays less detailed data with lower granularity according to the chosen drill-down path. It's an inverse operation to drill-down.
Sample data	A set of predefined content, consisting of reports, dashboards, data sets and predefined imports.
Sharing	Sharing is a feature, which allows collaboration and cooperation among users. It's possible to share your data sets, reports or dashboards with other users in the same domain. You can also choose to delegate permission to edit your data sets and reports. Besides iFrame and iGoogle are another sharing tools, which allows you to share your views on corporate intranet or publicly via Internet.
Subset	Subset is a virtual copy of Attribute which holds only desired members. Subsets members can have custom order.
Time aggregation	Specifies the way, how time units with higher granularity are aggregated in the units with lower granularity. If you (for example) gather data in days but display them in months, the time aggregation determines if the "month" units show sum, average, maximum, minimum or count of the included day records.
Time Series	Time Series in another way of date aggregation . In contrast to Date Units , it aggregates values but displays them on continuous time axis.
Translation	Is one of the Data Set column type. It holds language translation of Attribute members (values).
User group	A set of individual users. The Domain administrator has a right to create, delete or edit these groups. He also adds new users to groups, remove users from groups and assigns user roles to the groups. User group is a very useful tool, because users can share their reports or data sets with the whole group instead of choosing of all its individual members.
User roles	User roles are designed to separate access to different functions for different types of users. Available user roles are "Domain administrator", "Report editor" and "Data manager". These roles can be combined without any restrictions. Only the Domain administrator has a right to delegate those roles to other users.
View	Basic element of the report - displays values of the indicators depending on user's criteria. Views represent flexible points of view to the user's data in a form of pivot table, chart or possibly custom content. Views allow you to use the "slice and dice" features.

Permissions and Roles

BellaDati distinguishes between two basic permission schemes:

- Assigning user roles
- Permissions assigned by sharing

User roles

BellaDati implements following user roles:

- **General BellaDati user** - this role is assigned by default to all BellaDati users and this cannot be changed. Such users have only the very basic access to BellaDati functions: report and dashboard view, user profile editing

This role is usually sufficient for **report** or **dashboard** consumers such as general managers or company management members.

He can't create his own reports, but other users (report editors) can share their reports with him - even with permission to edit their reports. For users who can only view the report can be also different control types. See [the list](#) of allowed operations for selected control type.

- **Report editor** - report editor is able to create, edit, comment and share analytical reports. He can create reports only from his own or shared data sets. Therefore it's usual that users with "report editor role" have also the "data manager" role.
- **Data manager** - data manager cares about the parts of the data warehouse. His job is to prepare and import data into particular data sets, control and edit the source data, create alarms and join existing data sets. He is also able to prepare translation for particular indicators, attributes and members. He is owner of data sets, which he creates during the imports. He can share data sets with other users (report editors) or directly use them if he has also "report editor" role.
- **Domain administrator** - domain administrator a specific and important user role. He cares about the users and user groups. He is able to create or import users, delete them, change their profile information and passwords and assign user roles to particular users or whole user groups (he may assign "domain administrator" role to other users as well) . He has available all the statistics of his domain. He can even delete all the content of domain (data sets, reports, dashboards) or forbid the publishing of domain content on the web. He is the only one user who sees all the data sets in his domain. Therefore this user role should be assigned to only one competent user.

It's possible to combine all user roles mentioned above. Eg. both roles data manager and report editor assigned at the same time allows such user performing the whole process from setting up a [data source](#), modelling data set, report creation and sharing dashboards.



System administrator: There is also a System administrator role in BellaDati. This user role is not required for BellaDati Cloud usage. However this role could be useful for BellaDati On-Premise or Unlimited Cloud tariffs, especially for large enterprise companies or international business groups that require managing more separated domains (eg. for their SBU).



User roles can be assigned to user groups as well. These roles are merged with standard user roles results - particular user has both roles together. Here is an example:

- user has report editor role
- user is member of a user group, which have the data editor role
- in result, user has report and data editor roles, the second one is inherited from the user group.



Only domain or system administrators can reassign user roles.

Permissions

Permissions are granted to users while sharing data sets or reports. There two levels of shared permissions:

- Read-only access
- Full access

Owner: Each data set, report or dashboard has always assigned one user that has full access and also can manage sharing in addition to that. These user are called owners and usually are the creators of the data set, report or dashboard.



Permissions assigned by sharing particular data sets or reports have **priority** over standard user roles. This means user with only general user role assigned can have permission to edit particular data set or report which has been shared with him on full access level!

Report viewers - Control types

BellaDati offers 3 different control types for user permissions of report viewers:

	All controls	Limited controls	Minimal controls	No controls
Edit view settings (for current session only)	✓	✗	✗	✗
Create new formula indicators	✓	✗	✗	✗
Add existing indicators to the views	✓	✓	✓	✗
Modify existing indicators	✓	✗	✗	✗
Create report filters and variables	✗	✗	✗	✗
Create filters of the views	✓	✓	✗	✗
Change filters of the views	✓	✓	✗	✗
Change date/time range	✓	✓	✗	✗
Change report filters and variables values	✓	✓	✓	✓
Add/Remove comments	✓	✓	✓	✓
Export report	✓	✓	✓	✓
Export view	✓	✓	✗	✗
Publish view to the dashboard	✓	✓	✓	✓
Add/Remove drill-downs	✓	✓	✓	✗
Add custom ad-hoc drill-down (clicking +)	✓	✓	✗	✗
Add predefined ad-hoc drill-down (clicking +)	✓	✓	✓	✗
Do predefined drill-down (clicking +)	✓	✓	✓	✓
Schedule email export	✓	✓	✓	✗
Sort values by indicators and attributes	✓	✓	✓	✗
View source data	✓	✓	✓	✗
Display media gallery	✓	✓	✓	✗
Create/Upload BellaApps	✓	✓	✓	✗



Report viewers are not available to save changes. Changes are available only for current session.



Control types can be changed in *Share with users* dialogue.

Data Sets

BellaDati BI has its own integrated data warehouse. This warehouse contains virtual databases that represents data with similar characteristics. These virtual databases are called data sets and reports are established on them. Each data set can be connected to multiple data source.

In data set following objects are defined and managed:

- Indicators
- Attributes
- Data sources

Following actions can be performed within data sets:

- Importing data
- Browsing, editing and exporting data
- Structure backup and restore
- Managing joins
- Data Mapping
- Watching data changes
- Cleaning data
- Sharing data
- Creating and removing datasets



Only users with data manager role are allowed to manage the data sets. If you don't have this role, please contact your BellaDati administrator.

Data set summary window

Data set summary window displays:

- Calendar scheduler

Typical data set summary window looks like following:

The screenshot shows the BellaDati interface for a data set named 'ATMs'. The top navigation bar includes 'Dashboards', 'Reports', 'Data sets', and 'Users'. The left sidebar contains 'Settings' (Data set summary, Attributes (3), Drill down paths (0), Indicators (2), Joined data sets (0)), 'Data' (Import Data, Data source, Browse data, Create alarm, Erase data), and 'Action' (Share with users, Duplicate, Save XML backup, Remove Data Set). The main content area displays metadata for the 'ATMs' data set, including the owner 'Alex Williams', creation date 'May 28, 2013 12:30:12 PM', and last changed date 'May 28, 2013 12:30:23 PM'. It also shows a 'Data' section with a calendar view for May 2013, where the date '28' is highlighted with a green background and the time '12:30:23 PM' is displayed. To the right of the calendar is a 'Reports' section listing three reports: 'ATM Analytics summary' (last changed by Sales Admin on May 30, 2014), 'test' (last changed by Peter Fedorov on February 7, 2014), and 'ATM Analytics' (last changed by Alex Williams on May 28, 2013). The bottom of the interface shows the version 'BellaDati 2.7.9.2.0.1' and copyright information.

Action list (submenu layout)

- Data set name: Edit is possible by clicking the name (i.e. in-line edit).
- [Attributes settings](#)
- [Indicators settings](#)
- [Joins settings](#)
- Import data: Allows data manager to [import new data from clipboard or files](#).
- Data source: Management of [connection to external databases, URLs or third party systems](#).
- Browse data: [Data browsing](#), editing and export.
- Create alarm: [Data changes watching](#) option.
- Erase data: Deletes all data from the data set.
- [Share](#) with users: Grants access to the data for other users or user groups, including data filter settings.
- [Backup structure](#): Stores the whole structure of the data set and related reports to XML file.
- Remove data set: Allows removing the whole data set and related reports from BellaDati.

Creating Data Set



Only users with [data manager role](#) are allowed to create and manage the data set.

Open the **Data sets** page from main menu on the top of the screen.

1. Click "Create data set" in the left submenu. A popup window appears.
2. Enter name of the new data set.



Data set name must be unique in the whole domain. In case the name isn't unique, warning message will be displayed and the new data set will not be created.

New empty data set contains **no indicators and attributes defined**. There are two ways how to create them:

- Define indicators and attributes while [importing data](#). If you already have existing data file, we recommend this option to set up the data set more interactively.
- Define [indicators](#) and [attributes](#) manually. The mapping to the imported data will be provided during the import settings stage.



The data set can be created also using the [data import](#).

Importing Data

Data import process allows user to populate BellaDati's datawarehouse. Data can be imported from various sources. There are two import types:

- **Manual** import from local files
- **Automatic** scheduled import from [data sources](#)

Manual import

You can manually import data in the following ways:

- Pasting from [Clipboard](#)
- [Importing data](#) in various formats - from CSV and XML to XLS/XLSX and ZIP.
- Adding row using [data browser](#)

Automatic import

Automatic import enables connecting to external sources and load the data periodically. It is available for following [data sources](#):

- [SQL Databases](#)
- [SAP HANA](#)
- [MongoDB](#)
- [URL/REST API](#) using the HTTP protocol
- [Google Analytics](#)
- [Google Drive](#)
- [SalesForce](#)
- [Facebook](#)
- [Twitter](#)
- [LinkedIn](#)
- [FTP](#)
- [Amiando](#)
- [YouTube](#)
- [Zendesk](#)
- [Existing data set](#)



You can start the data import in already existing data set or together with [creating new data set](#). To execute data import wizard, click on "Import data" item in the left menu in data set. When you are importing data to a new data set, you must specify it's unique name first. See [creating data set](#) for more details.

Importing from Clipboard

Clipboard provides a simple way how to quickly analyze small piece of your data in BellaDati.

- Typically you can use it to analyze data directly from Microsoft Excel or a table on a webpage.

To import data from the clipboard:

1. Select desired area in Excel or a web page
2. Copy it to clipboard (CTRL+C/command C).
3. Open the data sets page and select **Import data**
4. In the left menu, enter the data set name and select **Copy and Paste**.
5. Paste (CTRL+V/command V) the clipboard content into text field and click **Continue**

Import Data

Name:

Import type:

Insert text here:

id	date	dept	type	product group	supplier	value
1	1/22/14	Corporate	DIR	Software	A	2355.864729
2	4/30/14	Internal Audit	IND	Hardware	B	18262.00169
3	5/12/14	Strategy	DIR	OfficeC		12019.44663
4	2/25/14	Risk Management	DIR	Consulting	D	17277.34754
5	3/5/14	Development	DIR	Legal Services	E	7178.981324
6	3/7/14	Finance	DIR	IT Services	F	17703.44208
7	2/17/14	IT	IND	Infrastructure	G	7368.295588
8	5/6/14	HR	DIR	Contractors	H	13950.0146
9	1/23/14	Legal	DIR	Travel	I	3199.586895
10	3/23/14	Security	DIR	Transportation	J	3269.53394
11	2/8/14	Statistics	DIR	Software	A	10471.56306
12	4/24/14	Research	IND	Hardware	B	4487.468724
13	4/18/14	Trade & Exchange	DIR	OfficeC		8051.831944
14	1/13/14	Marketing	DIR	Consulting	D	14925.66229


Importing from File

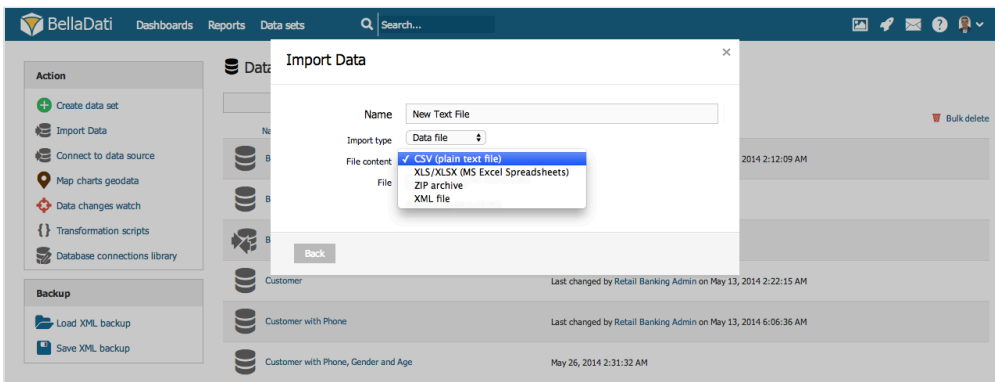
The following file formats are supported for manual import in BellaDati:


- **CSV** (plain text files)
- **Microsoft Excel** (XLS, XLSX) - Office 2003, 2007 and 2010 (previous versions not guaranteed)
- **XML** files
- **ZIP** files (containing one or more supported file formats above)

To import file:

- Go to the **Data Set** page
- Select **Import data**
- Choose **Data file** in **Import type**
- Select appropriate **Data file** format

 After selecting the data file, you need to wait until the file is uploaded.




 Please note, that default **maximum file size** to import is **20MB**. **BellaDati Unlimited** or **BellaDati On-Premise** may have different file size limits. You can compress the file size when importing it in a ZIP archive (see below).

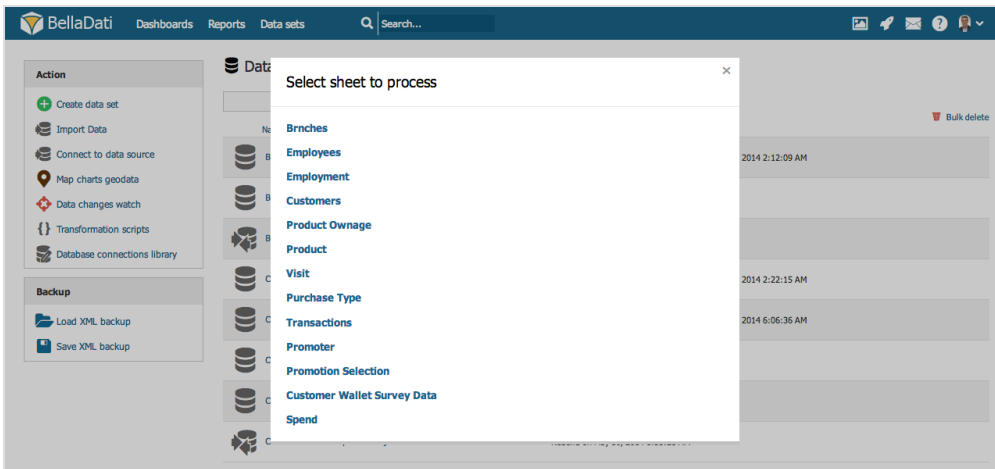
Importing from CSV

When you are importing from CSV, please continue directly to [Import settings](#) page.

Importing from Microsoft Excel

After uploading XLS/XLSX file you will be prompted to select the desired spreadsheet list.

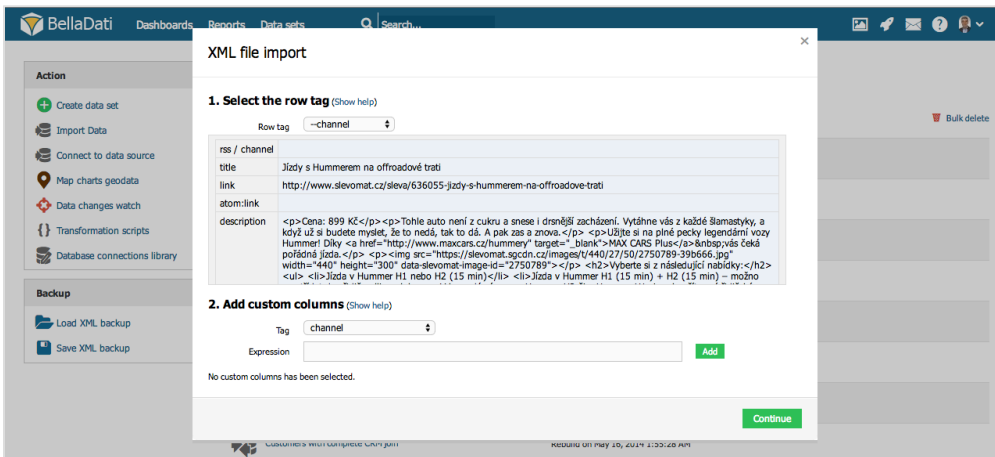
 List selection will not appear when your Excel spreadsheet contains only single list.



Importing from XML

In the XML importing guide, you will be prompted to select the **row tag**, which represents repeatable data sentence. The following example illustrates it on XML file containing employees:

In this case, the **row tag** is `<employee>`.



1. Row tag: Select repeating tag in XML structure. Check extracted content in the preview on the left.
2. Optionally, you can add custom columns repeatedly: Select items and/or attributes when XML structure is not straightforward.



You can use the xPath syntax for the custom columns definition. For more info proceed to [tutorial with example](#), how custom columns can be set.

Importing from ZIP

Importing data compressed as ZIP archive represents a effective way how to reduce imported file size and also upload times significantly. It can contain the following file formats:

- Plain text (CSV)
- Microsoft Excel (XLS, XLSX)
- XML

Please follow corresponding chapters above to continue importing Microsoft Excel or XML file formats.



New data set will not be created until the import process will have been successfully completed.

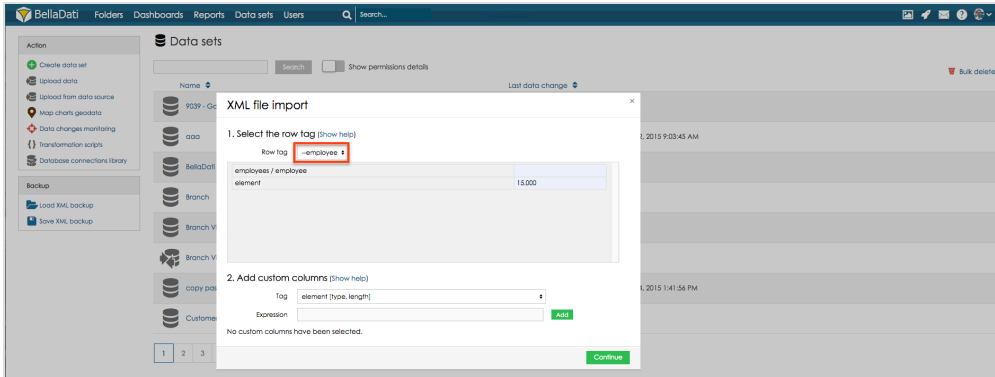


Adding custom columns in XML import

By adding custom columns can be selected, which rows will be imported.


We will use following XML structure for this example:

As row tag should be selected --employee (this tag repeats for each record).




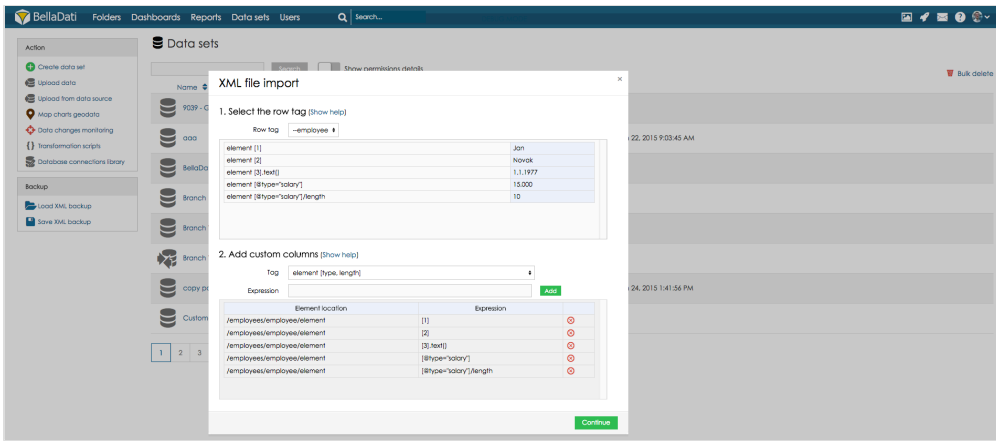
Now, there need to be added new columns, columns are divided by its type (name, surname, birthday, salary).

1. Select element [type, length] tag
2. Add expression for each column

 Custom columns need to be defined by XPath definition.

Expression	Result
[1]	Jan
[1].text()	Jan
[@type="name"]	Jan
[@type="name"].text()	Jan
[2]	Novak
[3]	1.1.1977
[@type="salary"]	15.000
[@type="salary"].text()	15.000
[@type="salary"]/length	10

 Column preview is displayed in section 1. Select the row tag



Click Continue to proceed to [Import settings](#).

Data Sources



Only users with **Data manager** role or full access permission to the data set can set up and control data sources. See [BellaDati permissions and roles](#) for details.

Data Sources are all third party remote systems which can be accessed by BellaDati.

BellaDati contains a set of functions and wizards to help you establishing connection to these data sources, managing them and also provides diagnostic tools. Part of the data sources management is also automatic import scheduler.



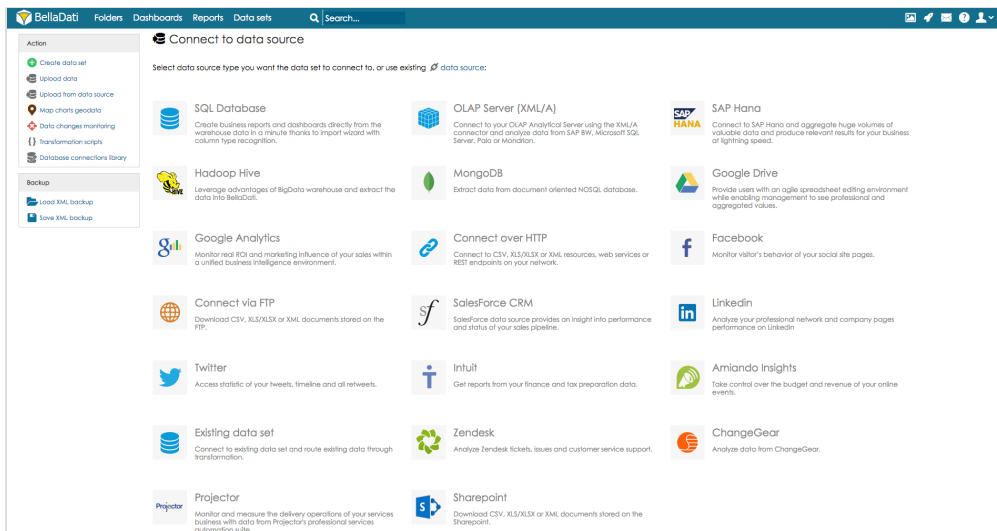
In addition to remote **Data Sources**, BellaDati can import local files. See [Importing Data](#) for more details about **Manual Import**.

BellaDati currently supports these data sources:

- [Connecting to SQL Database](#)
- [Connecting to Microsoft Analysis Services \(SSAS\)](#)
- [Connecting to SAP BW](#)
- [Connecting to SAP HANA](#)
- [Connecting to URL](#)
- [Connecting to MongoDB](#)
- [Connecting to FTP](#)
- [Connecting to Google Analytics](#)
- [Connecting to Google Drive](#)
- [Connecting to Facebook](#)
- [Connecting to Twitter](#)
- [Connecting to LinkedIn](#)
- [Connecting to Zendesk](#)
- [Connecting to Salesforce](#)
- [Connecting to Amiamo](#)
- [Connecting to Intuit](#)
- [Connecting to YouTube](#)
- [Connecting to Existing Data Set](#)
- [Connecting to Hadoop](#)
- [Connecting to MS SharePoint](#)
- [Data source overview](#)



All data must be first loaded into internal BellaDati data warehouse before they are accessible in reports.




Creating Connection

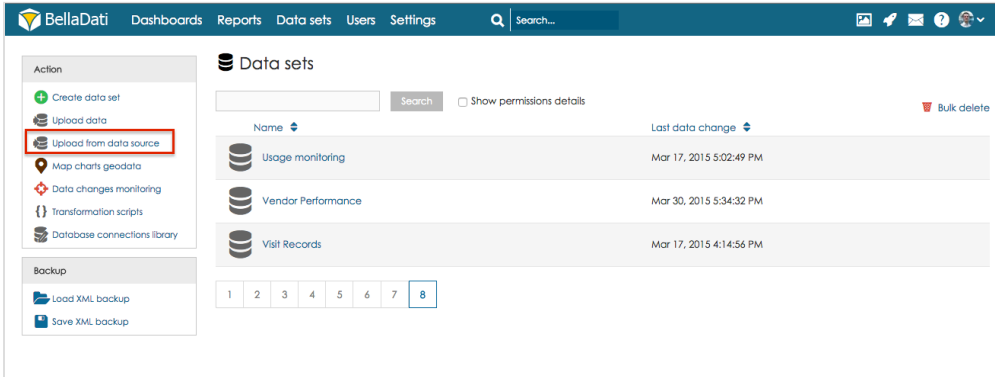
You can create new connection in:

- **Data Sets Panel** from **Action Menu**
- **Data Set Summary** from **Data Menu**

Connecting from Data Set Panel

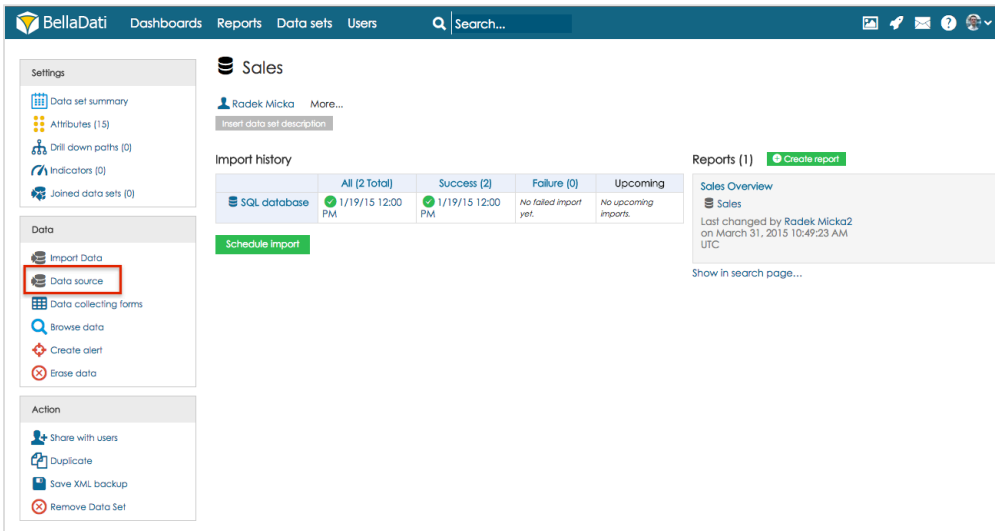
You can connect to **Data Source** from **Action Panel** after clicking **Connect to Data Source**.

 You have to provide name of newly created Data set.



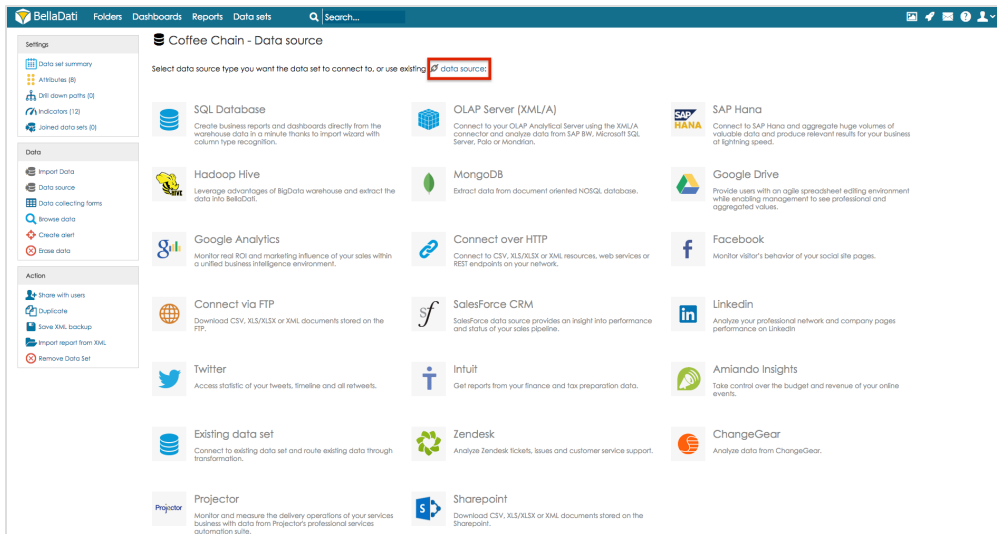
Connecting from Data Set Summary

You can connect to **Data Source** from **Data box** after clicking **Data Source**.



Reusing existing Data Source

If you intend to use the already configured data source more times, you can click "use existing data source" at the top of data source list. The data source configuration will be copied and you will be redirected to [Import Settings](#) page.



Modifications and Operations

General actions:

- **Add:** You can connect to **multiple data sources** within single data set (eg. analyzing different websites together). Same data structure (attributes and indicators mapping) is recommended in this case. You can select among more existing data sources via drop-down menu on the left.
- **Import data:** Launches instant synchronization ([overwriting policy](#) and repeating interval can be set).
- **Check availability:** Allows you to verify if the data source is available.
- **Import settings:** Allows you to change [import mapping](#) to reflect data source structure changes.
- **Schedule:** Links to [Synchronization scheduler](#).
- **Delete:** Delete the data source and all it's settings. Data already imported will remain intact.
- **Basic info:** You can edit data source name here. Data source type is displayed here.
- **Cancel scheduled executions:** You can cancel future planned synchronization.



Each data source has specific configurable parameters - see details for particular data source.

Connecting to SQL Database

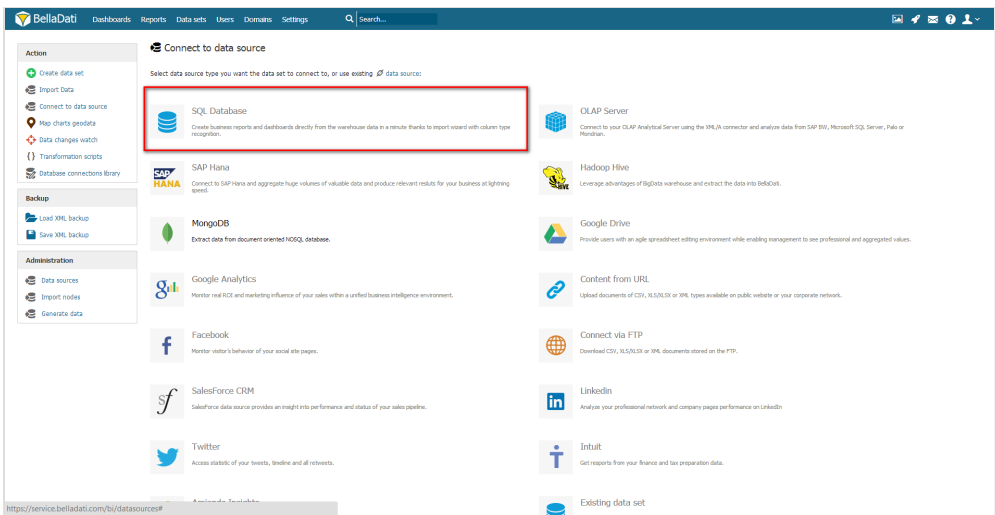
BellaDati can be connected to numerous Databases.



Please consult with [support](#) particular solutions.

Connecting to Database

From **Data Source Connectors** page select **SQL Database Connector**.



Supported Databases

Depending on application usage (cloud or On-Premise) BellaDati is able to connect to these SQL databases.



Support for database not listed below can be added on-demand.

Name	Supported versions	Driver versions
Oracle	10.1.0.5, 10.2.0.1-10.2.0.5, 11.1.0.7, 11.2.0.1, 11.2.0.2	v6-11.1.0.7, v6-11.2.0.1, v6-11.2.0.2, v14-10.1.0.5, v14-10.2.0.1, v14-10.2.0.2, v14-10.2.0.3, v14-10.2.0.4, v14-10.2.0.5
MySQL	3.1.3 and higher	5.1.13
PostgreSQL	8 and higher	9.0-801-jdbc4
Microsoft SQL Server	MSSQL 2008, MSSQL 2008 R2, MSSQL 2008 Express and higher	sqljdbc4
SAP Max DB	7.3 and higher	7.4.4 Build 003-000-002-502
SAP HANA	all	SAP In-Memory Database JDBC Driver, 1.00.48 Build 0372847-1510
Hadoop Hive	all	0.11
Microsoft Access	all	Java SE JDBC/ODBC
Sybase	all	jconn4
Teradata	11 and higher	terajdbc4
Firebird		Jaybird 2.2.7



Database support varies by BellaDati Cloud or On-Premise integration environment.

Connection Parameters

Connection parameters may vary depending on the selected database vendor. Most common parameters are the following

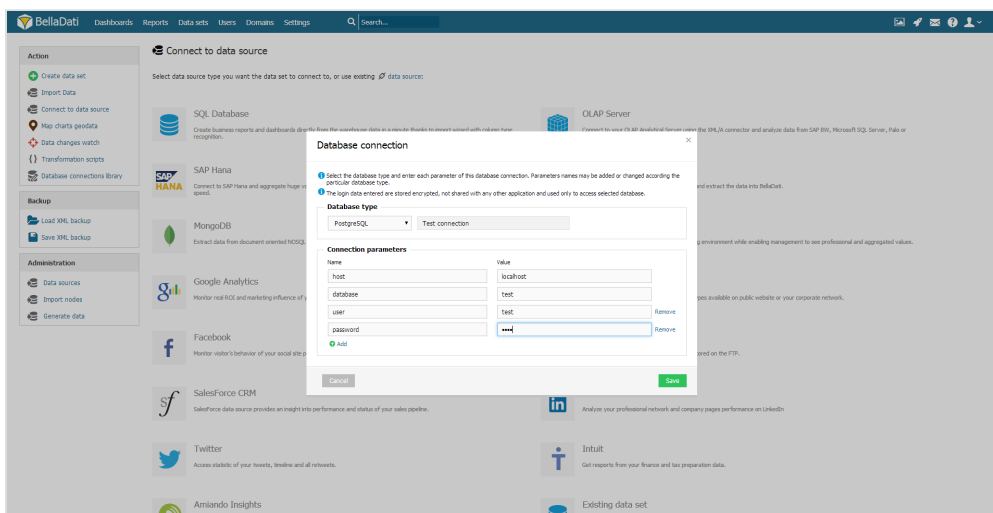
- **host:** IP address or domain name
- **database:** database name
- **password**
- **user**

Additional parameters can be specified by clicking on **Add** link in bottom left part of *Connection parameters* window. They can include:

- **port**
- **driver:** If different drivers are required for various database versions, you can select the right version here (eg. Oracle).



Connection parameters above may vary according database vendor. Please, refer to your database vendor's documentation for details or see the [Connection parameters examples](#) for examples.



Connection to the database will be checked immediately - if a problem arises, you will be informed via error message. Please also check your firewall settings - if BellaDati can connect to the data source.*



Direct connection using **localhost** keyword or localhost IP address is **disabled** due to security reasons. Please define an alias in hosts file (eg. C:\WINDOWS\system32\drivers\etc in Windows). Then use this alias in BellaDati.

Troubleshooting

If you cannot connect to your database, please verify:

1. **Host, port, driver** and **database name** (where applicable) are correct. **Host** should be an IP address or a domain name.
2. The **database server is reachable** from the server running BellaDati. For BellaDati cloud, this means your database must be reachable from the internet.
3. The **database server's firewall** allows incoming requests from the BellaDati server on the database port.
4. Database **username** and **password** are correct.

Querying Database


There are two options how to query database:

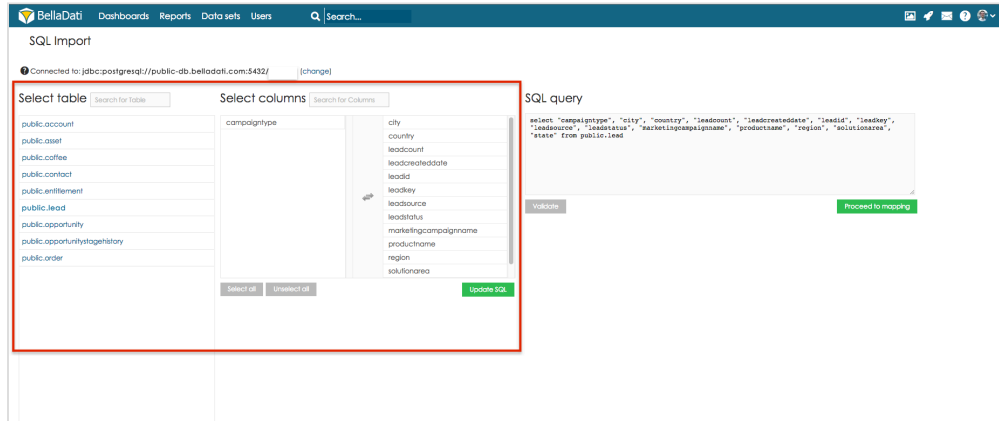
- **Database Discovery**
- **SQL Query Window**

Database Discovery

Database Discovery is a visual editor for specifying database queries. To obtain data from the your database:


1. Click **table** you want to query. BellaDati includes all columns by default.
2. All available **columns** are selected to be imported. Click on values that you don't want to import. Click on button **Update SQL** to update sql query according to changes. Use Select all/Unselect all buttons to quickly manipulate with columns.

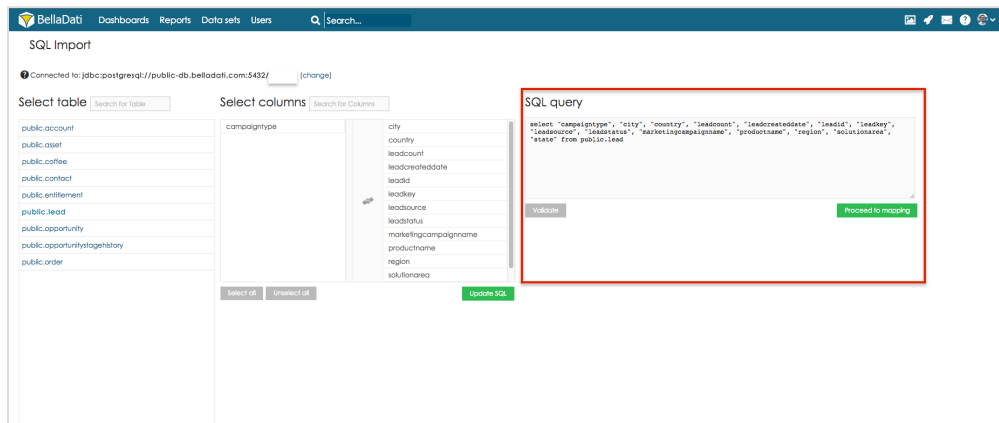
 BellaDati will construct corresponding SQL query in the right SQL window



SQL Query Window

For advanced users or queries, BellaDati offers **SQL Query Window**. Use **Query Window** to construct desired SQL commands.


 Click **validate** button to make sure that your command is correct before proceeding with data mapping.



Connection Modifications

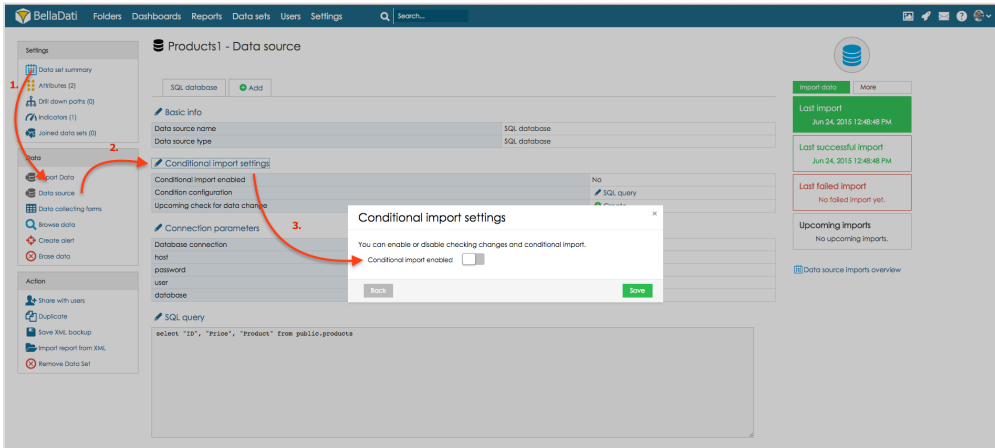
Following data source parameters can be modified within the existing data source in *Data Set* page:

- Connection parameters
- SQL statement

 When changing SQL query to extract more columns from the database, use **Reset values** function and then perform new import settings. Otherwise the additional columns will not be imported.

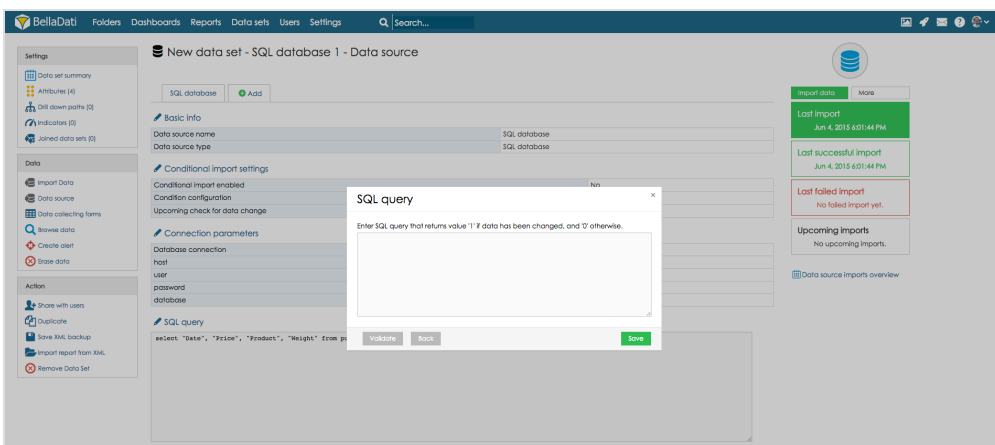
Conditional SQL import

Conditional SQL import allows you to set option to run the import only in cases when the data in database have changed. To allow this feature go to selected data source, open Conditional settings dialogue window and select option *Conditional import enabled*.



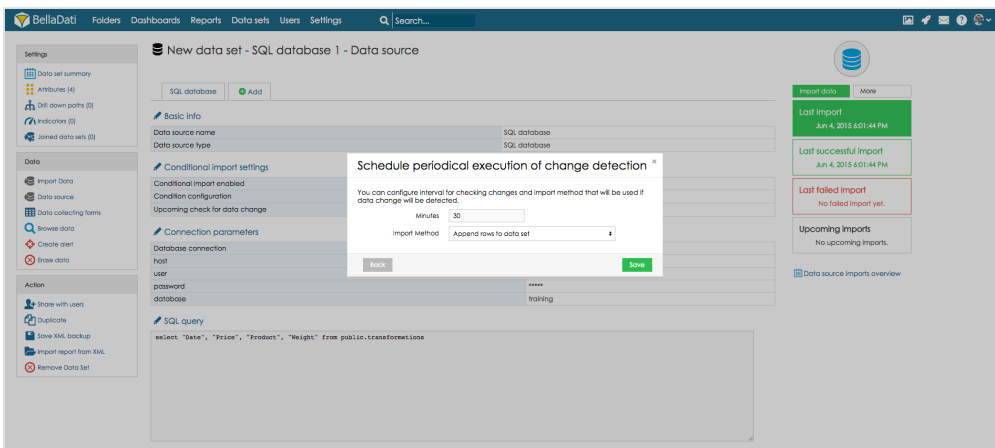
When this option is selected, scheduled import will run only if the data in the database have changed.

Click on button *SQL query* to open dialogue window where should be added SQL query which will check if the data have been changed. If the query returns data import will run.



Click on button *Create* to setup import settings of conditional import:

- Set *Time interval* in which changes will be checked (minimum time interval can be set on 3 minutes)
- Select import method used for this import



Connection parameters examples

PostgreSQL

host	db-host
port	5432
database	test
user	postgres
password	test
SQL test	<code>select * from test</code>

MySQL

host	db-host:3306
database	test
user	root
password	test
SQL test	<code>select * from test</code>



User account and accessed database must have the remote access enabled.
How Do I Grant Access To An Existing Database?


Oracle


host	db-host
port	1521
driver	V14_10_2_0_1
user	tester
password	test
SID	XE
SQL test	<code>select * from test</code>

MS SQL

host	db-host
port	1433
user	sa
password	test
database	test

SQL test	select * from test
----------	--------------------

 SQL Server Authentication must be enabled.

 You need an SQL account to access the MS SQL remotely.

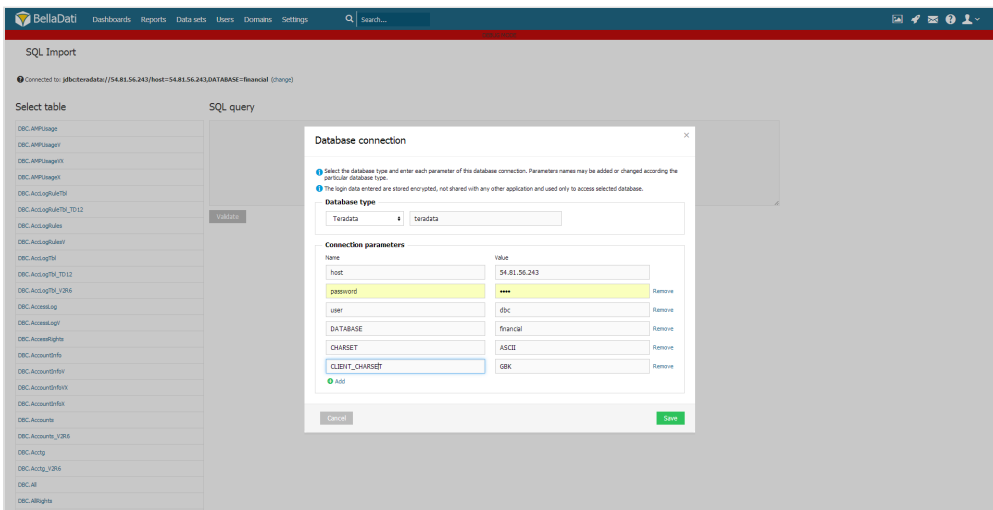
Sybase SQL Anywhere 11

host	db-host
port	2638
user	dba
password	sql
database	demo
SQL test	select * from products

 Browse all JDBC parameters

Teradata 13

host	db-host
user	dbc
password	test
SQL test	select * dbc.tables



 Visit **Teradata** documentation for additional parameters https://developer.teradata.com/doc/connectivity/jdbc/reference/current/jdbcug_chapter_2.html#BABJIHBJ

Upgrading JDBC drivers



This page is relevant for on-premise users only.

If you have installed BellaDati using BellaDati installer, **stop BellaDati**, **replace the desired JDBC driver** file in the `BELLADATI_HOME/app/WEB-INF/lib` directory and **start** your BellaDati instance.

If you are using Application Server (GlassFish, Websphere etc) **replace the desired JDBC driver** file in `BELLADATI_HOME/app/WEB-INF/lib` directory of BellaDati WAR archive (it's a plain ZIP archive) and **redploy** the application.

Using variables in SQL query

Date and Time Variables

If you need to change the SQL query dynamically, you can use predefined variables. BellaDati currently supports functions to get date, time or timestamp in user defined formats:

Name	Description	Examples
<code>\$date(dateString)</code>	Evaluates the <code>dateString</code> and outputs the date in <code>yyyy-MM-dd</code> format. The <code>dateString</code>	<code>\$date(now + 5d -4w)</code> <code>\$date(2011-01-01 + 5d -4w)</code> <code>\$date(actualMonth -1d)</code>
<code>\$date(dateString, format)</code>	Works like <code>\$date(dateString)</code> , but output format is controlled by <code>format</code> parameter	<code>\$date(now + 5d -4w, dd-MM-yyyy)</code> <code>\$date(2011-01-01 + 5d -4w, MMyyyy)</code> <code>\$date(actualMonth -1d, yyyy-dd-MM)</code>
<code>\$time(timeString)</code>	Evaluates the <code>timeString</code> and outputs the resulting time in <code>HH:mm:ss</code> format	<code>\$time(now)</code> <code>\$time(actualhour)</code> <code>\$time(actualminute)</code>
<code>\$time(timeString, format)</code>	Works like <code>\$time(timeString)</code> , but output format is controlled by <code>format</code> parameter	<code>\$time(now, HH:mm:ss)</code> <code>\$time(actualhour, MMss)</code> <code>\$time(actualminute, HHmmss)</code>
<code>\$timestamp()</code>	Returns the current time stamp value	<code>\$timestamp()</code>
<code>\$lastSuccessfulImport()</code>	Returns time stamp with last successful import	<code>\$lastSuccessfulImport()</code>
<code>\$firstValue(L_ATTRIBUTE_CODE)</code>	Returns the lowest value (sorted ascending) of the attribute specified by <i>attribute ID</i> stored in the current data set. Returns empty string if there are no data or the attribute code is not valid.	<code>\$firstValue(L_ID) //returns 123456</code> <code>\$firstValue(L_DATE_ATTRIBUTE) //returns 2013-01-01</code> <code>\$firstValue(L_TIME_ATTRIBUTE) //returns 10:00:54</code>
<code>\$lastValue(L_ATTRIBUTE_CODE)</code>	Returns the highest value (sorted descending) of the attribute specified by <i>attribute ID</i> stored in the current data set. Returns empty string if there are no data or the attribute code is not valid.	<code>\$lastValue(L_ID) //returns 123456</code> <code>\$lastValue(L_DATE_ATTRIBUTE) //returns 2013-12-31</code> <code>\$lastValue(L_TIME_ATTRIBUTE) //returns 23:59:59</code>

DateString

- **now** - represents actual date
- **actualyear** - represents the first day of actual year (1.1.20XX). For example actualyear selected on 21.9.2010 represents date 1.1.2010
- **actualquarter** - represents the first day of actual quarter (1.1.20XX, 1.4.20XX, 1.7.20XX, 1.10.20XX). For example actualquarter selected on 21.9.2010 represents date 1.7.2010
- **actualmonth** - represents the first day of actual month (1.1.20XX, 1.2.20XX, ...). For example actualmonth selected in 21.9.2010 represents date 1.9.2010
- **actualweek** - represents first day of actual week (Monday). For example actualweek selected on 21.9.2010 represents date 20.9.2010 (Monday of this week in calendar)
- **availableFrom, availableTo** - represents the first and last available date entry
- relative and absolute enterig of date can be adjusted by operators using this syntax: **date +/- n[d|w|m|q|y]**, where **n** is integer, **d** represents day, **w** represents week, **m** represents month **q** represents quartal and **y** represents year. We can for example define time in this way: `actualyear + 2m -4d`. Today is 21.9.2010, so this value represents 1.1.2010 + 2 months - 4 days, which means date 25.2.2010.

TimeString

- **now** - represents actual time

- **actualhour** - represents the actual hour at 0 minutes and 0 seconds.
- **actualminute** - represents the actual minute at 0 seconds
- **actualecond** - represents the actual second

Connecting to Microsoft Analysis Services (SSAS)

Prerequisites

Microsoft SQL Server 2005+ with configured [XMLA Analysis Services](#) endpoint. To allow XML for Analysis :

- configure the firewall <http://technet.microsoft.com/en-us/library/ms174937>
- configure the IIS <http://technet.microsoft.com/en-us/library/gg492140>

Exploring OLAP cubes and creating MDX query

Connection string URL format:

MDX query example:

The screenshot shows the BellaDati XMLA Import interface. The top navigation bar includes 'Dashboards', 'Reports', 'Data sets', and 'Users'. The main content area is titled 'XMLA Import' and shows a connection to 'jdbcxmla:Server=http://bdmssql.cloudapp.net/OLAP/msmdpump.dll;'. Below this, there are three main sections: 'Available cubes', 'Available dimensions', and 'MDX query'. The 'Available cubes' section lists 'Adventure Works' and 'Mined Customers'. The 'Available dimensions' section lists various dimensions like 'Measures', 'Date', 'Ship Date', 'Delivery Date', 'Customer', 'Reseller', 'Geography', 'Employee', 'Promotion', 'Product', 'Sales Territory', 'Source Currency', 'Sales Reason', 'Sales Summary Order Details', 'Internet Sales Order Details', 'Reseller Sales Order Details', and 'Sales Channel'. The 'MDX query' section contains a sample query: 'SELECT ([Measures].[Sales Amount], [Measures].[Tax Amount]) ON COLUMNS, ([Date].[Fiscal].[Fiscal Year]) ON ROWS FROM [Adventure Works]'. Below the query, there are 'Validate' and 'Proceed to mapping' buttons.

Available cubes	Available dimensions	MDX query
<ul style="list-style-type: none">Adventure WorksMined Customers	<ul style="list-style-type: none">MeasuresDateShip DateDelivery DateCustomerResellerGeographyEmployeePromotionProductSales TerritorySource CurrencySales ReasonSales Summary Order DetailsInternet Sales Order DetailsReseller Sales Order DetailsSales Channel	<pre>SELECT ([Measures].[Sales Amount], [Measures].[Tax Amount]) ON COLUMNS, ([Date].[Fiscal].[Fiscal Year]) ON ROWS FROM [Adventure Works]</pre>

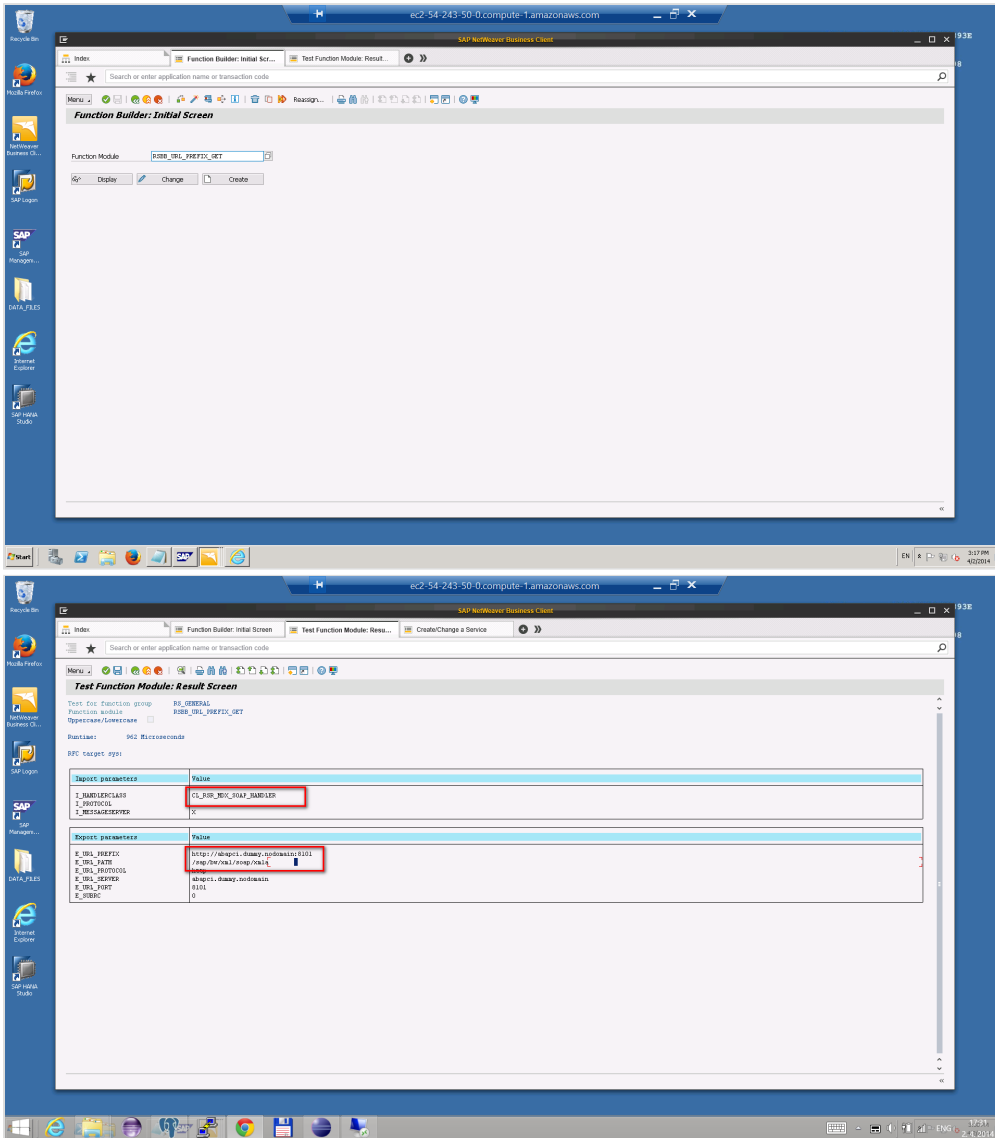
Connecting to SAP BW

Prerequisites

SAP Netweaver BW 7.3+. SOAP web services must be enabled in order to use the [XML for Analysis](#) interface. Further requirements:

- web services and ICF must be enabled, see http://help.sap.com/saphelp_nw73/helpdata/en/b3/1dd13ffc9a4a21e10000000a1550b0/frameset.htm

Screenshots from testing the ICF module:



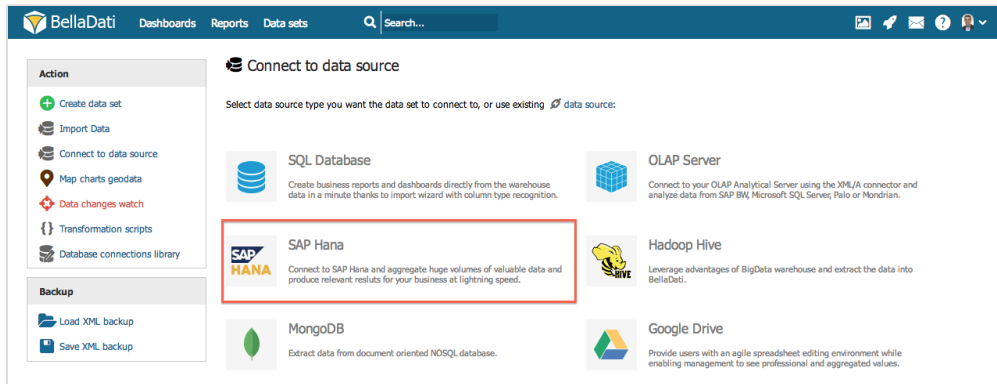
You can then easily verify that everything is configured well - just try to get the WSDL schema over the HTTP. This is the result:

Connecting to SAP HANA

BellaDati can be connected to SAP HANA in-memory Big Data database.

Connecting to HANA

From **Data Source Connectors** page select **SAP HANA**.

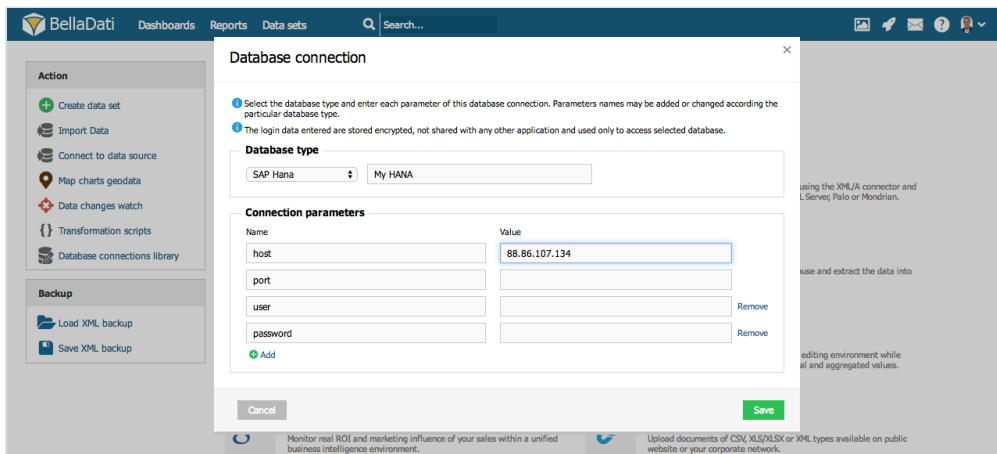


Connection Parameters

You must specify following parameters:

- **host**: IP address or domain name
- **port**
- **password**
- **user**

Additional parameters can be specified by clicking on **Add** link in bottom left part of *Connection parameters* window.



Connection to the database will be checked immediately - if a problem arises, you will be informed via error message. Please also check your firewall settings - if BellaDati can connect to the data source.*

Querying Database


There are two options how to query SAP HANA:

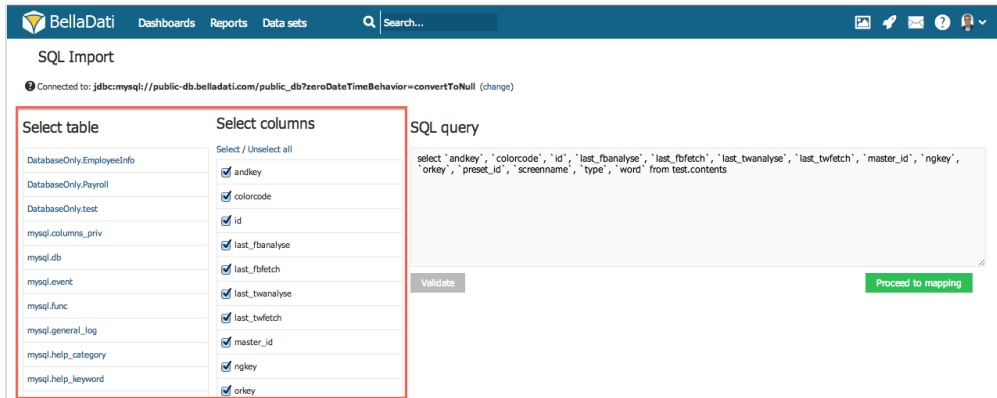
- **Database Discovery**
- **SQL Query Window**

Database Discovery

Database Discovery is a visual editor for specifying database queries. To obtain data from the your database:

1. Click the **table** you want to query. BellaDati includes all columns by default.
2. Check **columns** you want to import. Use select/deselect all to quickly manipulate with columns.

 BellaDati will construct corresponding SQL query in the right SQL window



SQL Import

Connected to: jdbc:mysql://public-db.belladati.com/public_db?zeroDateTimeBehavior=convertToNull (change)

Select table	Select columns
DatabaseOnly.EmployeeInfo	Select / Unselect all
DatabaseOnly.Payroll	<input checked="" type="checkbox"/> andkey
DatabaseOnly.test	<input checked="" type="checkbox"/> colorcode
mysql.columns_priv	<input checked="" type="checkbox"/> id
mysql.db	<input checked="" type="checkbox"/> last_fbanalyze
mysql.event	<input checked="" type="checkbox"/> last_fbfetch
mysql.func	<input checked="" type="checkbox"/> last_twanalyze
mysql.general_log	<input checked="" type="checkbox"/> last_twfetch
mysql.help_category	<input checked="" type="checkbox"/> master_id
mysql.help_keyword	<input checked="" type="checkbox"/> ngkey
	<input checked="" type="checkbox"/> orkey


SQL query

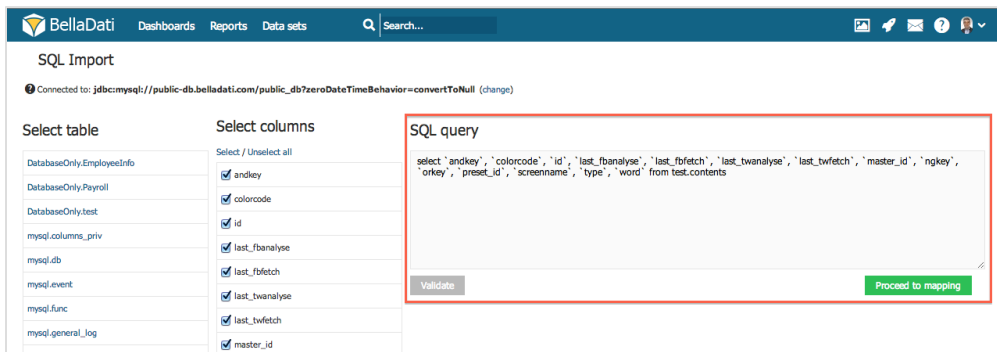
```
select 'andkey', 'colorcode', 'id', 'last_fbanalyze', 'last_fbfetch', 'last_twanalyze', 'last_twfetch', 'master_id', 'ngkey', 'orkey', 'preset_id', 'screenname', 'type', 'word' from test.contents
```

Validate Proceed to mapping

SQL Query Window

For advanced users or queries, BellaDati offers **SQL Query Window**. Use **Query Window** to construct desired SQL commands.

 Click **validate** button to make sure that your command is correct before proceeding with data mapping.



SQL Import

Connected to: jdbc:mysql://public-db.belladati.com/public_db?zeroDateTimeBehavior=convertToNull (change)

Select table	Select columns
DatabaseOnly.EmployeeInfo	Select / Unselect all
DatabaseOnly.Payroll	<input checked="" type="checkbox"/> andkey
DatabaseOnly.test	<input checked="" type="checkbox"/> colorcode
mysql.columns_priv	<input checked="" type="checkbox"/> id
mysql.db	<input checked="" type="checkbox"/> last_fbanalyze
mysql.event	<input checked="" type="checkbox"/> last_fbfetch
mysql.func	<input checked="" type="checkbox"/> last_twanalyze
mysql.general_log	<input checked="" type="checkbox"/> last_twfetch
mysql.help_category	<input checked="" type="checkbox"/> master_id
mysql.help_keyword	<input checked="" type="checkbox"/> ngkey
	<input checked="" type="checkbox"/> orkey

SQL query

```
select 'andkey', 'colorcode', 'id', 'last_fbanalyze', 'last_fbfetch', 'last_twanalyze', 'last_twfetch', 'master_id', 'ngkey', 'orkey', 'preset_id', 'screenname', 'type', 'word' from test.contents
```

Validate Proceed to mapping

Connecting to URL

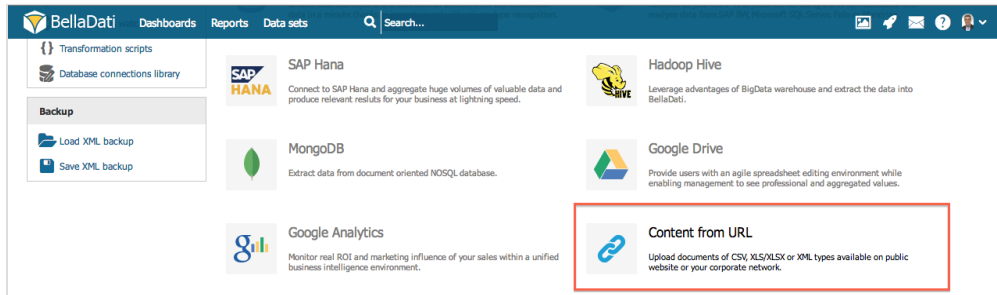
BellaDati can import data from URLs, web services and REST APIs.

Connecting to URL has similarities with [file import](#). In addition URL imports can be scheduled to execute automatically and repeatedly.

 URL popup offers advanced section for **authentication** and specific **HTTP headers** settings.

Connecting to URL

From **Data Source Connectors** page select **Connect from URL**.




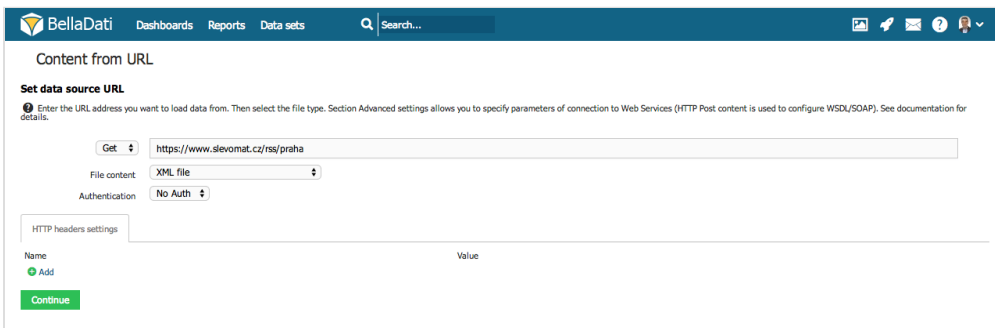
Connection Parameters

Enter following parameters to connect to URL source:

- **URL:** Web address
- **File content:** Select the file format - CSV (text file), Excel (XLS, XLSX), XML or ZIP
- **Authentication:** No Authentication, Simple, oAuth1a, oAuth2

Then continue setting like for [file import](#).

 Login and password are optional parameters.



Date and Time Variables

If you need to change the URL or HTTP post content dynamically, you can use predefined variables. BellaDati currently supports functions to get date, time or timestamp in user defined formats:

Name	Description	Examples
<code>\$date(dateString)</code>	Evaluates the <code>dateString</code> and outputs the date in <code>yyyy-MM-dd</code> format. The <code>dateString</code>	<code>\$date(now + 5d -4w)</code> <code>\$date(2011-01-01 + 5d -4w)</code> <code>\$date(actualMonth -1d)</code>

<code>\$date(dateString, format)</code>	Works like <code>\$date(dateString)</code> , but output format is controlled by <code>format</code> parameter	<code>\$date(now + 5d -4w, dd-MM-yyyy)</code> <code>\$date(2011-01-01 + 5d -4w, MMyyyy)</code> <code>\$date(actualMonth -1d, yyyy-dd-MM)</code>
<code>\$time(timeString)</code>	Evaluates the <code>timeString</code> and outputs the resulting time in HH:mm:ss format	<code>\$time(now)</code> <code>\$time(actualhour)</code> <code>\$time(actualminute)</code>
<code>\$time(timeString, format)</code>	Works like <code>\$time(timeString)</code> , but output format is controlled by <code>format</code> parameter	<code>\$time(now, HH:mm:ss)</code> <code>\$time(actualhour, MMss)</code> <code>\$time(actualminute, HHmmss)</code>
<code>\$timestamp()</code>	Returns the current time stamp value	<code>\$timestamp()</code>

DateString

- **now** - represents actual date
- **actualyear** - represents the first day of actual year (1.1.20XX). For example actualyear selected on 21.9.2010 represents date 1.1.2010
- **actualquarter** - represents the first day of actual quarter (1.1.20XX, 1.4.20XX, 1.7.20XX, 1.10.20XX). For example actualquarter selected on 21.9.2010 represents date 1.7.2010
- **actualmonth** - represents the first day of actual month (1.1.20XX, 1.2.20XX, ...). For example actualmonth selected in 21.9.2010 represents date 1.9.2010
- **actualweek** - represents first day of actual week (Monday). For example actualweek selected on 21.9.2010 represents date 20.9.2010 (Monday of this week in calendar)
- relative and absolute enterig of date can be adjusted by operators using this syntax: **date +/- n[d|w|m|q|y]**, where **n** is integer, **d** represents days, **w** represents week, **m** represents month **q** represents quartal and **y** represents year. We can for example define time in this way: *actualyear + 2m -4d*. Today is 21.9.2010, so this value represents 1.1.2010 + 2 months - 4 days, which means date 25.2.2010.

TimeString

- **now** - represents actual time
- **actualhour** - represents the actual hour at 0 minutes and 0 seconds.
- **actualminute** - represents the actual minute at 0 seconds
- **actalsecond** - represents the actual second

Connecting to SOAP web Services

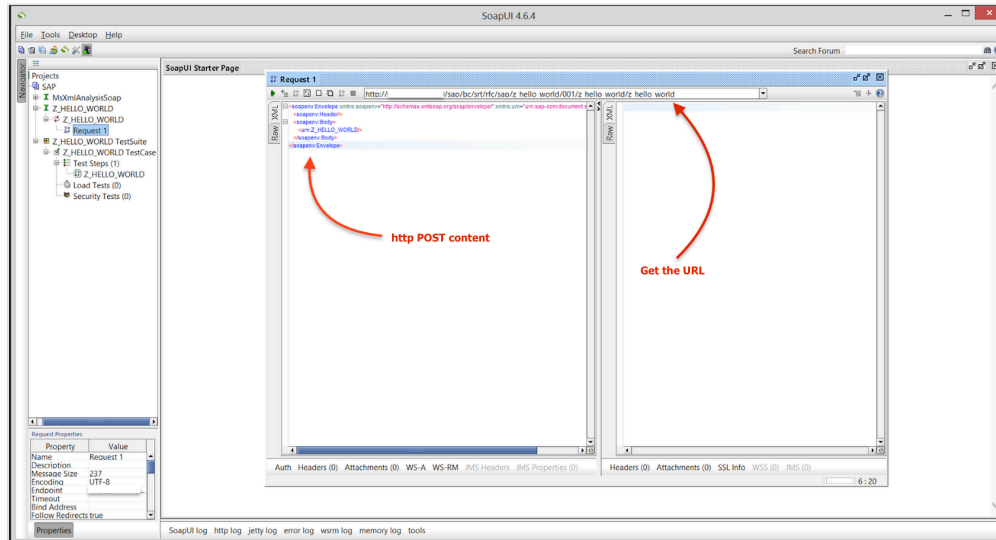
BellaDati is able to connect resources available on network via the HTTP protocol. Except this simple usage, we can connect also more complex resources available as Web services. Web services are using the SOAP protocol, which is based on the plain HTTP protocol. The SOAP message comes in standard XML format, which is in BellaDati perfectly supported. Here is an example how to do it:

1. Enter the endpoint URL of your web service and choose the [XML](#) file format.
2. Open the advanced settings, choose the POST method and set the following parameters:
 - a. SOAPAction - value is contained in the WSDL file, which describes your web service. It is defined in the `soapAction` tag, e.g.: `<soap:operation soapAction="http://www.sap.com/Z_HSI_HRP04_RZH_READ_DATA"/>`
 - b. Content-Type - set the value to `text/xml`
3. Insert the POST content in the depicted structure:



In case you suffer any issues while connecting to SAP Web Services, we recommend to use external tools like [SoapUI](#) to which will generate right structure of XML structure based on WSDL.

SoapUI will also help you to generate right http Post content.



Connecting to REST web services

You can connect to REST web services over HTTP using BellaDati. Just select the proper HTTP GET header type and file type (eg. CSV).

Authentication

The following authentication methods are supported:

- **Basic and Digest HTTP Access Authentication** ([RFC2617 standard](#)).



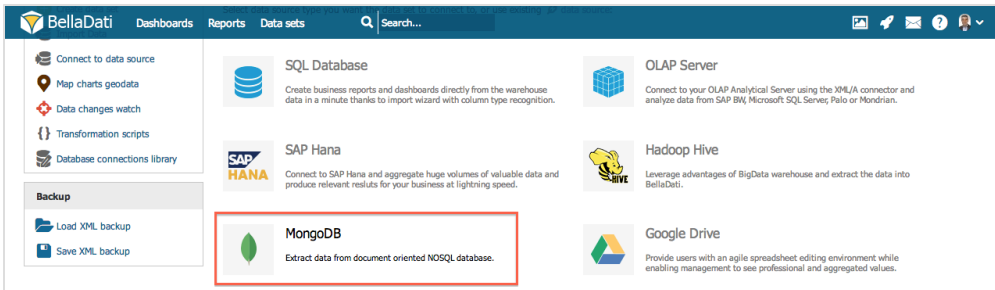
REST API supports **OAuth** standard with security token.

Connecting to MongoDB

BellaDati can be connected to [MongoDb](#) NoSQL database.

Connecting to MongoDB

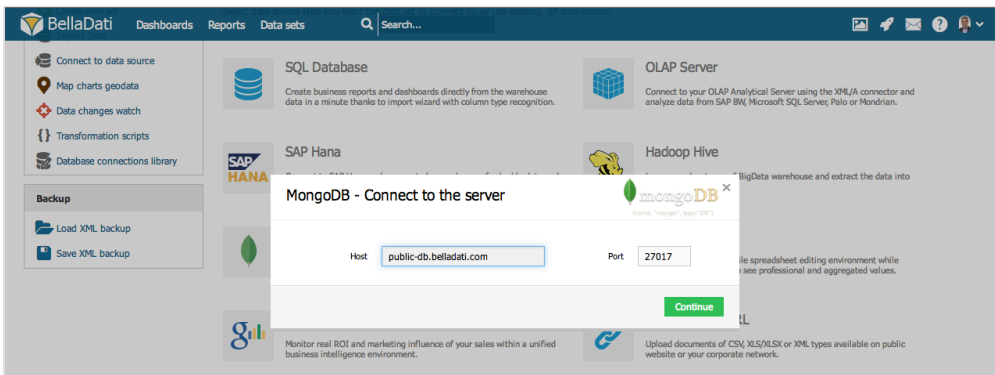
From **Data Source Connectors** page select **MongoDB**.



Connection Parameters

You must specify following parameters:

- **host**: IP address or domain name
- **port**



Selecting Database


Select required database from drop down menu. BellaDati will open **Mongo Console**.

Defining Query

✓ You can easily change actual database by clicking on **change** link.

⚠ BellaDati lists all **collections** from selected databases. Don't forget to use desired **collection** in your queries.


Write Mongo query and hit **Execute**. Results will be displayed in the right window on Mongo Console. Click **Proceed to data mapping** to import data.






BellaDati

[Dashboards](#)


[Reports](#)

[Data sets](#)

Search...



MongoDB import

 Connected to: [public-db.belladati.com:27017/test](#) (change)

Available collections: [system.indexes, tesla]

Command

Execute

```
1 db.tesla.findOne()
```


Result preview

```
{
  "_id": {
    "$oid": "5278c6644635602f10fa049"
  },
  "metadata": {
    "result_type": "recent",
    "iso_language_code": "en"
  },
  "created_at": "Thu Jul 18 10:00:16 +0000 2013",
  "id_str": "357801987238146050",
  "text": "RT @YourGuyYY: Tesla cars are awesome except i don't want to buy a car because self-driving cars will",
  "source": "<a href='\"http://twitter.com/download/iphone\"' rel='\"nofollow\">Twitter for iPhone</a>",
  "truncated": false,
  "in_reply_to_status_id": null,
  "in_reply_to_status_id_str": null,
  "in_reply_to_user_id": null,
  "in_reply_to_user_id_str": null,
  "in_reply_to_screen_name": null,
  "user": {
    "id": 67391878,
    "id_str": "67391878",
    "name": "Nicole ",

```

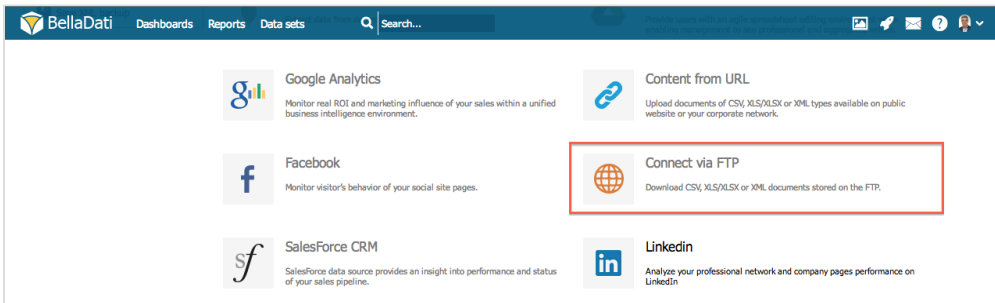

Connecting to FTP

BellaDati can import data from files stored at FTP servers.

 Connecting to FTP is similar to local [file import](#). However, FTP imports can be scheduled and advanced security can be achieved using **SSL** (FTPS/FTPES modes) in conjunction with basic authentication.

Connecting to FTP

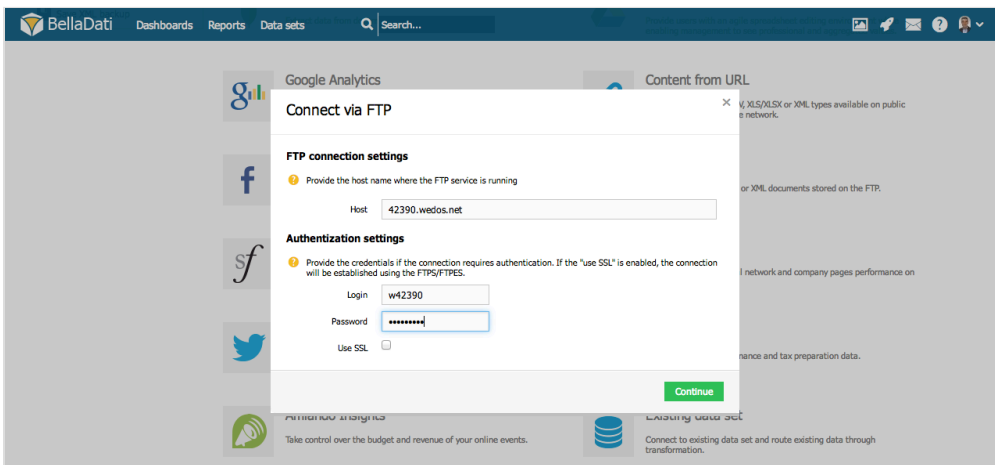
From **Data Source Connectors** page select **Connect via FTP**.




Connection Parameters

Enter following parameters to connect to FTP server:

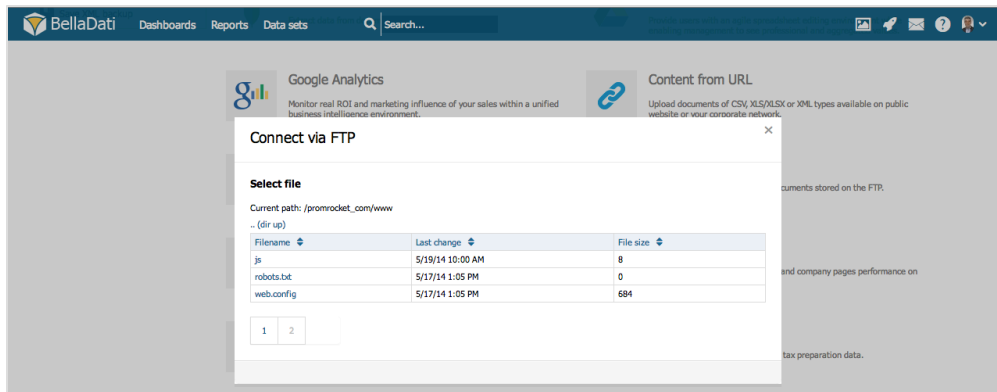
1. **Host:** FTP server address
2. **Login**
3. **Password**
4. **Use SSL:** Enables FTPS/FTPES mode. *(optional)*



 Login and password are optional parameters.

Selecting files

BellaDati will display server files structure. Select file you want to import.



Defining file content

Select file format. Continue to [file import settings](#) to learn more about available file types.



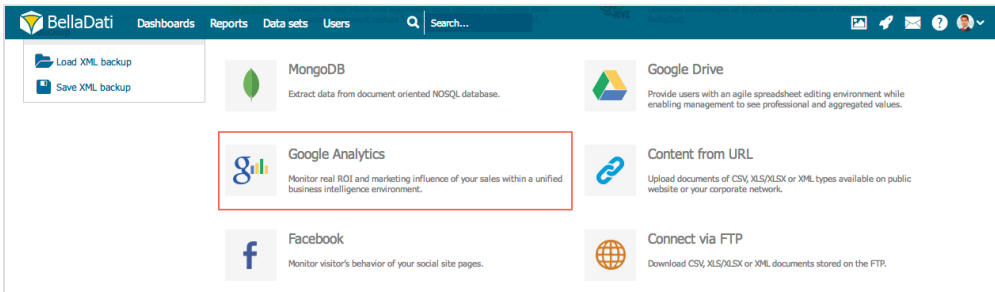
Scheduled imports can be set up for files imported via FTP. Continue to [Scheduling Import](#) to learn more.

Connecting to Google Analytics

BellaDati allows you to connect and analyze data from Google Analytics.

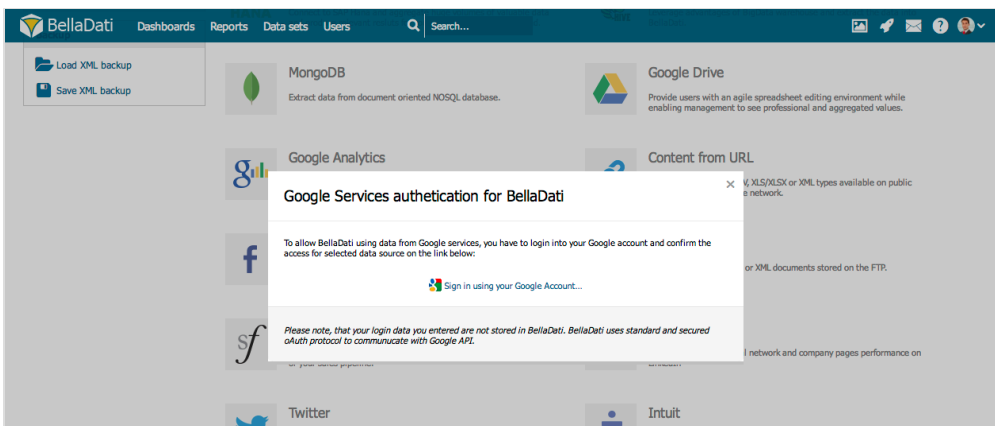
In order to connect to a Google Analytics data source:

- Click **Data sets** from the **Main menu**
- Select **Connect to data sources** at the left menu under **Action** panel.
- Click on the logo of **Google Analytics** as indicated in the red box below.



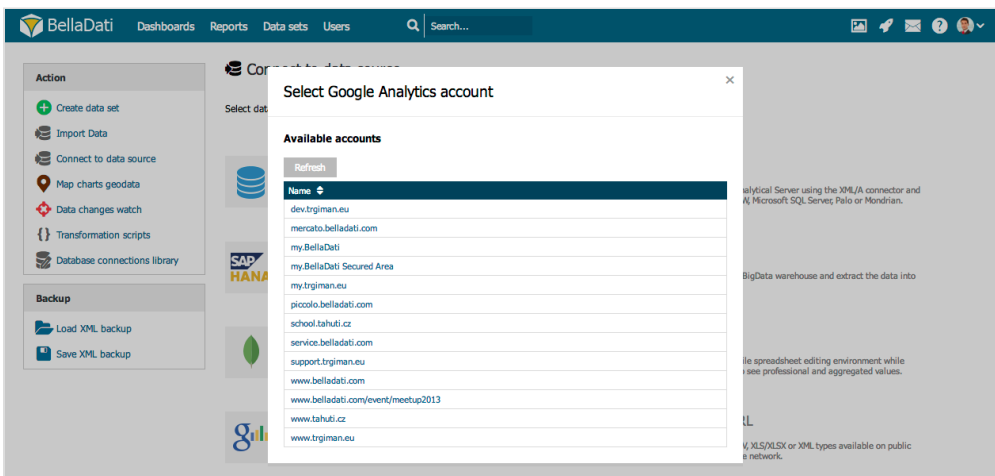
Authentication

Following window will request granting BellaDati access to your Google account. Click **Sign in using your Google account** to open login screen.



Select Page

BellaDati lists all available web pages. Select one to continue.

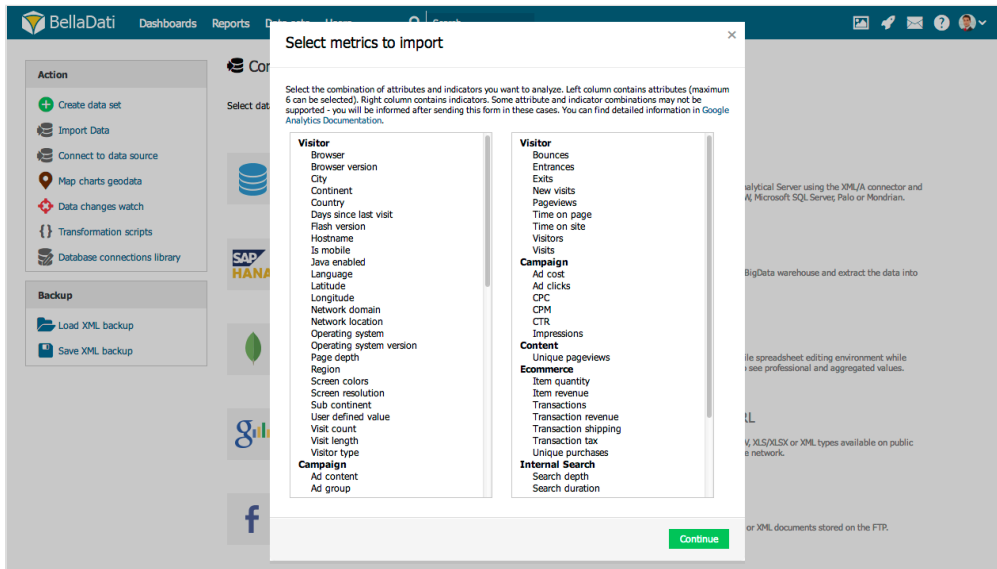


Select Dimensions and Metrics

Click on desired attributes and indicators to be imported.



Some metrics and attributes combinations are not allowed. See [Google Analytics documentation](#) for details.



Modifications

The following modifications are available for the existing Google Analytics data source:

- **From/To date interval:** Influences the period data are imported within (see entering date/time parameters section below).
- Authentication revocation

Entering date/time parameters

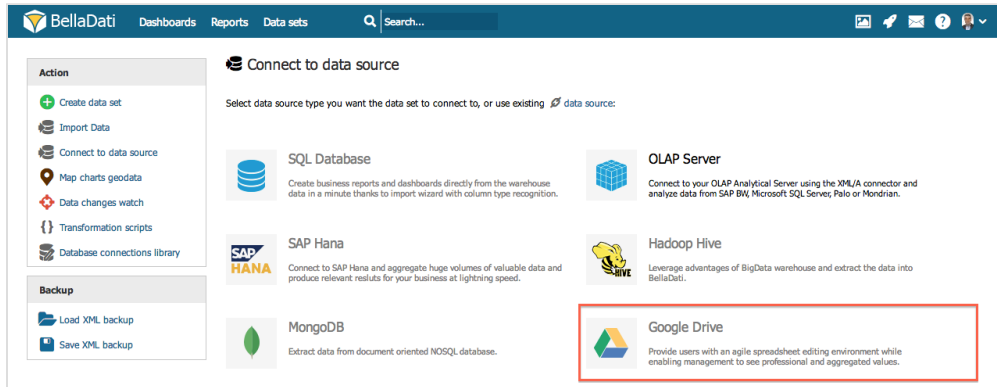
- you can enter time (date) **absolutely in two different formats**: **dd.MM.yyyy** (e.g. 1.12.2010), or **yyyy-MM-dd** (e.g. 2010-12-01)
- it's also possible to enter date **relatively**:
 - **now** - represents actual date
 - **actualyear** - represents the first day of actual year (1.1.20XX). For example actualyear selected on 21.9.2010 represents date 1.1.2010
 - **actualquarter** - represents the first day of actual quarter (1.1.20XX, 1.4.20XX, 1.7.20XX, 1.10.20XX). For example actualquarter selected on 21.9.2010 represents date 1.7.2010
 - **actualmonth** - represents the first day of actual month (1.1.20XX, 1.2.20XX, ...). For example actualmonth selected in 21.9.2010 represents date 1.9.2010
 - **actualweek** - represents first day of actual week (Monday). For example actualweek selected on 21.9.2010 represents date 20.9.2010 (Monday of this week in calendar)
 - relative and absolute enterig of date can be adjusted by operators using this syntax: **date +/- n[d|w|m|q|y]**, where **n** is integer, **d** represents day, **w** represents week, **m** represents month **q** represents quartal and **y** represents year. We can for example define time in this way: *actualyear + 2m -4d*. Today is 21.9.2010, so this value represents 1.1.2010 + 2 months - 4 days, which means date 25.2.2010.

Connecting to Google Drive

BellaDati allows you to connect and analyze data from Google Drive Spreadsheets.

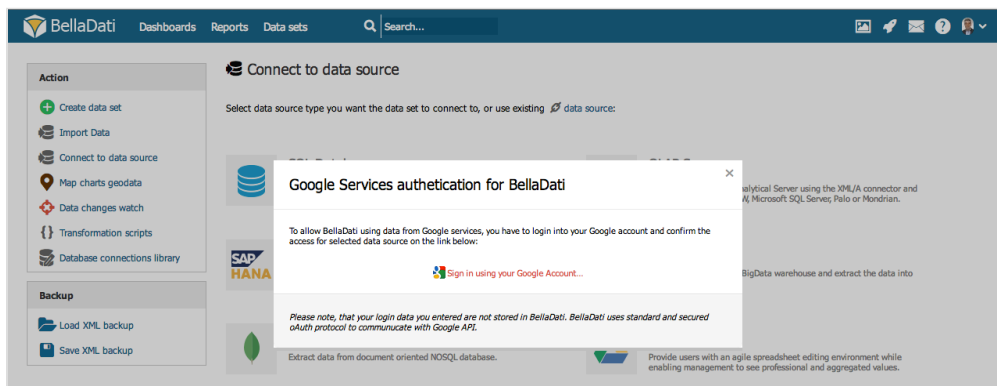
In order to connect to a Google Drive data source:

- Click **Data sets** from the **Main menu**
- Select **Connect to data sources** at the left menu under **Action** panel.
- Click on the logo of **Google Drive** as indicated in the red box below.



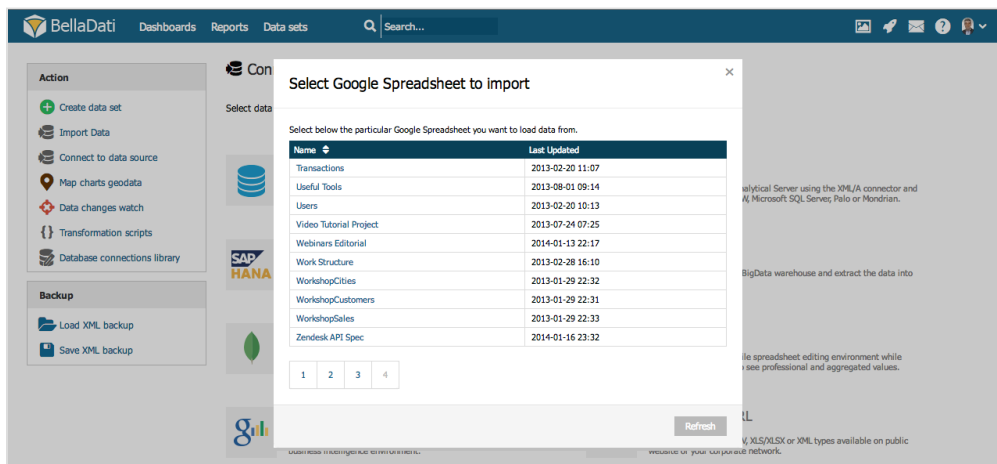
Authentication

Following window will request granting BellaDati access to your Google account. Click **Sign in using your Google account** to open login screen.



Select Data Set

BellaDati lists all available spreadsheets. Select one to continue.

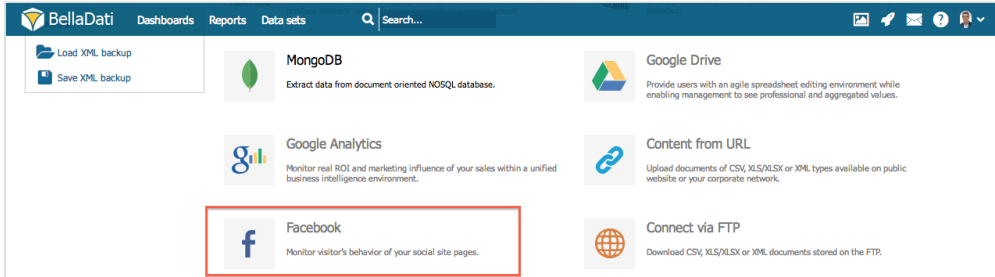


Connecting to Facebook

BellaDati allows you to connect and analyze data from Facebook.

In order to connect to a Facebook data source:

- Click **Data sets** from the **Main menu**
- Select **Connect to data sources** at the left menu under **Action** panel.
- Click on the logo of **Facebook** as indicated in the red box below.

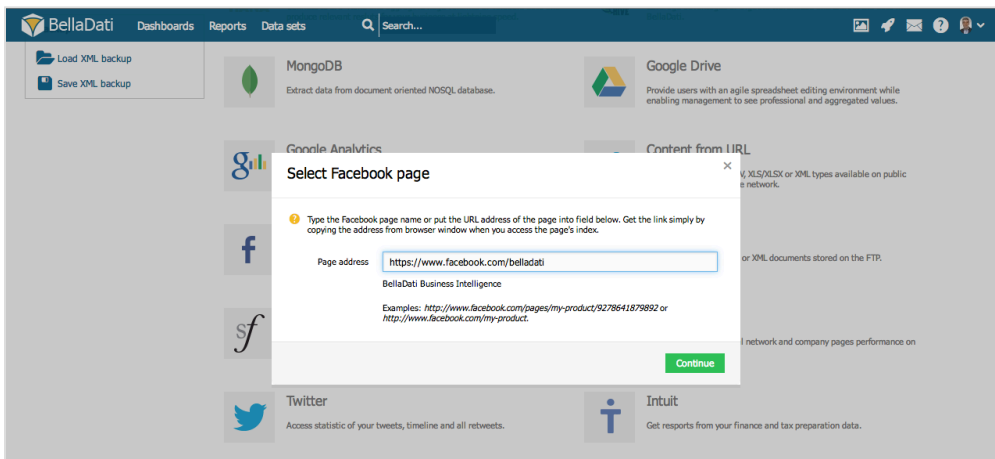


Authentication

Following window will request granting BellaDati access to your Facebook account. Click **Sign in using your Facebook account** to open login screen.

Selecting Facebook Page

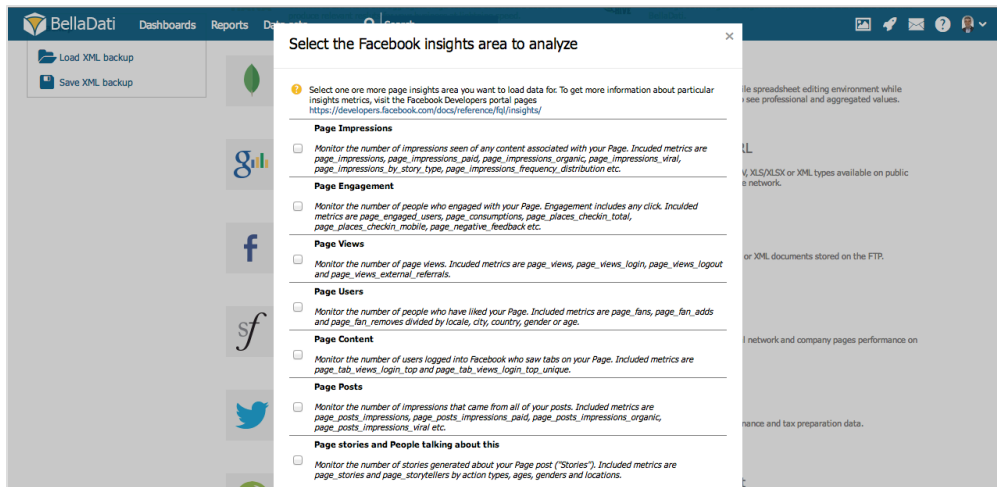
Paste Facebook page URL from which you would like to analyze data.



Selecting Data

Select data you want to analyze. BellaDati offers:

- Page Impressions
- Page Views
- Page Engagement
- Page Users
- Page Content
- Page Posts
- Page Stories



Import Settings

Once you select desired area, you will be able to proceed to data import.

Select the requested columns and change column types if necessary via [Import settings](#). Once import settings are configured and you can click **Continue** on the top right corner to start data importing.

You can also check [Facebook API](#) for an overview of current Facebook data available for developers.

Configuring Facebook connector

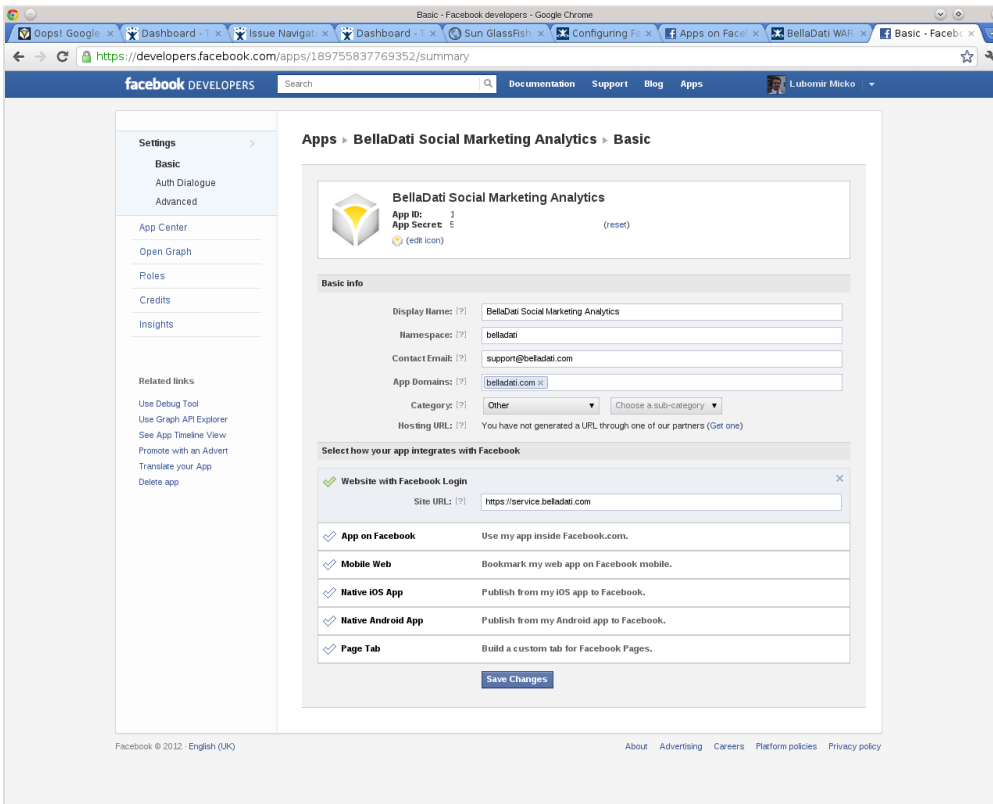


This section is for the OnPremise installation users only. If you are using the Cloud version, the Facebook Connector is ready and you don't need to do any configuration changes.

Creating Facebook Application

To be able to access Facebook Insights (via the Open Graph Protocol), you need to create a [Facebook Application](#). All you need is to configure the application domain and callback URL, which corresponds to the URL, where is your BellaDati instance running.

Here is how it can look like:



Configuring BellaDati

Once you have created an application, you will receive the:

- Application ID
- Application Secret

Example:

This parameters must be defined in the `application.properties` file of you OnPremise installation. To edit application properties:

1. **Login** to BellaDati
2. Select **Settings** from the **Main Menu**
3. Navigate to **Configuration**
4. Scroll to **Facebook** table
5. Click **Edit** in **ApplicationID** row and paste your **ApplicationID**
6. Click **Edit** in **ApplicationSecret** row and paste your **Application Secret**
7. **Restart** BellaDati

BellaDati

Dashboards

Reports

Data sets

Users

Settings

Search...

SalesForce

Property Name	Property Value
ConsumerKey	
ConsumerSecret	

LinkedIn

Property Name	Property Value
API key	
Secret key	

Intuit

Property Name	Property Value
ConsumerKey	
ConsumerSecret	
AppToken	

Facebook

Property Name	Property Value
ApplicationId	
ApplicationSecret	

Twitter

Property Name	Property Value
ConsumerKey	
ConsumerSecret	

HANA

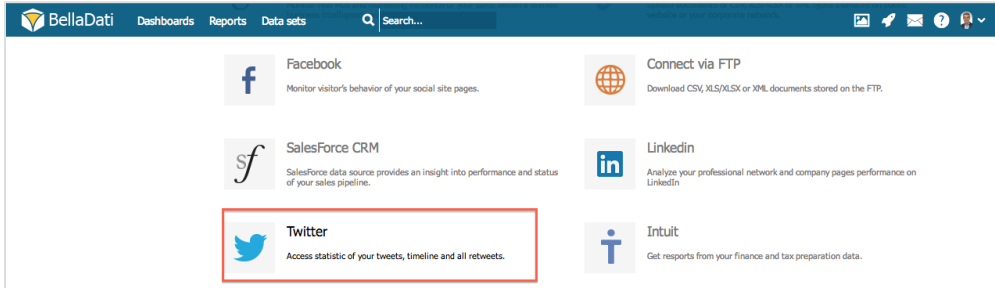
Property Name	Property Value
Host	
Port	
Username	
Password	

Connecting to Twitter

BellaDati allows you to connect and analyze data from Twitter.

In order to connect to a Twitter data source"

- Click **Data sets** from the **Main menu**
- Select **Connect to data sources** at the left menu under **Action** panel.
- Click on the logo of **Twitter** as indicated in the red box below.



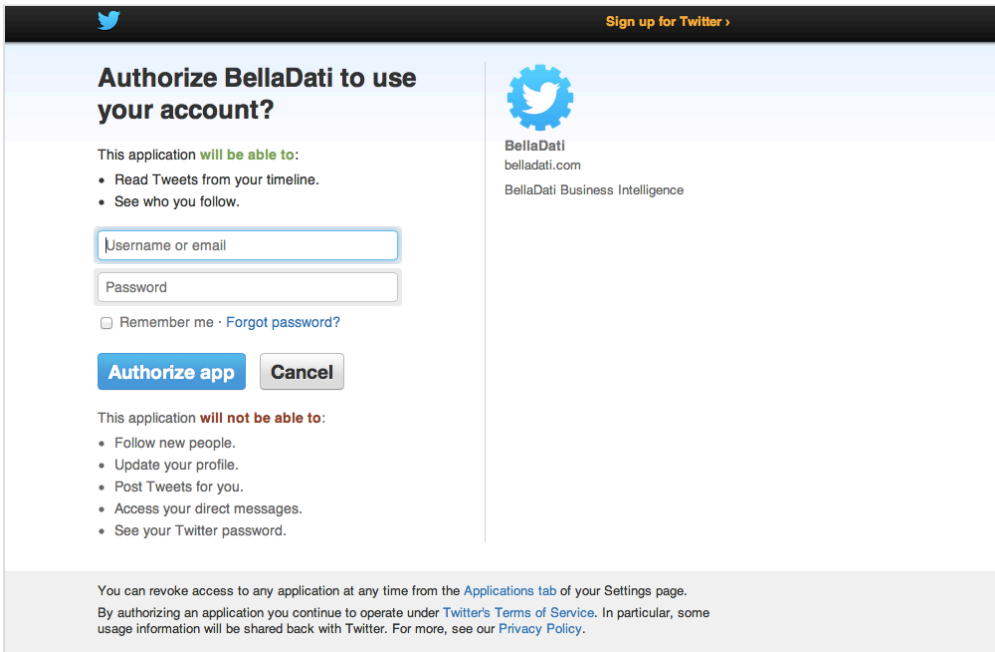
Authentication

Following window will request granting BellaDati access to your Twitter account. Click **Sign in using your Twitter account** to open login screen.

Authorization

You have to authorize BellaDati to access data of your Twitter account.

If you are not logging into your Twitter account yet, the following screen will be shown and you need to enter your Twitter username and password and then click on **Authorize app**.



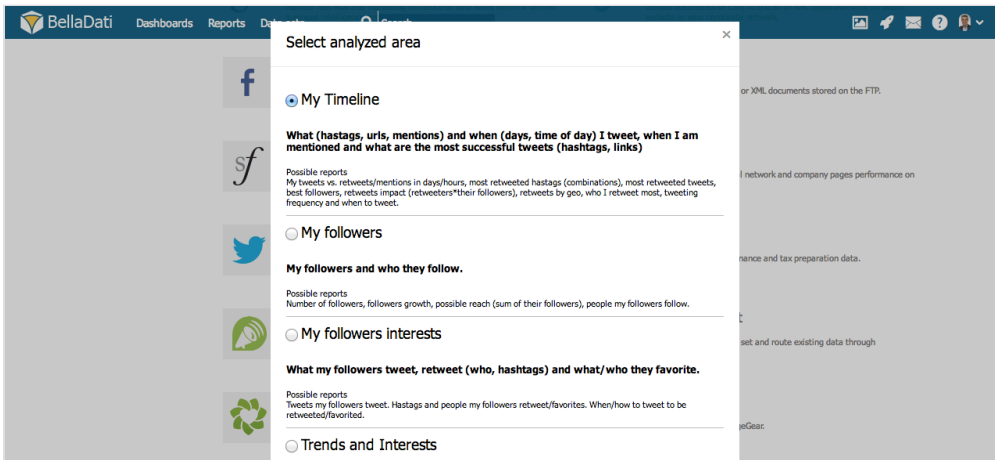
If you are already logged into your Twitter account, click on **Authorize app** directly to proceed with Data set selection.



Aggregated data are stored in BellaDati according to the Twitter API terms of use. Authentication is provided by Twitter servers thus we do not store your login credentials.

Select Data Set

The following page shows all the available areas (data sets) for your analysis.



Select desired area and click **Continue**.



For **Trends and Interests**, you can specify keywords in **Search Query** to obtain trending Tweets. Separate the keywords using space in between as shown in the image below.

BellaDati Dashboards Reports D

Load XML backup
Save XML backup

What my followers tweet, retweet (who, hashtags) and what/who they favorite.

Possible reports
My tweets vs. retweets/mentions in days/hours, most retweeted hashtags (combinations), most retweeted tweets, best followers, retweets impact (retweeters*their followers), retweets by geo, who I retweet most, tweeting frequency and when to tweet.

☐ My followers

My followers and who they follow.

Possible reports
Number of followers, followers growth, possible reach (sum of their followers), people my followers follow.

☐ My followers interests

What my followers tweet, retweet (who, hashtags) and what/who they favorite.

Possible reports
Tweets my followers tweet. Hashtags and people my followers retweet/favorites. When/how to tweet to be retweeted/favorited.

☒ Trends and Interests

Keywords

From

To

Whats intersted for me.

Possible reports
Top influencs with these keywords, who to follow, most retweeted influencers.

Continue

Cancel

Import Data

Once you select desired area, you will be able to proceed to *Import Data*.

Select the requested columns and change column types if necessary via [Import settings](#). Once import settings are configured and you can click **Continue** on the top right corner to start data importing.

You can also check [Twitter REST API](#) for an overview of current Twitter data available for developers.

Configuring Twitter connector



This section is for BellaDati On-Premise users only. If you are using BellaDati in the Cloud, the Twitter connector is ready and you don't need to make any configuration changes.

Creating a Twitter Application

To be able to access Twitter, you need to create a Twitter application.

The screenshot shows the Twitter Developer 'Create an application' page. The browser address bar shows <https://dev.twitter.com/apps/new#>. The page title is 'Create an application'. The form includes the following fields:

- Name:** * BellaDati Demo Application (32 characters max)
- Description:** * Imports Twitter data into BellaDati (10 to 200 characters max)
- Website:** * http://www.belladati.com (Fully-qualified URL)
- Callback URL:** (Where should we return after successfully authenticating?)

Below the form, there is a section titled 'Developer Rules Of The Road'.

Configuring BellaDati

After you have created an application, you can get your consumer key and consumer secret on the next page:

BellaDati Demo Application

Details Settings OAuth tool @Anywhere domains Reset keys Delete

Imports Twitter data into BellaDati
http://www.belladati.com

Organization
Information about the organization or company associated with your application. This information is optional.

Organization None

Organization website None

OAuth settings
Your application's OAuth settings. Keep the "Consumer secret" a secret. This key should never be human-readable in your application.

Access level Read-only
[About the application permission model](#)

Consumer key 0FWdqOkA8mdXg2j1Z1yg

Consumer secret fOCr4dnQBqIhZT3aiIhctdJAf2vbECPj00HVG4u9g

These parameters must be set in the `application.properties` file of your On-Premise installation. To edit application properties:

1. **Login** to BellaDati
2. Select **Settings** from the **Main Menu**
3. Navigate to **Configuration**
4. Scroll to the **Twitter** table
5. Click **Edit** in **ConsumerKey** row and paste your **Consumer key**
6. Click **Edit** in **ConsumerSecret** row and paste your **Consumer secret**
7. **Restart** BellaDati

BellaDati Dashboards Reports Data sets Users Settings Search...

SalesForce

Property Name	Property Value
ConsumerKey	
ConsumerSecret	

LinkedIn

Property Name	Property Value
API key	
Secret key	

Intuit

Property Name	Property Value
ConsumerKey	
ConsumerSecret	
AppToken	

Facebook

Property Name	Property Value
ApplicationId	
ApplicationSecret	

Twitter

Property Name	Property Value
ConsumerKey	
ConsumerSecret	

HANA

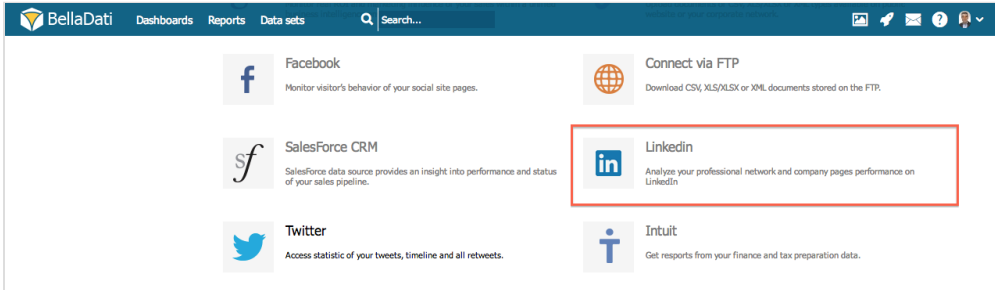
Property Name	Property Value
Host	
Port	
Username	
Password	

Connecting to LinkedIn

BellaDati allows you to connect and analyze data from LinkedIn.

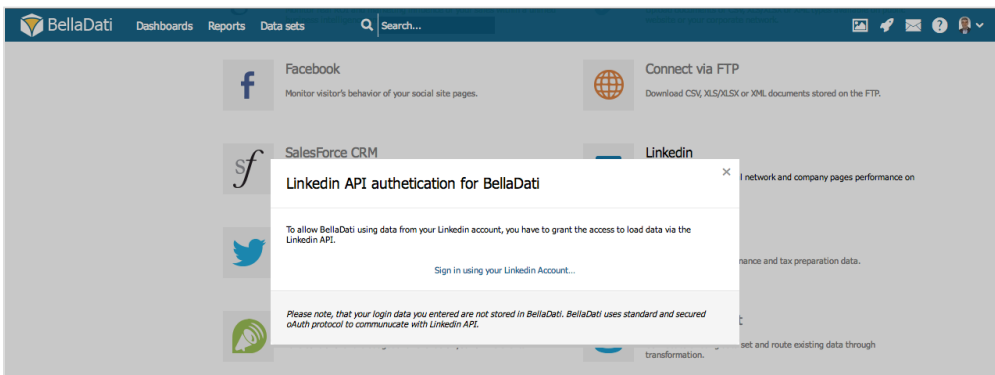
In order to connect to LinkedIn data source:

- Click **Data sets** from the **Main menu**
- Select **Connect to data sources** at the left menu under **Action** panel.
- Click on the logo of **LinkedIn** as indicated in the red box below.



Authentication

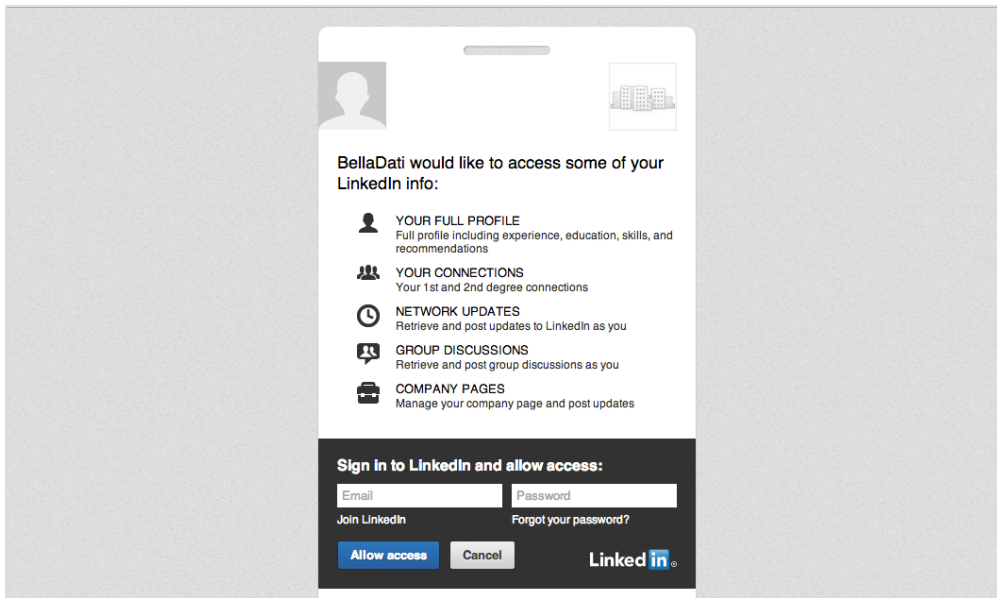
Following window will request granting BellaDati access to your LinkedIn account.
Click **Sign in using your LinkedIn account** to open login screen.



Authorization

You have to authorize BellaDati to access data of your LinkedIn account.

If you are not logging into your LinkedIn account yet, the following screen will be shown and you need to enter your LinkedIn username and password and then click on **Authorize app**.



If you are already logged into your LinkedIn account, click on **Authorize app** directly to proceed with Data set selection.

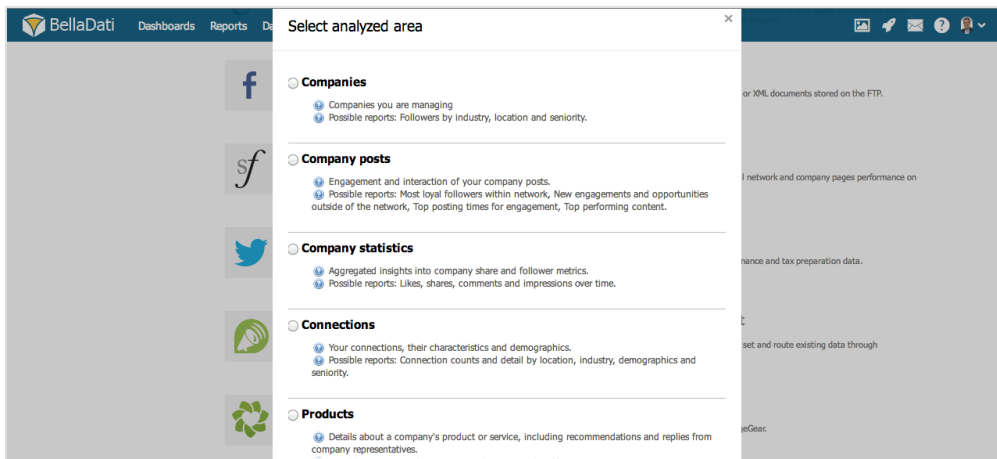


Aggregated data are stored in BellaDati according to the LinkedIn API terms of use. Authentication is provided by LinkedIn servers thus we do not store your login credentials.

Select Data Set

The following page shows all the available areas (data sets) for your analysis.

Select desired area and click **Continue**.



Import Data

Once you select desired area, you will be able to proceed to *Import Data*.

Select the requested columns and change column types if necessary via [Import settings](#). Once import settings are configured and you can click **Continue** on the top right corner to start data importing.

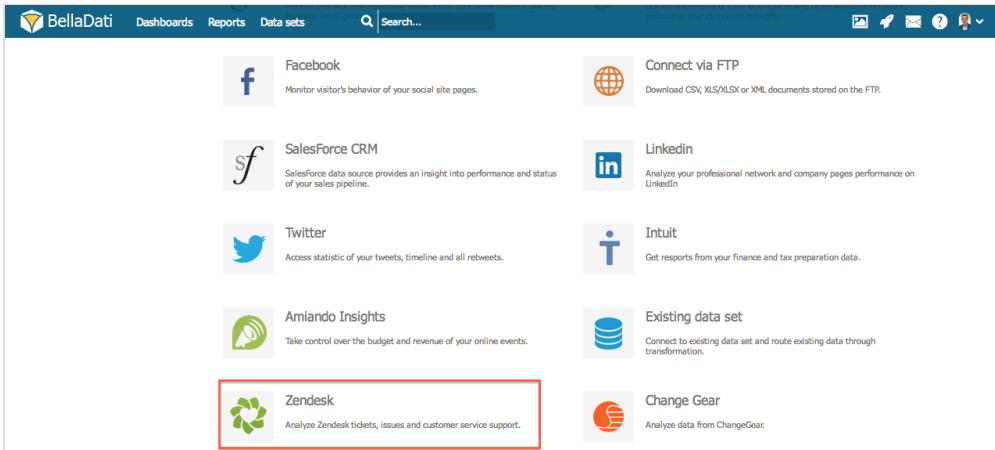
You can also check [LinkedIn REST API](#) for an overview of current LinkedIn data available for developers.

Connecting to Zendesk

BellaDati allows you to connect and analyze data from [Zendesk](#) that offers help desk ticketing, issue tracking, and customer service support.

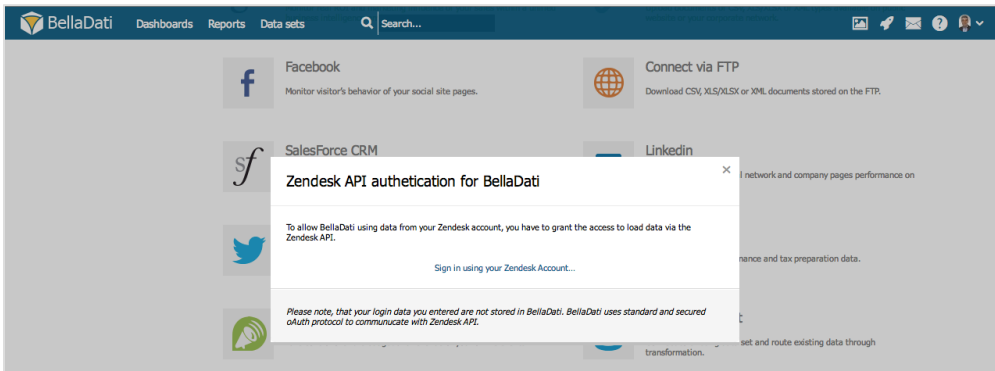
In order to connect to Zendesk data source:

- Click **Data sets** from the **Main menu**.
- Select **Connect to data sources** at the left menu under **Action** panel.
- Click on the logo of **Zendesk** as indicated in the red box below.



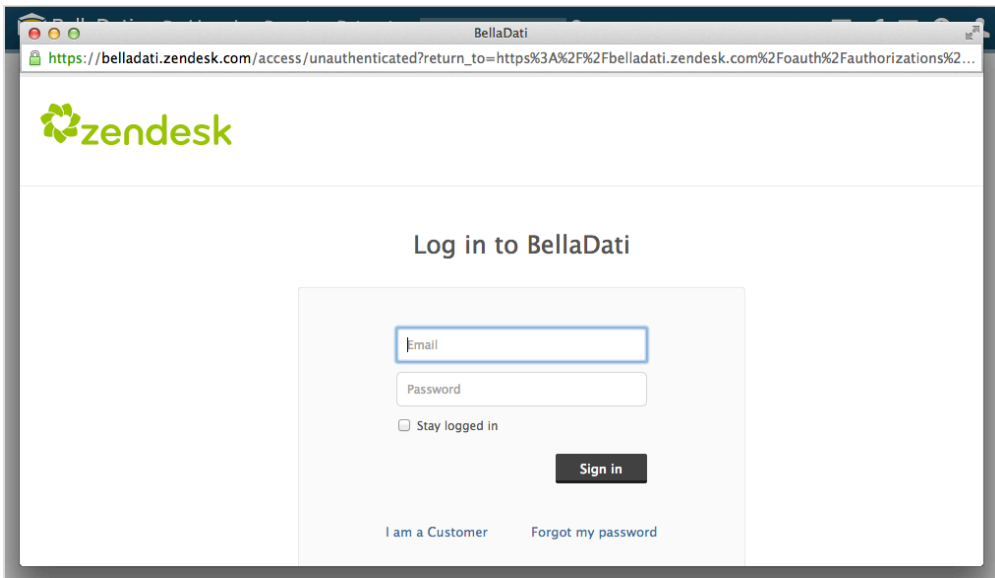
Authentication

Following window will request granting BellaDati access to your Zendesk account. Click **Sign in using your Zendesk account** to open login screen.



Authorization

Login in to your Zendesk account.



Aggregated data are stored in BellaDati according to the Zendesk API terms of use. Authentication is provided by Zendesk servers thus we do not store your login credentials.

Select Resource

The following page shows all the available resources (data sets) you can connect to. It includes:

- Tickets
- Users
- Groups
- Topics
- Ticket Metrics
- Forums
- Categories
- Organizations
- Satisfaction Rating

Select desired area and click **Continue**.

Import Data

Once you select desired area, you will be able to proceed to *Import Data*.

Select the requested columns and change column types if necessary via [Import settings](#). Once import settings are configured and you can click **Continue** on the top right corner to start data importing.

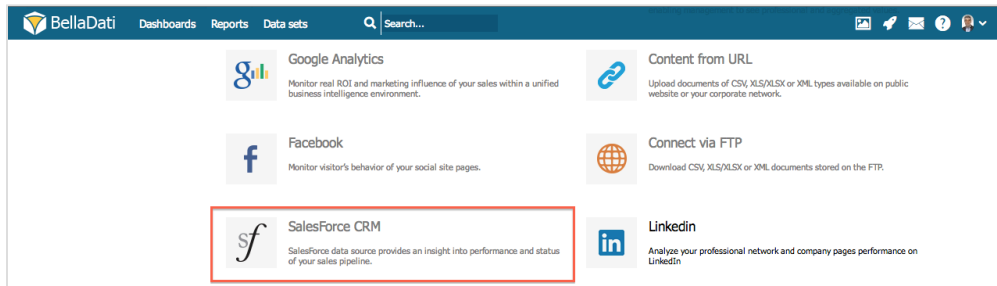
You can also check [Zendesk REST API](#) for an overview of current LinkedIn data available for developers.

Connecting to Salesforce

BellaDati can be connected to Salesforce datasource.

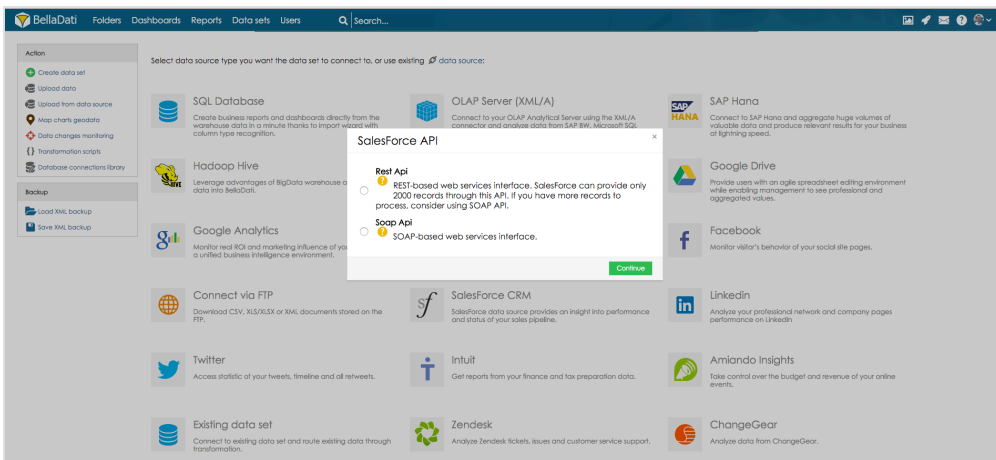
Connecting to Salesforce

From **Data Source Connectors** page select **Salesforce CRM**.



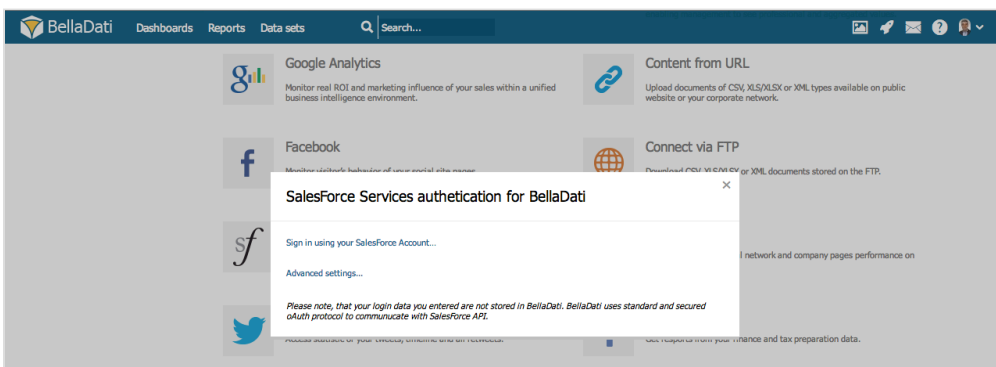
Select Salesforce API

Select which of the APIs will be used. Process of import is the same for both options. SOAP API option supports import of more than 2000 rows.



Authentication

Login to Salesforce and grant access to data stored there.



The login data you enter are stored encrypted and not shared with any other application. This login and password is used only to enable data transfer from secured Salesforce API and is not allowed to extract any other personal information from your Salesforce account.

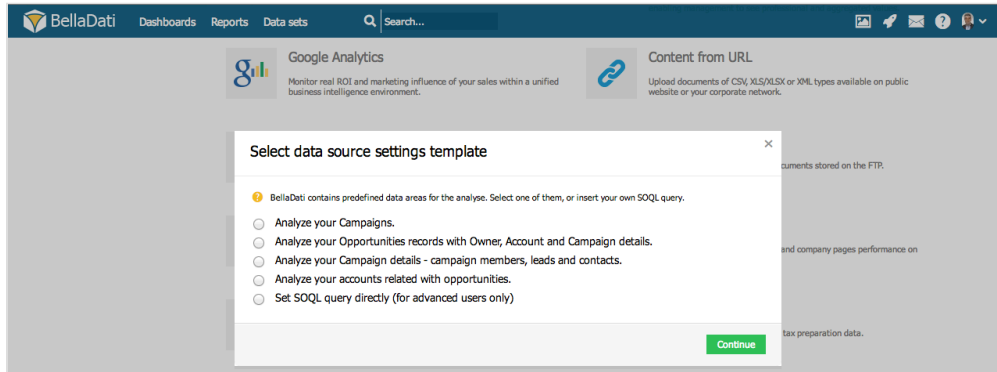
Data Area Selection

Select predefined Salesforce areas or create general SOQL query:

Predefined objects include:

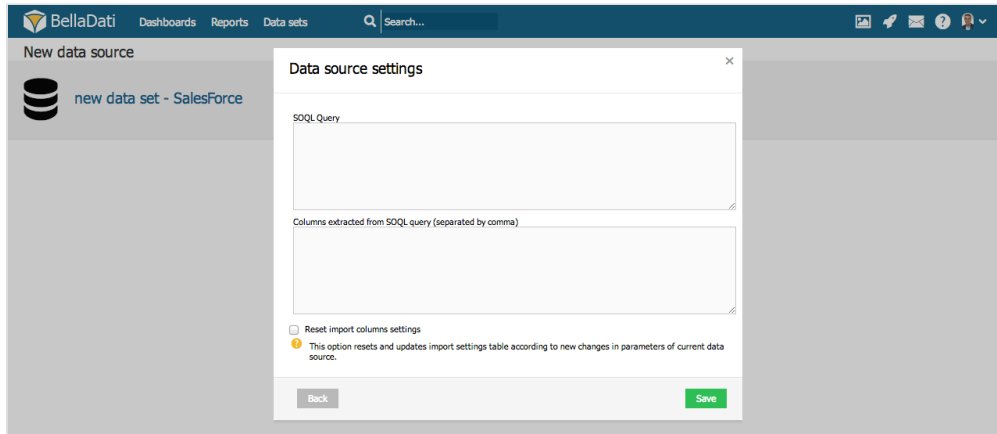
- Campaigns
- Opportunities records with Owner, Account and Campaign details
- Campaign details - campaign members, leads and contacts
- Accounts related with opportunities

To create your own objects, continue to [SOQL](#) Salesforce documentation (for advanced users only)



Extracting SOQL Columns

You may require only a subset of columns returned by executed SOQL query. This function allows you to define which columns will be finally imported to BellaDati's data warehouse.



Modifications

Following specific data source parameters can be modified within the existing data source via data set summary:

- SOQL query
- Extracted SOQL columns
- **SalesForce authentication revoke:** You can terminate BellaDati's access to your data in Salesforce.

BellaDati

Dashboards

Reports

Data sets

Search...

Settings

Data set summary

Attributes (9)

Drill down paths (0)

Indicators (11)

Joined data sets (0)

Data

Import Data

Data source

Browse data

Create alarm

Erase data

new data set - Salesforce - Data source

SalesForce

Add

Import data

Check availability

Import settings

Schedule

Notifications

Delete

Basic info

Data source name

SalesForce

Data source type

SalesForce

Data source imports overview

Executed when

Periodicity

Jun 11, 2014 3:25:51 AM

Data source settings

SQL Query:

SELECT Owner.Name, Owner.UserRole.Name, Owner.Department, Owner.Division, Campaign.Id, Campaign.Name, Campaign.Type, Campaign.BudgetedCost, Campaign.ActualCost, Campaign.ExpectedResponse, Campaign.NumberSent, ...

Columns extracted from SQL query (separated by comma):

Owner.Name,Owner.UserRole.Name,Owner.Department,O
wner.Division,Id,Name,Type,BudgetedCost,ActualCost,Ex
pectedResponse,NumberSent,NumberOfLeads,NumberOf
ConvertedLeads,NumberOfContacts,NumberOfResponses,
NumberOfOpportunities,NumberOfWonOpportunities,Am

BellaDati Documentation, Version 2.7.11.5

Page 80 of 370

Configuring Salesforce connector

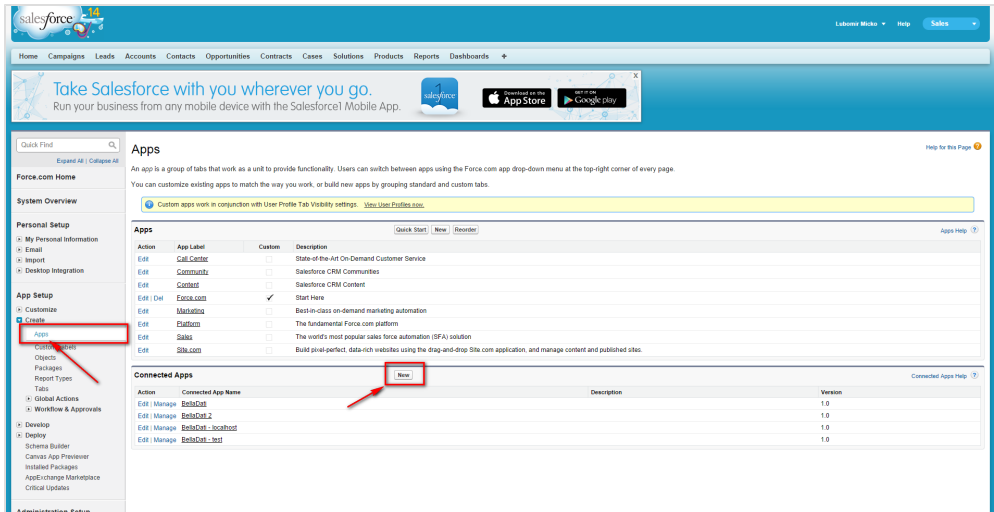


This section is for the OnPremise installation users only. If you are using the Cloud version, the Salesforce Connector is ready and you don't need to do any configuration changes.

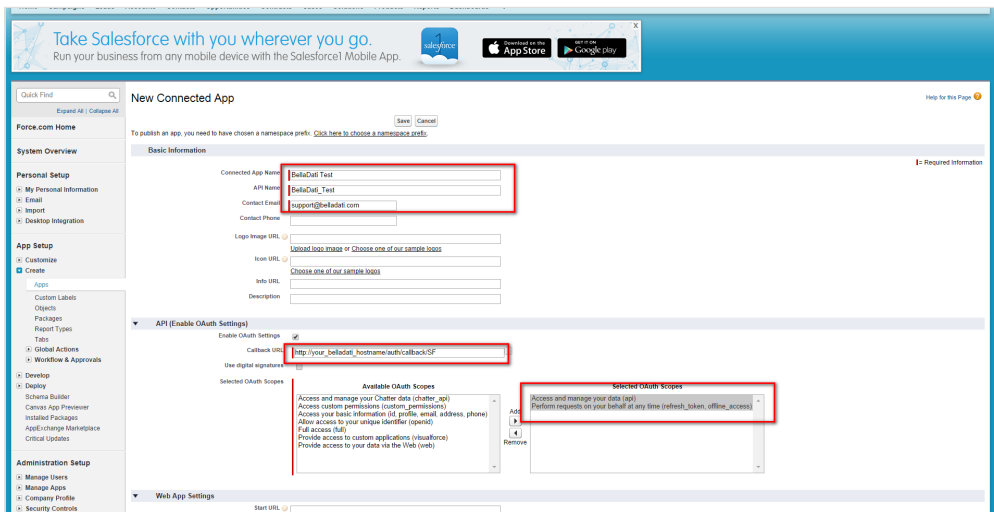
To be able to access the Salesforce API, you need to create new [connected Salesforce Application](#). All you need is to configure the application callback URL, which corresponds to the URL, where your BellaDati instance is running. For example

Step-by-step Salesforce configuration

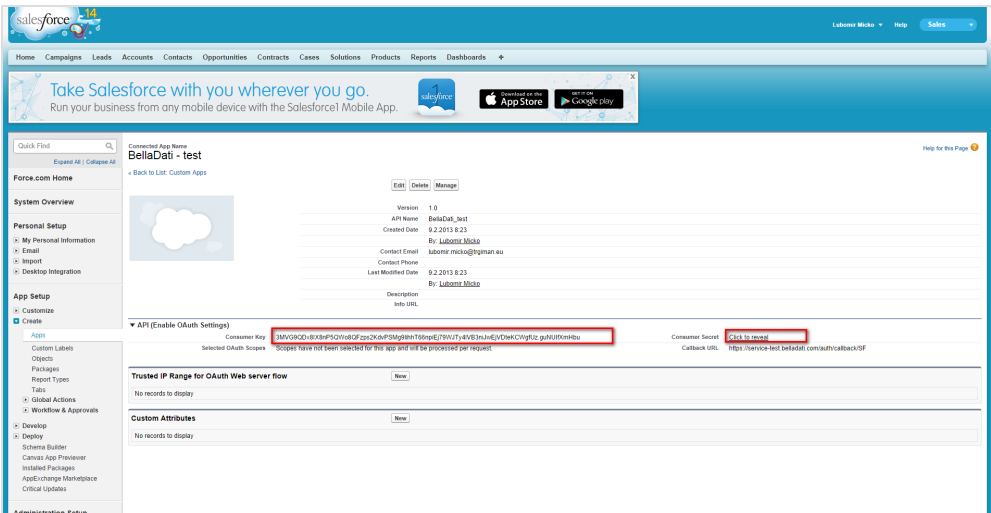
1. Login to Salesforce, navigate to Setup and create new Connected application



2. Enter application details, callback URL (e.g. `http://your_host/auth/callback/SF`) and select the `api` and `refresh_token` permission scope.

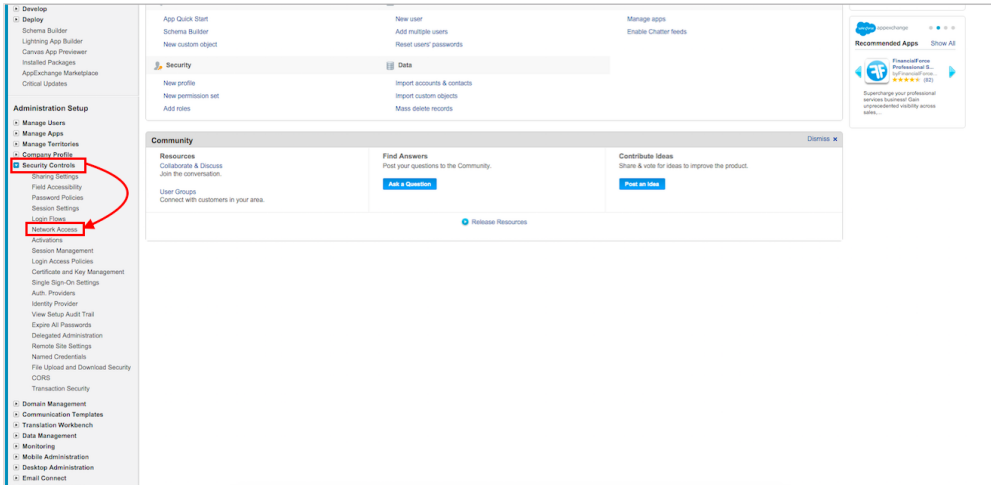


3. Get consumer key/secret and enter it into BellaDati

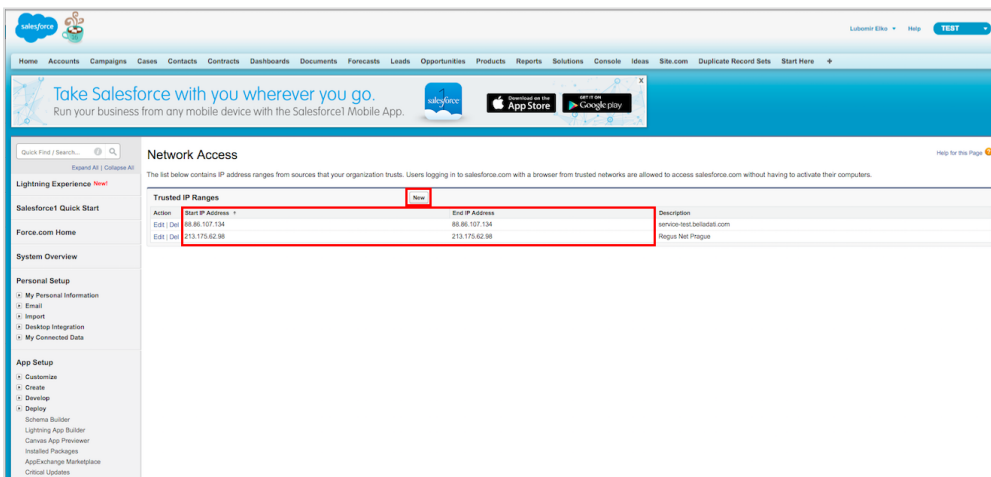


Allow SOAP API login

In case you want to use SOAP API login, BellaDati IP address of your environment has to be in white list, go to salesforce.com -> go to your Account -> select Security Controls -> select Network Access



Add your IP address



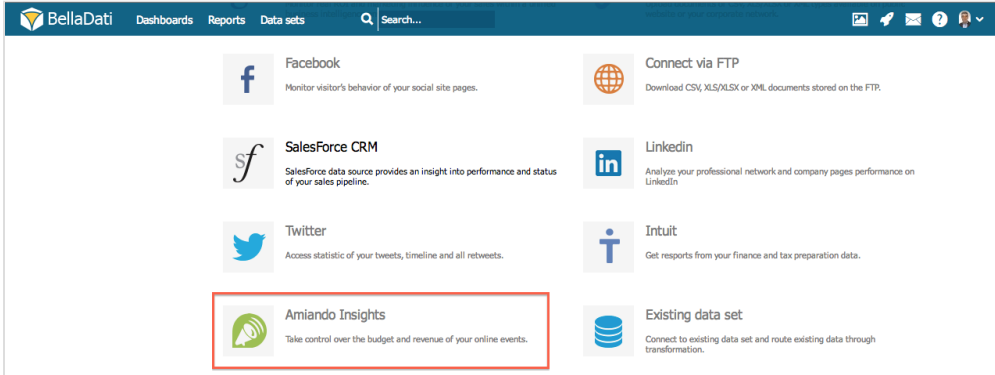
Please note, that changing Salesforce configuration may take same time to be effective.

Once you have created an application, you will receive the `Consumer key` and `Consumer Secret`. Now you have to enter these parameters into BellaDati. Log-in as domain administrator and navigate to **Settings -> Configuration** page. See [Managing Configuration](#).

Connecting to Amiando

To import data from Amiando, perform the following steps:

1. [Get Amiando API key](#) and enter it in BellaDati.
2. Choose which columns do you want to import in [import settings](#).



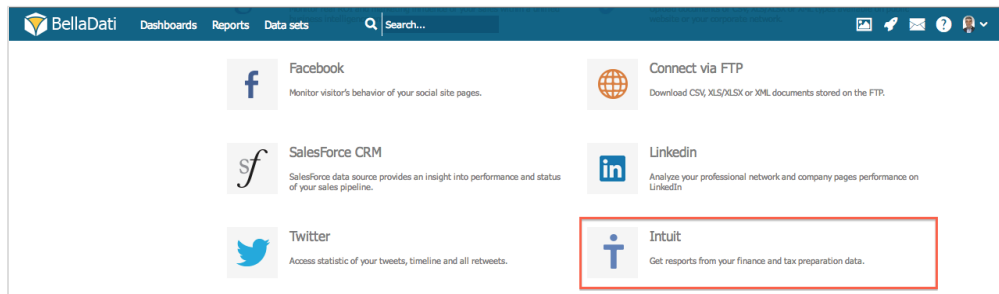
BellaDati selects predefined columns from Amiando only. Please contact our [support](#) for options how to obtain another data from Amiando.

Specific Modifications

- API key change
- Import settings: Reset import columns settings

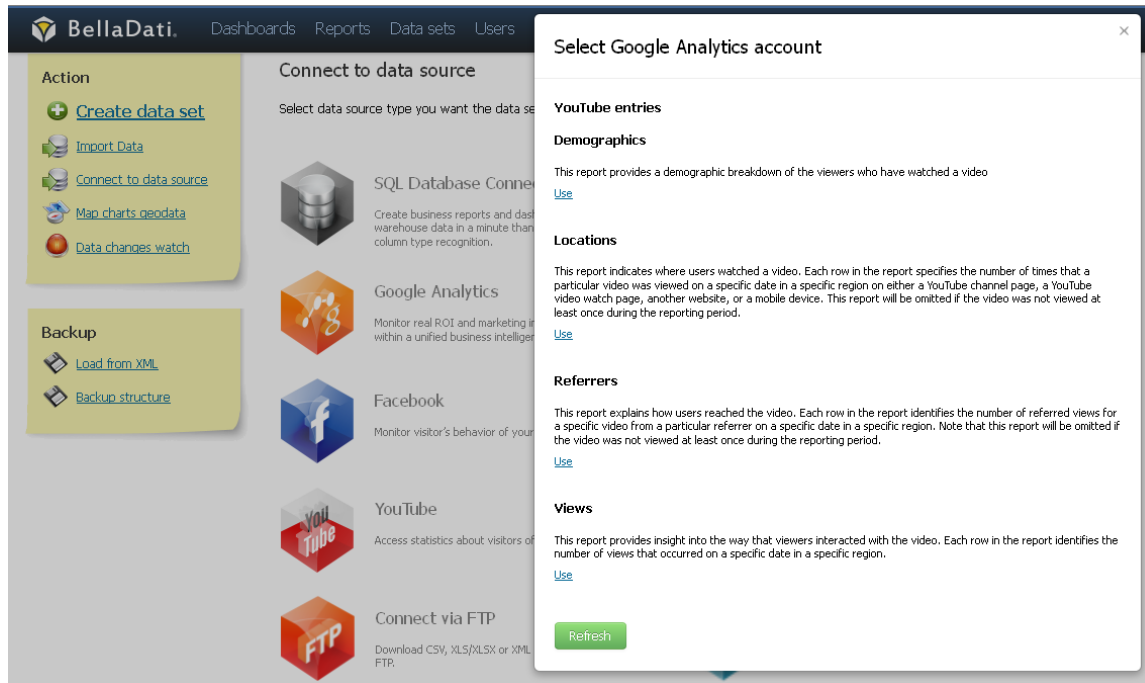
Connecting to Intuit

BellaDati allows you to load your data from Intuit Quickbooks accounting tool.



Connecting to YouTube

BellaDati provides connector to YouTube.



Authentication

You have to grant BellaDati access to your data on YouTube first. Please log in using your Google account credentials.



The login data you entered are stored encrypted and not shared with any other application. This login and password is used only to enable data transfer from secured Google API and is not allowed to extract any other personal information from your Google Account.

YouTube Entries Selection

Now select one of the YouTube entries (data areas):

- **Demographics:** This report provides a demographic breakdown of the viewers who have watched a video.
- **Locations:** This report indicates where users watched a video. Each row in the report specifies the number of times that a particular video was viewed on a specific date in a specific region on either a YouTube channel page, a YouTube video watch page, another website, or a mobile device. This report will be omitted if the video was not viewed at least once during the reporting period.
- **Referrers:** This report explains how users reached the video. Each row in the report identifies the number of referred views for a specific video from a particular referrer on a specific date in a specific region. Note that this report will be omitted if the video was not viewed at least once during the reporting period.
- **Views:** This report provides insight into the way that viewers interacted with the video. Each row in the report identifies the number of views that occurred on a specific date in a specific region.

Then continue [Import settings](#).

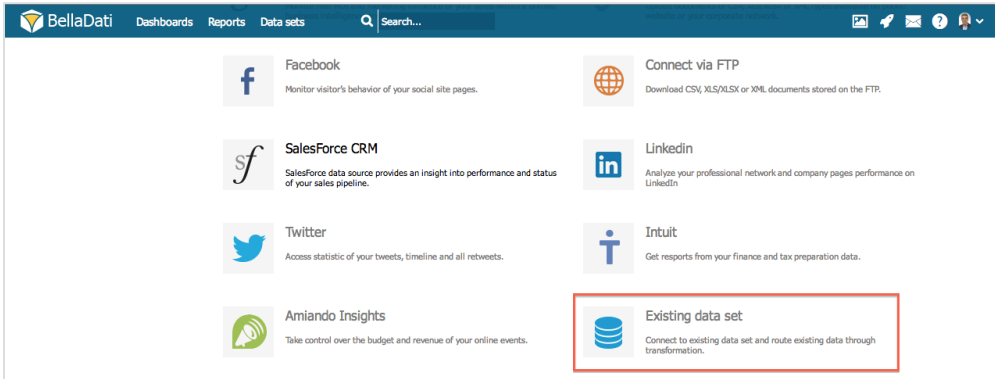
Modifications

The following parameters and actions can be performed for the existing YouTube data source:

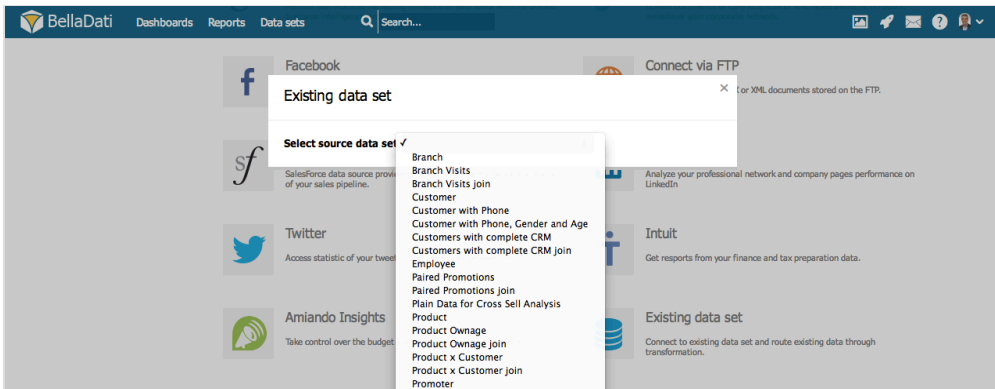
- Revoke access to Google services

Connecting to Existing Data Set

You can connect to existing **Data Set** by selecting **Existing Data Set** connector in *Data Sources* window.

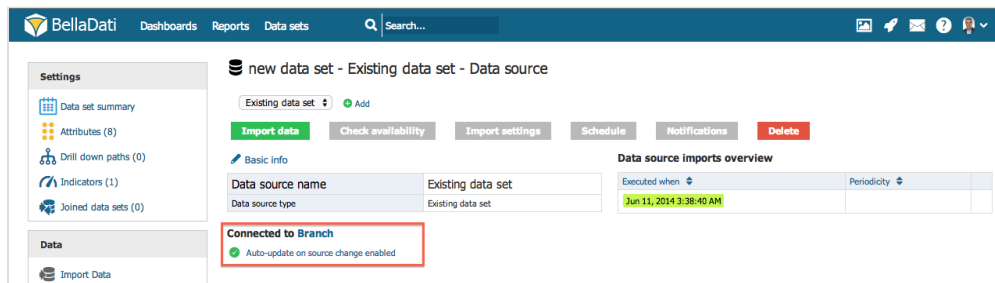


Select desired **Data Set** from the list of all available tables in the domain.



Auto Update on Source Change

In order to ensure that Data Set will be automatically updated after change in its source Data Set, click on **Auto-update on source change enabled** link in *Data Source* window of appropriate **Data Set**.



Connecting to Hadoop

BellaDati can access the Hadoop File System (HDFS) over the Hadoop Hive, which is one of the supported JDBC SQL databases. Refer to [Connecting to SQL Database](#).

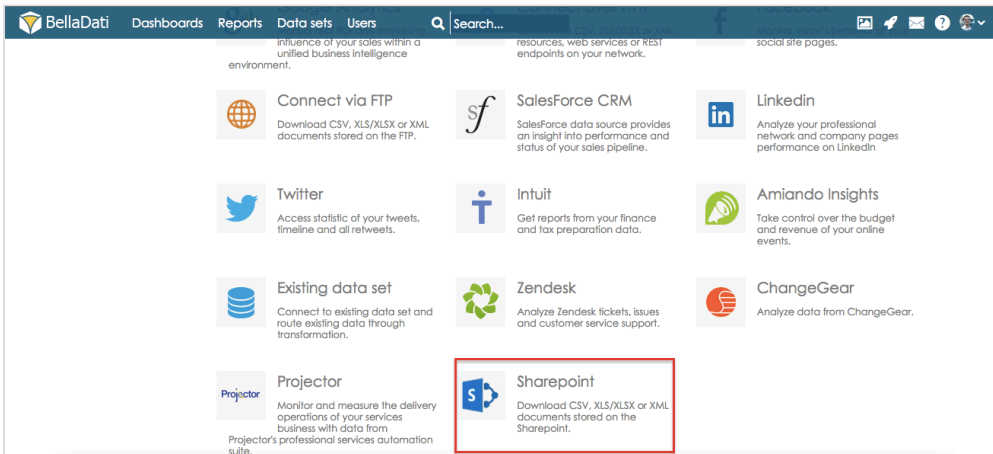
Connecting to MS SharePoint

BellaDati can import data from MS SharePoint.

Connecting to MS SharePoint has similarities with [file import](#). In addition imports from SharePoint can be scheduled to execute automatically and repeatedly.

Connecting to MS SharePoint

From **Data Source Connectors** page select **Sharepoint**.



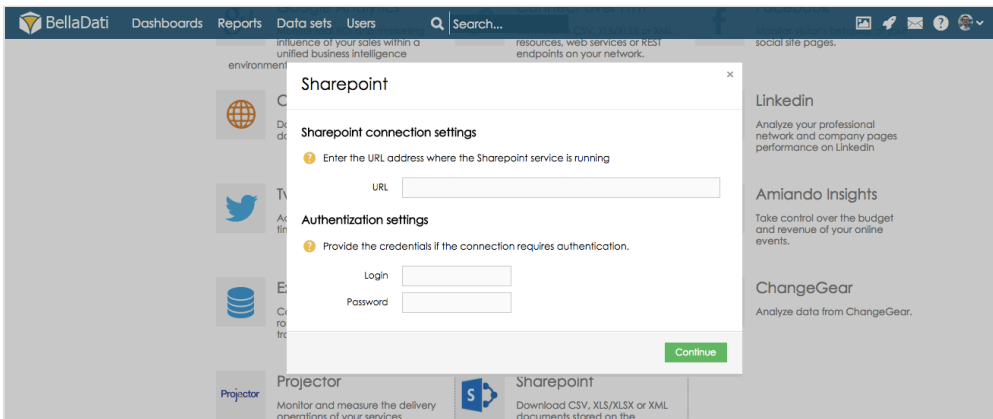
Connection parameters

Enter following parameters to download files stored on the Sharepoint:

URL: SharePoint server address (format of server address must be like: https://server.com/my_site_id/_api/my_context/, where my_context is site collection or another context)

Login

Password

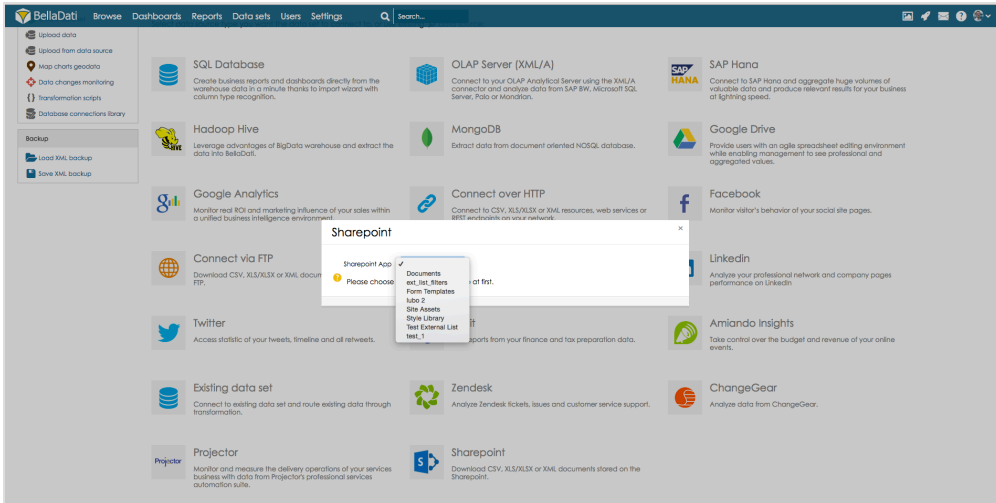


Select source

Select from which source data should be imported. There are following options available:

- *Document library*
- *External list*
- *Custom list*
- *Calendar list*
- *Task list*

If you request other type of SharePoint list, contact us on support@belladati.com.

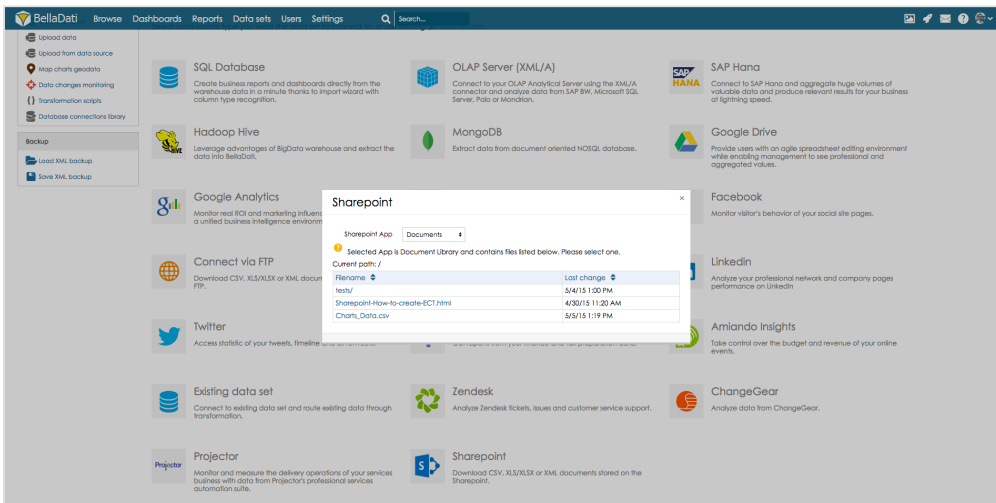


Importing documents

Select SharePoint app called Documents.

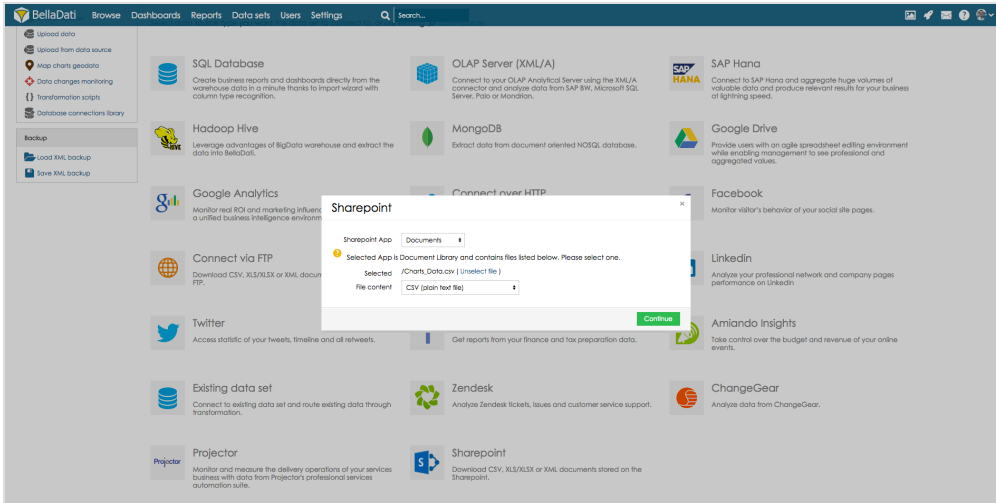
Selecting file


BellaDati will display SharePoint files structure. Select file you want to import.



Select file content

Select file format. Continue to [file import settings](#) to learn more about available file types.



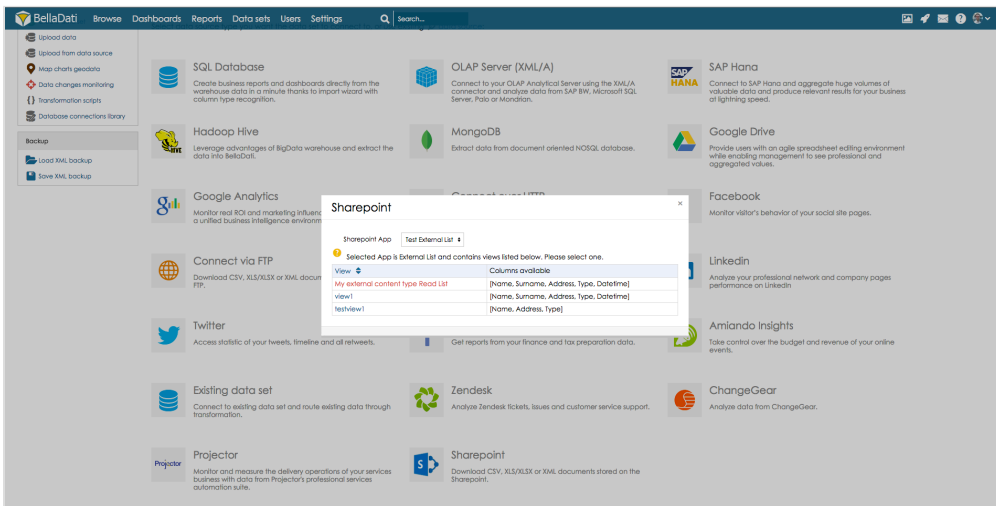
 Scheduled imports can be set up for files imported from SharePoint. Continue to [Scheduling Import](#) to learn more.

Importing data from external lists

Select one of the external list available in your SharePoint environment.


Select view from selected list

Selecting view from selected external list will redirect you to import settings page.

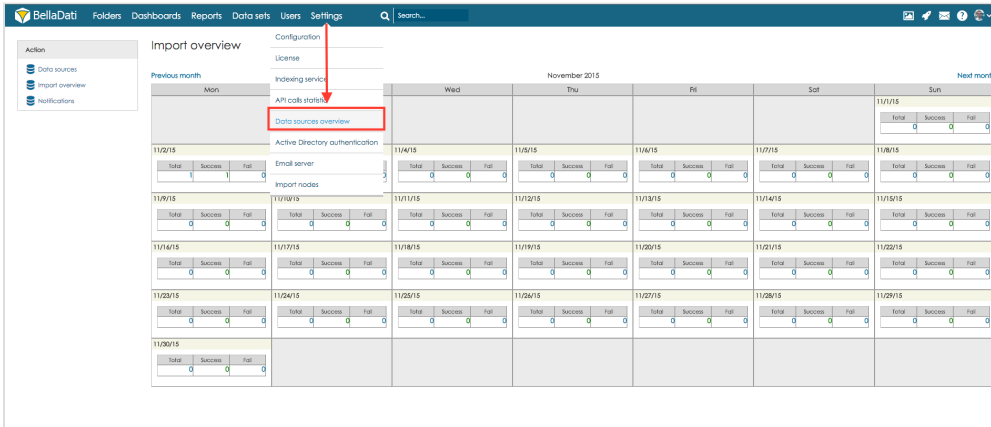


Data source overview

Data sources overview serves to provide monitoring mechanism of all imports and data sources connections.

 Data sources overview is available only for domain administrator (single domain installation) and global administrator (multi-domain installation).

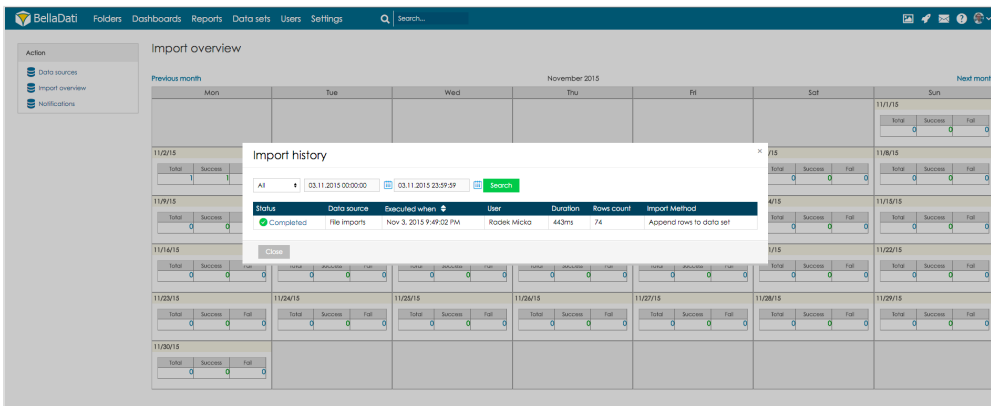
In order to access Data sources overview, hover your mouse over tom menu item *Settings* and select option *Data sources overview*.



This page provide you complete overview of import history - detailed number of run imports, successful imports and failed imports for each day. To view such a details hit a click on value for selected day in the calendar. Details are displayed in dialogue window with following details:

- Import **Status**
- **Data source** the import is based on
- Time when import has been **executed**
- Name of **user** who scheduled this import
- Import **duration**
- Number of imported **rows**
- Used **import method**

You can also use filtering options to select which status of the import will be displayed (All, Success, Failure, Cancelled, Deleted), and define interval for which the results will be displayed.



Data sources

Click button *Data sources* in section *Action* to open page with all data source connections details:

- **Name** of the data source
- **Domain** to which data source relates to
- **Data set** related to the data source
- **Data source type**
- **Next run** when import will be executed

BellaDati Folders Dashboards Reports Data sets Users Settings Search...				
Data sources				
Domain				
Name	Domain	Data set	Data source type	Next run
Existing data set	Radek - Localhost	Paired Promotions	DataSetImportable	
Twitter	Radek - Localhost	Twitter	TwitterImportable	
Existing data set	Radek - Localhost	Plain Data for Cross Sell Analysis	DataSetImportable	
SQL 数据库	Radek - Localhost	D_ORGAN	SQLImportable	
SQL 数据库	Radek - Localhost	Policies (MYSQL, JFPI)	SQLImportable	
SQL 数据库	Radek - Localhost	D_SALES_CHANNEL	SQLImportable	
SQL 数据库	Radek - Localhost	D_BANK_ORG	SQLImportable	
SQL 数据库	Radek - Localhost	FA_PLAN_POLICYCHANGE	SQLImportable	
SQL 数据库	Radek - Localhost	Policies (MYSQL, JFPI)	SQLImportable	
SQL 数据库	Radek - Localhost	D_SALES_CHANNEL_pu2u	SQLImportable	
SQL 数据库	Radek - Localhost	FA_PLAN_POLICYCHANGE_OwRL	SQLImportable	
SQL 数据库	Radek - Localhost	D_ORGAN_Judge	SQLImportable	
SQL 数据库	Radek - Localhost	D_BANK_ORG_Totals	SQLImportable	
SQL 数据库	Radek - Localhost	PHILIPS_building_controllers	SQLImportable	
SQL 数据库	Radek - Localhost	PHILIPS_event_enrollments	SQLImportable	
SQL 数据库	Radek - Localhost	PHILIPS_trends	SQLImportable	
SQL 数据库	Radek - Localhost	PHILIPS_datasources	SQLImportable	
FTP	Radek - Localhost	Gonit Test - Current	CSVImportable	
Group by Week	Radek - Localhost	AAK-GroupWeek	ProjectorImportable	
Project Listing	Radek - Localhost	AAK-ProjectList	ProjectorImportable	
Projector	Radek - Localhost	New data set - Projector	ProjectorImportable	
FTP	Radek - Localhost	New data set - FTP	XLImportable	
Projector	Radek - Localhost	New data set - Projector 1	ProjectorImportable	
FTP	Radek - Localhost	New data set - FTP 1	CSVImportable	
FTP	Radek - Localhost	EAM - Application Support	CSVImportable	

Notifications

Click button *Notifications* in section *Action* to open dialogue window with notification settings where can be set for which import results and to which users email notification will be sent

- Possible import results:
 - Completed
 - Completed with errors
 - Error
 - You don't have enough space in your data warehouse
 - Wrong data source configuration

BellaDati Folders Dashboards Reports Data sets Users Settings Search...				
Import overview				
November 2015				
Mon	Tue	Wed	Thu	Fri
				Sat
				Sun
11/2/15	11/9/15	11/7/15	11/7/15	11/8/15
Total Success Fail	Total Success Fail	Total Success Fail	Total Success Fail	Total Success Fail
0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
11/9/15	11/16/15	11/14/15	11/14/15	11/15/15
Total Success Fail	Total Success Fail	Total Success Fail	Total Success Fail	Total Success Fail
0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
11/14/15	11/21/15	11/21/15	11/21/15	11/22/15
Total Success Fail	Total Success Fail	Total Success Fail	Total Success Fail	Total Success Fail
0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
11/22/15	11/24/15	11/24/15	11/24/15	11/29/15
Total Success Fail	Total Success Fail	Total Success Fail	Total Success Fail	Total Success Fail
0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
11/30/15				
Total Success Fail				
0 0 0				

Import Settings

Import settings page allows you to control **ETL** (Extract-Transform-Load) operations and verify the structure of data being imported. The main task is to define the mapping of tabular data to [attributes](#), [indicators](#) and date/time dimensions.

Following adjustments and actions are available:

- **First row is header:** Use the texts in the first row as names for corresponding columns; *only for plain text (clipboard), CSV or Excel*
- **Excluded rows:** Allows you to exclude some rows from the beginning of the file imported (eg. additional information, not data).; *only for plain text (clipboard), CSV or Excel*
- **Encoding:** Select appropriate encoding for the source file (UTF-8, ISO-8859-1, Win-1250, Win-1252, Auto are available); *only for plain text, CSV or XML*
- **Separator:** Auto-detection (the most frequent separator is the semicolon ";"), otherwise select character that separates each column (comma, tab, semicolon, space, vertical bar, custom); *only for plain text (clipboard) and CSV*
- **Fill the empty cells:** Generally for the whole import, or this substitution can be performed individually for particular columns.
- **Apply import template:** See "import templates" chapter below.
- **Use default settings:** Resets all import settings to defaults.
- **Get new columns:** Gets new columns from updated data source - when there is existing import template, newly added columns from data set don't exist in template from previous import)

Another functions are:

- Data cleaning and transforming using the transformation scripting
- Assigning imported columns to existing attributes or indicators
- **Renaming columns**
- **Column merging**
- Adding new columns
- **Preview changes**



Automatic encoding detection is not always reliable. We recommend to check for strange characters in the preview.

BellaDati Dashboards Reports Data sets Search...

New data source

new data set - SQL database

Edit SQL query
Data source settings
Fill the empty cells... Apply import template... Use default settings

Continue

Columns available

Search: Search

All (14)

Attribute (5)
Indicator (5)
Translation (0)
Time (2)
Exclude (0)

Proceed action...

6 L purchase type
7 L time of transaction
8 L trans lat
9 L trans location
10 L trans long
11 M lat
12 M long
13 M value
14 Transaction id

New column

Preview selected

Data preview

1	2	3	4	5	6	7
Import id	L card	L customer id	L date of transaction	L id	L purchase type	L time
194062	6.71348811876E12	161	2014-01-29	175	17	12:12:30
194062	7.314451898801E12	16	2014-01-29	301	9	14:32:08
194062	3.621750649118E12	49	2014-01-30	737	1	21:57:05
194062	8.83551001489E11	36	2014-01-30	1345	8	15:46:44
194062	8.53265279497E12	57	2014-01-30	1772	2	17:23:23
194062	1.147094135026E13	103	2014-01-29	1852	19	19:37:24
194062	1.114929819050E13	107	2014-01-30	1895	0	10:06:15
194062	1.009795251070E13	197	2014-01-30	2079	3	15:12:53
194062	1.202838143466E13	6	2014-01-30	2326	2	17:33:10
194062	3.10166239056E12	11	2014-01-29	2504	8	17:49:09
194062	9.050207543128E12	11	2014-01-30	2695	7	20:55:24
194062	4.301145664813E12	33	2014-01-30	3064	9	12:31:53
194062	3.3059949563E12	137	2014-01-30	3136	15	11:36:10
194062	1.018075804832E13	177	2014-01-29	3257	20	17:11:52
194062	8.49496759297E12	129	2014-01-29	3266	13	11:16:49
194062	1.08719466861E12	182	2014-01-29	3788	10	19:34:48
194062	1.0841912148E11	5	2014-01-29	4281	1	21:27:54
194062	6.330633479656E12	145	2014-01-29	4338	11	19:53:55
194062	3.426421171134E12	32	2014-01-30	4399	18	08:13:28

Availability of the adjustments on the screenshot above could vary depending on file format imported. Options are stated for **manual import**, see [data sources](#) for specific information about automated imports.

Column Settings

ETL name - it is the name of the column as it is coming with the source data (e.g. SQL column name, CSV header row column, etc). ETL name is not affected by renaming the column at the import settings page. The purpose of storing the ETL name is simple - BellaDati can easily map the incoming data on the existing objects (attribute/indicator) - e.g. when you change the order of columns in your data, but the column names remain the same, BellaDati will map the column onto the right object.

If you want to change the type of particular column, click on the name of the selected column in the list of columns (in the left side of the import screen). It's also possible to change meaning of more columns to one type in just one step - just mark selected columns by clicking in the checkboxes next to them and then select their meaning from menu above.

There are eight possible meanings of columns:

Date/Time (Separate) - time index of particular rows. It can be displayed in a lot of different time formats (also depending on language - for more information see the related part of this chapter). You can choose multiple date/time columns in single import.

Datetime - datetime index of particular rows. It can be displayed in a lot of different datetime formats (also depending on language - for more information see the related part of this chapter). You can choose multiple datetime columns in single import.

Long text - defines long text - description. This type of column can be used only in statistic tables only. This type of column cannot be used in aggregations.

Attribute - defines categories of the drill-down path. It's usually a short text (e.g. affiliate, product, customer, employee, division etc.). Every attribute column creates exactly one attribute in the data set. Those attributes can be freely combined in the drill-down paths.

GEO Point - you can map the latitude/longitude onto the GEO point attribute type. This attribute can be then used in Geo Map view type to plot data into its particular location.

GEO JSON - you can map shape onto the GEO JSON attribute type. This attribute can be then used in Geo Map view type to plot data into its particular location displayed as shape.

Translation - defines language translation of other column identified as Attribute

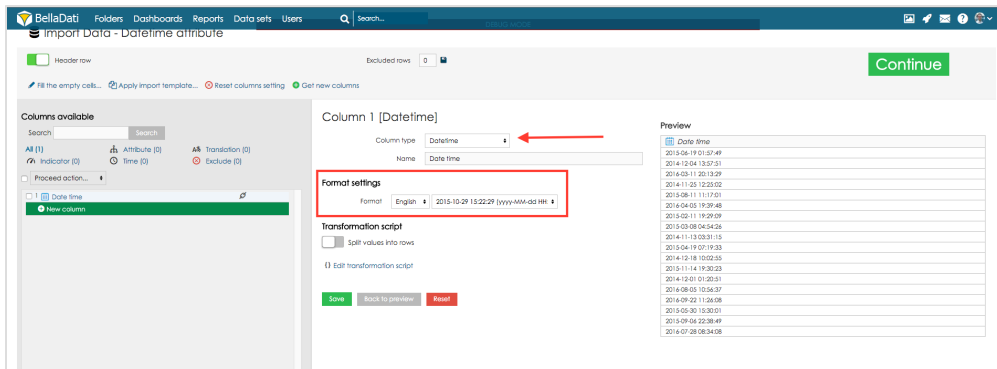
Indicator - indicators are usually the numeric data, which are the main point of the user's interest.

Don't import - these columns won't be imported at all (it's useful if column contains no, invalid or unimportant data).

Preview of marked columns may be displayed by clicking on the "Preview selected" button. In this way you can get a better view into your data and their meaning settings. If your data contain too much columns, you can use a search label above the column list to find appropriate column and check its settings. Under this searching field is displayed statistics, which shows number of particular types of columns.

Datetime

If your source data contains Datetime values, you can map them to **Datetime attribute**. This single column will contain both, date and time, e.g. *Apr 2014 10:43:43 AM*. See the following example:



Datetime Format

Every time column has a specific type of format. This format should be automatically detected during import. However it's possible, that you have your time data in some very specific format. In this case you can use the list of available format in different languages.

If you don't choose from available formats, you can also define your own specific custom format for your data. In this case, you should choose your language from the list below and enter a code, which describes your data format according to these meanings (note, that the number of characters influences the interpretation of the code):

Code	Meaning	Number of characters in code
y	Year	Two characters (yy) represents two digits year number (89). Otherwise is the code interpreted as four digits year number (1989).
M	Month in year	Three or more characters (MMM) are interpreted as text representation of month (e.g. "January" or "Jan"). In other cases are characters interpreted as number of month in year (1-12).
d	Day in month	Number of characters (d) in code should be equal to minimal number of digits in source data. It's always a numeric format.
E	Day in week	Number of character determines, if the day is displayed in its full name (EEEE - "Monday") or in its shortcut (EE - "Mo").
H	Hour in day	Number of characters (H) in code should be equal to minimal number of digits in source data. It's always a numeric format.
m	Minute in hour	Number of characters (m) in code should be equal to minimal number of digits in source data. It's always a numeric format.
s	Second in minute	Number of characters (s) in code should be equal to minimal number of digits in source data. It's always a numeric format.

Separator character should be equal to separator character contained in source data (space, dot, semicolon, etc.). If your source data contains time in more separated columns (months, days, years), it's necessary to merge those columns first (described in previous part of this chapter). Next table shows some combination of source data and appropriate time code.

Source data	Appropriate code
09/15/10	MM.dd.yy
26/03/1984	dd/MM/yyyy
15.September 2010	dd.MMMM yyyy
15 Sep 10	dd MMM yy
Wed 15 09 10	EE dd MM yy
Sep 15, 2010	MMM d, yyyy
15:55:35.231	HH:mm:ss.SSS
28 October 2015 15:55	dd MMMM yyyy HH:mm

Date/Time

If your source data contains date/time values, you can map them to the appropriate **Date Attributes** or **Time Attributes**. Single column can contain both, date and time, e.g. *5 Apr 2014 10:43:43 AM*. In this case, the date part, *5 Apr 2014* will be mapped to date attribute, the time part, *10:43:43 AM* to time attribute. See the following example:

The screenshot shows the BellaDati interface for configuring a new data source from a SQL database. On the left, a list of available columns includes 'L date of transaction'. In the main configuration area for 'Column 4', the 'Column type' is set to 'Date/Time'. Below this, there are checkboxes for 'Has date values?' and 'Has time values?', both of which are checked. The 'Date attribute name' is set to 'Date of Transaction'. The 'Format settings' section is highlighted with a red box, showing the 'Format' set to 'Custom' with the value 'yyyy-MM-dd' and the 'Language' set to 'English'. To the right, a 'Preview' table shows the resulting data format, with a single column 'L date of transaction' containing dates like '2014-01-29' and '2014-01-30'.

Definition of Date/Time format is the same as for the Datetime column.

Translation

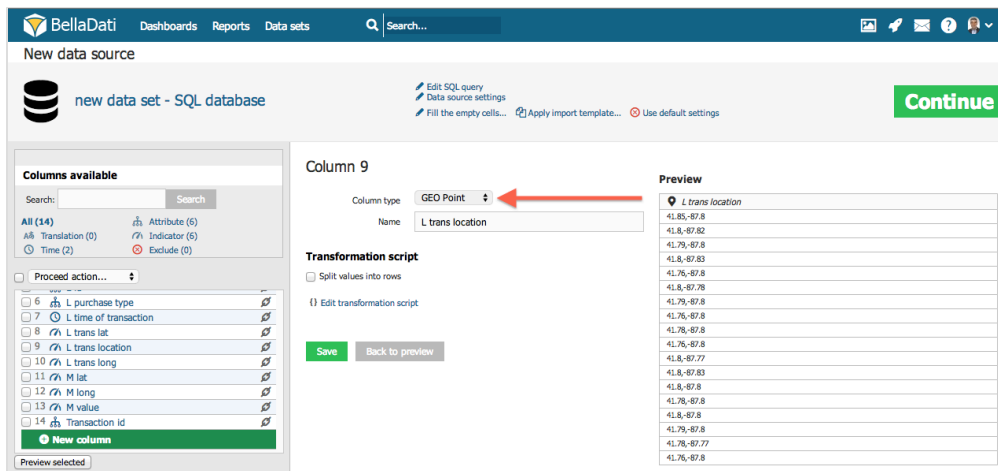
BellaDati allows you to directly import **Attribute translations**. In order to set up **Attribute translation** navigate to column with language metaphor and:

- choose **Translation** in **Column Type**
- select translation **Language**
- specify **index** of **original column**

The screenshot shows the BellaDati interface for configuring a new data source from Google Drive. On the left, a list of available columns includes 'Lastname', 'Firstname', 'Education', 'Position', 'Department', 'State', 'County', 'Paydate', and 'Amount'. In the main configuration area for 'Column 5', the 'Column type' is set to 'Translation'. Below this, the 'Language' is set to 'English' and the 'Translation of column' is set to '1'. There are 'Save' and 'Back to preview' buttons. To the right, a 'Preview' table shows the resulting data format, with a single column 'All Translation of column 1' containing values like 'Foodz, Inc.'.

GEO Point

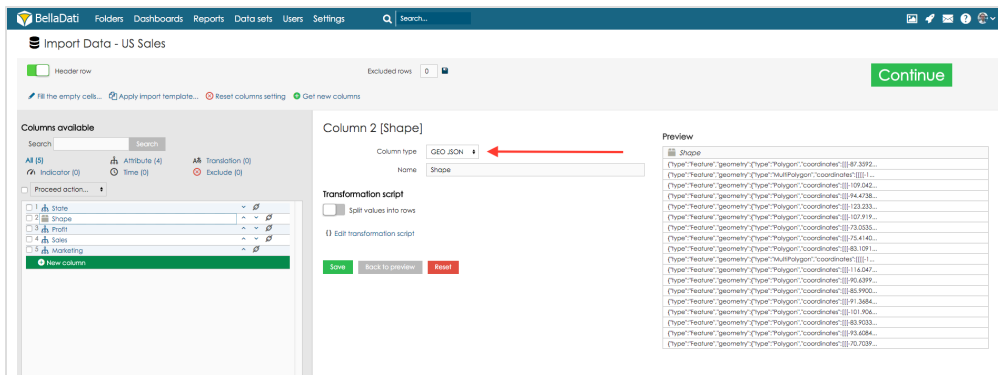
In order to map the longitude/latitude onto the GEO Point attribute, you have to specify the latitude/longitude in single column in format **latitude;longitude**, e.g. *43.56;99.32*. Decimal separator is . (dot). You can do it using the transformation script, e.g. **value(2) + ";" + value(1)** in case the longitude is stored in column 1 and latitude in column 2.



GEO JSON

BellaData allows you to use GEO JSON directly in the data set. GEO JSON has to be specified in format like followings:

- {"geometry":{"coordinates":[[[38.792341,33.378686],[36.834062,32.312938],[35.719918,32.709192],[35.700798,32.716014],[35.836397,32.868123],[35.821101,33.277426],[38.792341,33.378686]]],"type":"Polygon"},"type":"Feature"}
- {"type":"FeatureCollection","features":[{"type":"Feature","geometry":{"type":"Point","coordinates":[102.0,0.5]},"properties":{"prop0":"value0"}}]}
- {"type":"FeatureCollection","features":[{"type":"Feature","geometry":{"type":"Point","coordinates":[102.0,0.5]},"properties":{"prop0":"value0"}},{type":"Feature","geometry":{"type":"LineString","coordinates":[[102.0,0.0],[103.0,1.0],[104.0,0.0],[105.0,1.0]]},"properties":{"prop0":"value0","prop1":0.0}},{type":"Feature","geometry":{"type":"Polygon","coordinates":[[100.0,0.0],[101.0,0.0],[101.0,1.0],[101.0,1.0],[100.0,1.0],[100.0,0.0]]},"properties":{"prop0":"value0","prop1":0.0}}]}
- {"type":"Feature","geometry":{"type":"MultiPolygon","coordinates":[[[[120.715609,-10.239581],[138.668621,-7.320225],[102.498271,1.3987],[103.07684,0.561361],[103.838396,0.104542],[104.53949,-1.782372],[104.887893,-2.340425],[105.622111,-2.428844],[106.108593,-3.061777],[105.857446,-4.305525],[105.817655,-5.852356]]]]}}



Filling of Empty Cells

It's usual that imported data contains empty cells. It's usually necessary to replace this empty cells with own values (e.g. "0", "none", "N/A" etc.). If you want to do this, you have two possibilities, how to fill in these empty cells:

1. **globally** - fill empty cells with chosen value in all columns (located below batch column settings)
2. **locally** - fill empty cells with chosen value in particular column (located in the window of particular column settings)

Global changing is available in the top blue line directly under encoding settings. After clicking just type the value, which will be entered in all empty cells in your data.

Local changing is available after clicking on column name in the list. There you can enter your own value for empty cells (but only for this particular column). You can freely combine these two methods - for example you can fill in all the empty cells with "0" value, but one particular attribute column can be refilled with "N/A" value.

Merging Columns

Column merging function enables to load data from more source columns to one target column during import process.

Typical use cases are:

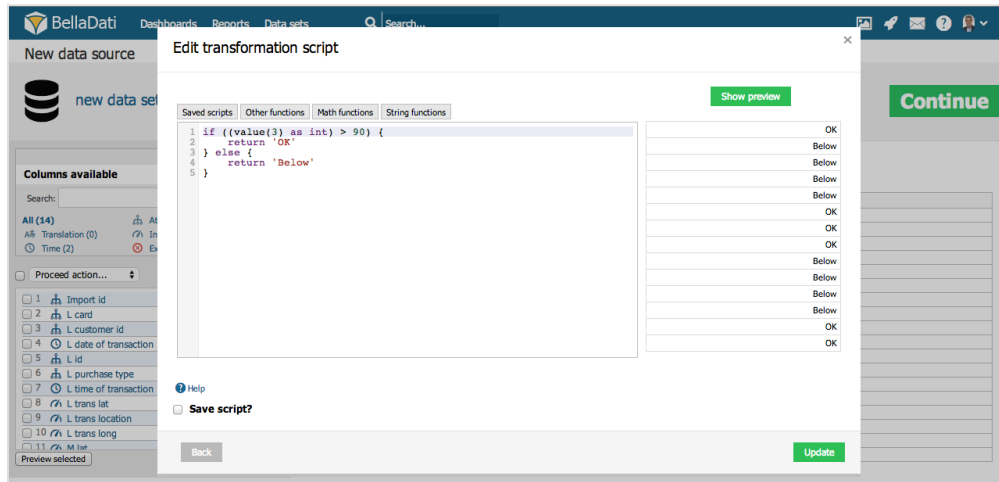
- Time is separated in more columns (days, months and years or time in different columns)
- Two columns representing one entity (eg. first name and surname of one person)

Click the chain icon in the columns list, choose another column to merge with and select appropriate separator which will be added between values (space, comma, dot, semicolon, pipe). You can disconnect merged columns too.

Another way to merge columns and set up more advanced options is by [transformation scripts](#).

Transformation Scripts

Transformation scripts allow advanced data transformations during import. These scripts are based on Groovy programming language syntax.



Transformation scripts allows you the following:

- Modify values stored in BellaDati data warehouse according defined functions and conditions.
- Create new columns (date/time, attributes, indicators) with transformed or combined values from other columns. Values in different cells are indexed from 0 and displayed near column names within import settings screen.
- Perform advanced calculations in date/time (eg. period of some action undertaken between two dates).

Basic script commands:

- `value()` - returns actual value of the current cell
- `value(index)` - returns value of cell on desired (indexed) position in the actual row
- `name()` - returns name of the column
- `name(index)` - returns name of the column at desired position
- `format()` - returns value of the format in actual column (only time and indicator column types)
- `actualDate()` - returns actual date in dd.MM.yyyy format
- `actualDate('MM/dd/yyyy')` - returns actual date in chosen format (e.g. MM/dd/yyyy)
- `excludeRow()` - excludes the row



These transformations are applied for each import including scheduled automatic imports from [Data Sources](#).

Go to [Transformation scripting guide](#) for more details.

Reusing Transformation Scripts

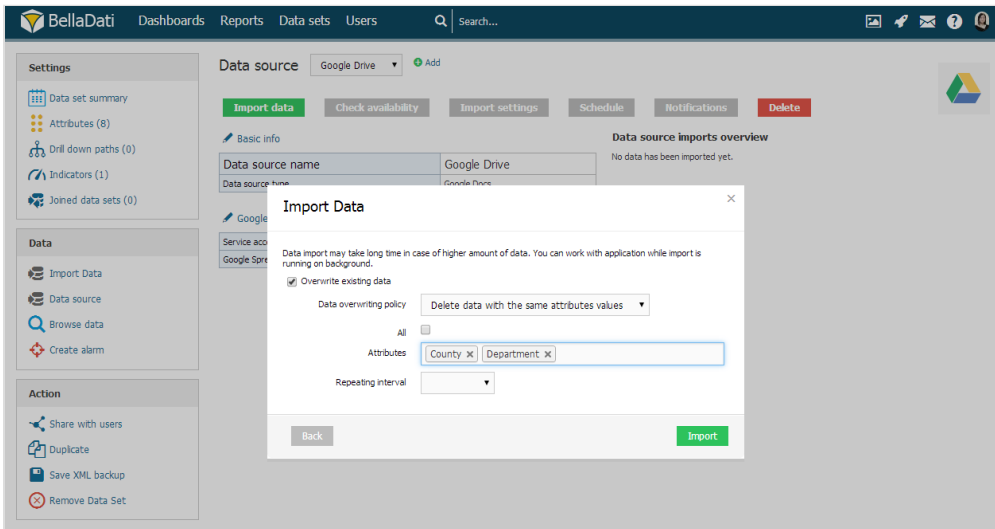
If you know that you are going to use your script again in the future, you can save it by switching the toggle in bottom left corner. Your saved scripts can be found in the top menu in "Edit transformation script" pop-up window.

- **No overwrite:** Imported data will be appended to existing (default).
- **Delete data with the same attributes values:** Deletes all existing records with the same combination of attributes as in the imported data.
- **Delete all data** before import.
- **Replace** rows with identical data.

Delete data with the same attributes values

When deleting data according attributes, BellaDati allows you to:

- select **All** attributes.
- select **specific** attributes - the import procedure will compare desired attributes and will overwrite the row if the current attribute is equal to the value already stored in the database.



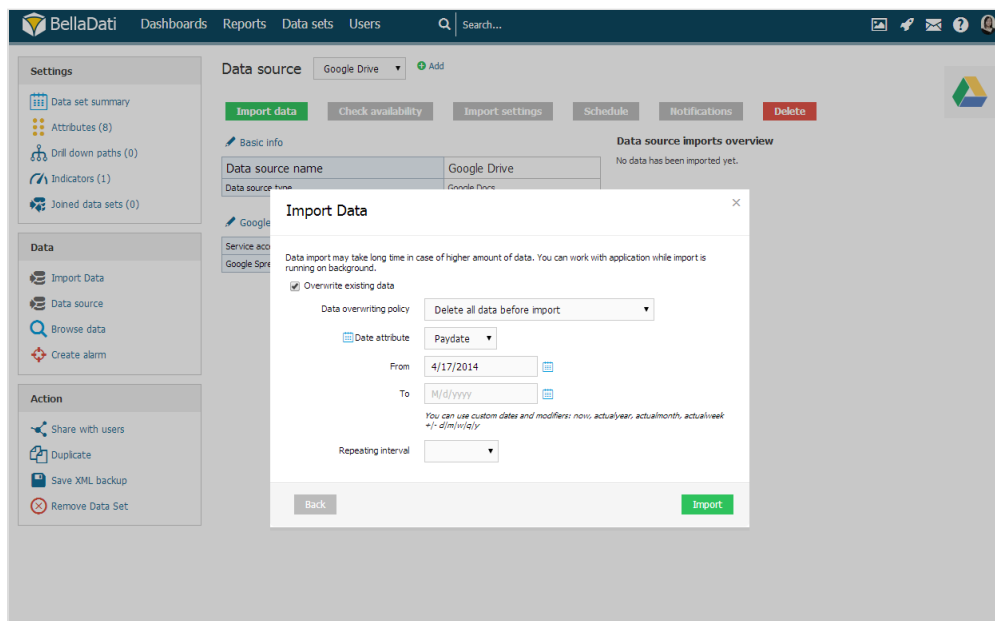
BellaDati also allows you to apply these settings only for the data imported from selected data source.

Delete all data before import

When deleting all data before import, BellaDati allows you to select specific **time interval**. Setup **From** and **To** to restrict data erasing.

✔ Use calendar icons to comfortably select desired time intervals.

ℹ You can use custom dates and modifiers: now, actualyear, actualmonth, actualweek +/- d|m|w|q|y.



Import Progress



Import of lot of data may take a long time to complete.

Data are being imported asynchronously, therefore BellaDati functions are still available during import. The user can be logged out during the import too.

Data set summary page shows actual import progress bar with estimated time and percentage.

Before import finishes, you are able to:

- **Cancel** running import: All data related to this import will be erased from BellaDati data warehouse.
- **Notify by e-mail**: An e-mail will be send to you after the import has been finished.

Import Results

You can display the detailed results for each import. Each import can reach the following status:

- **green** - successfully finished import without errors
- **orange** - finished import containing errors
- **red** - aborted or unfinished import
- **gray** - deleted import
- **blue** - scheduled future import (only in case of external data sources)



Records with errors are not imported!

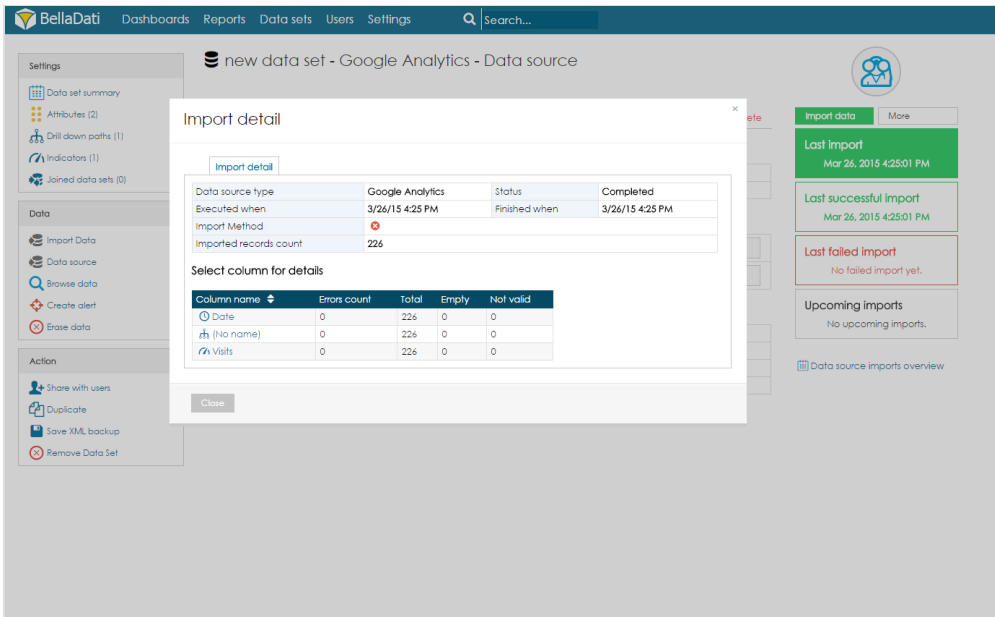
If errors occurred during import, you can find the reasons in the import detail popup:


- The column name with the error is highlighted in red and errors count is displayed.
- Details for each column (click on it's name):
 - Total: Successfully imported records count.
 - Empty: Records count containing no values in selected columns (these records are imported however).
 - Number (only for indicators): Record counts with numeric values.
 - Not valid: Records with errors count (these records are not imported).
 - Error details: Displays records number in source file to find problem.



Typical errors during import are:

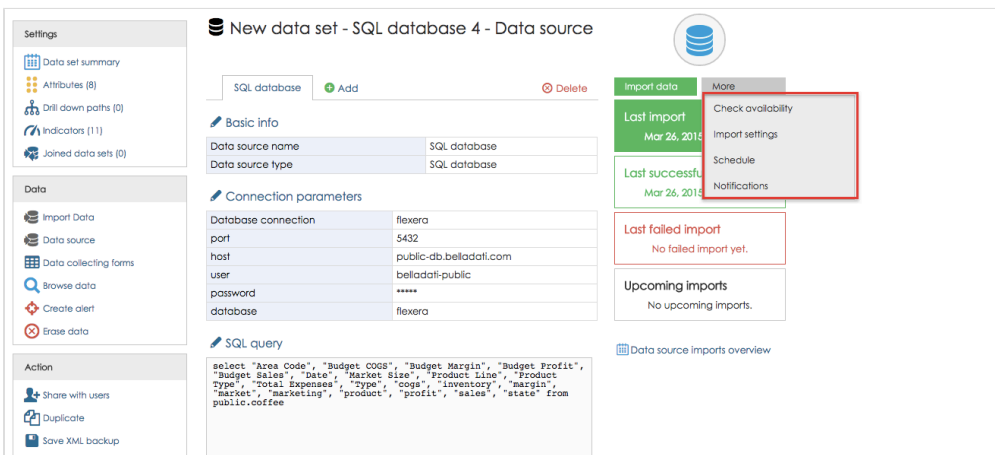
- Date/time format mismatch: Check and set the proper format for date/time column in [Import Settings](#).
- You are trying to assign an indicator not numeric values: Consider setting such column as attribute.



 Display window with list of the import by clicking button Data source imports overview.

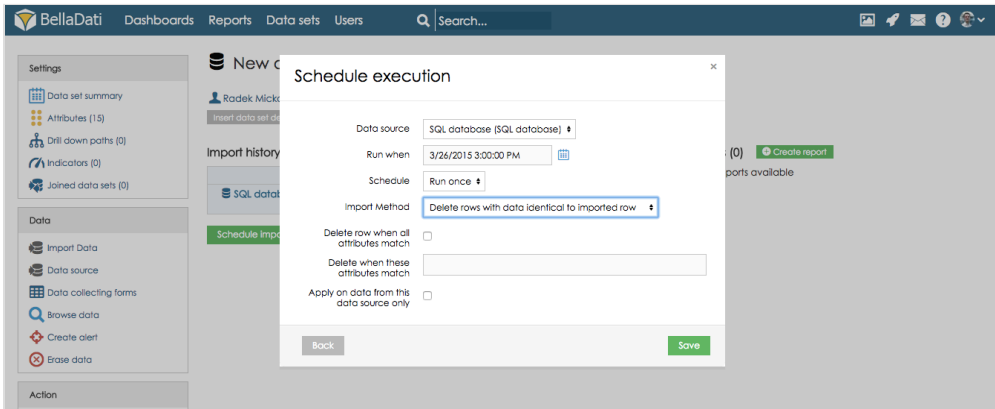
Hover your mouse over button more to display more actions which allow you to:

- **Check availability** - displays dialogue window which allows user to check data source availability
- **Import settings** - redirects to [import settings](#) page
- **Schedule** - redirects to [Schedule import](#) page
- **Notifications** - opens dialogue window with notification settings where can be set for which import results and to which users email notification will be sent
 - Possible import results:
 - Completed
 - Completed with errors
 - Error
 - You don't have enough space in your data warehouse
 - Wrong data source configuration



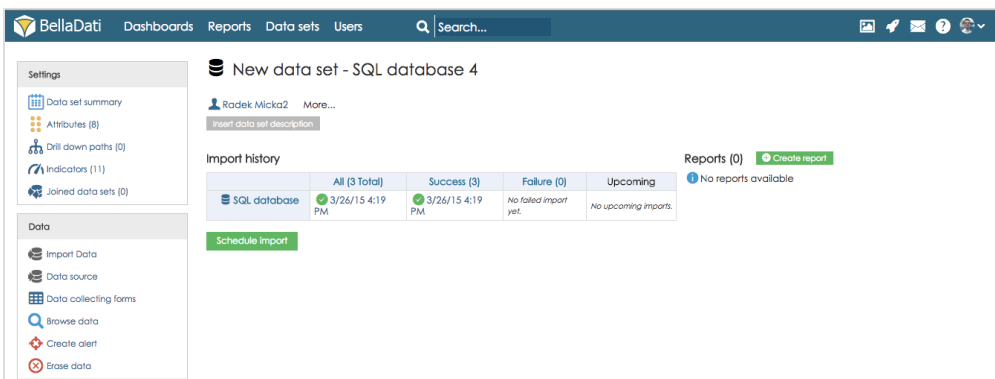
Scheduling Import

BellaDati's integrated scheduler allows you to control, monitor and schedule automatic import from [data sources](#). The overview is represented by a navigable calendar.



Actions available:

- **Schedule import**
 - Set the exact time of automated execution
 - Configure the [data overwriting policy](#) if required
 - If you need the import to be repeated after specific period, configure the **repeating interval** (day, week, month, quarter, year). Leave it empty, to execute the import once.
- Tooltips actions (only for already finished imports):
 - [Browse imported data](#)*
 - Display the [import results](#) summary: Useful for import statistic or problems diagnostics.
 - **Delete imported data**: Deletes all data created within the particular import.
 - **Cancel import** (cross symbol): You can cancel future scheduled imports.




Additional tooltip information for executed imports:

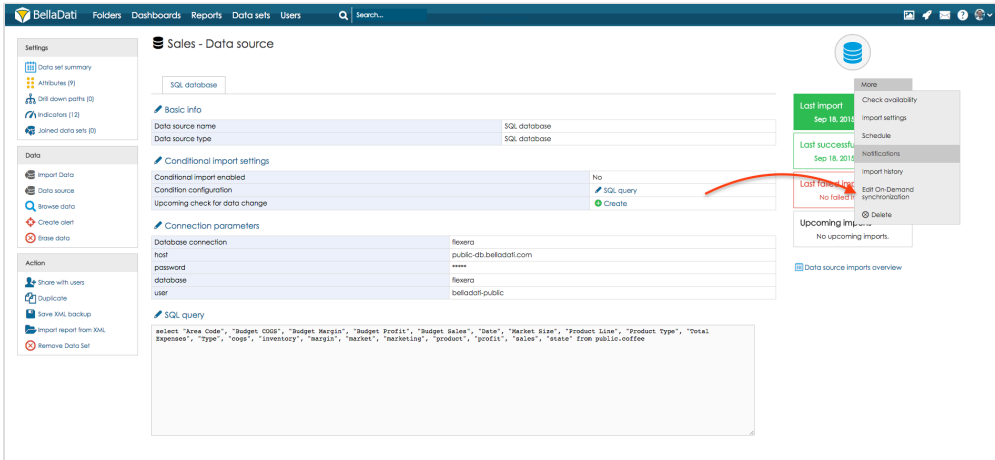
- Download the original file
- Execution date and time
- Status: Completed, Running
- Overwrite existing data: yes, no
- Scheduled by: author name

You can also enable [on-demand data synchronization](#), which allows users to manually trigger data import from report.

On-demand data synchronization

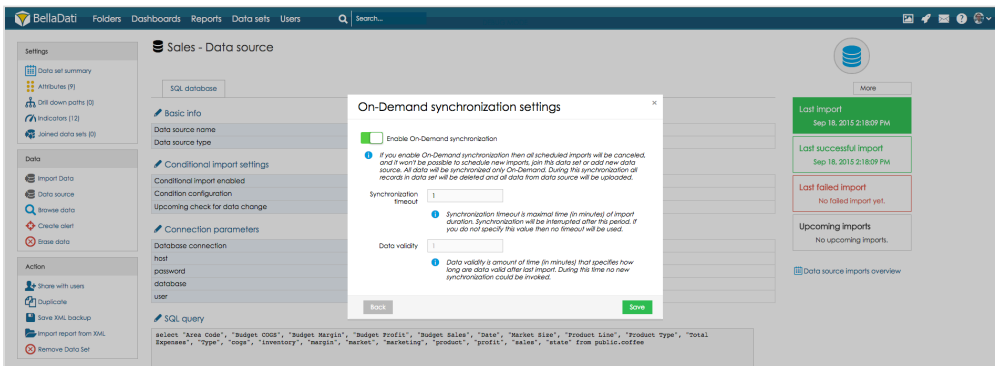
On-demand data synchronization allow users to trigger import process. To enable on-demand data synchronization, go to data source settings and select option Edit-On Demand synchronization

 On-demand synchronization deletes all the data in the data set and imports complete set of the data.

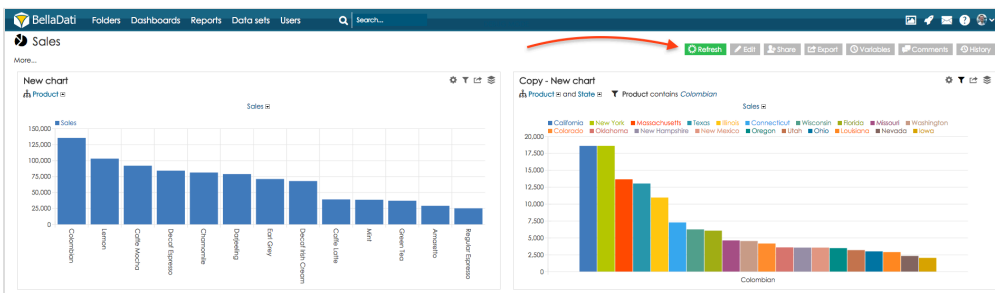



Selecting this option will open dialogue window with On Demand Synchronization settings. This dialogue window allows you to:

- **Enable on-demand synchronization**
- Set **Synchronization timeout** (maximal time of import duration in minutes. Synchronization will be interrupted after this period. If you do not specify this value then no timeout will be used.)
- Set **Data validity** (amount of time (in minutes) that specifies how long are data valid after last import. During this time no new synchronization could be invoked.)



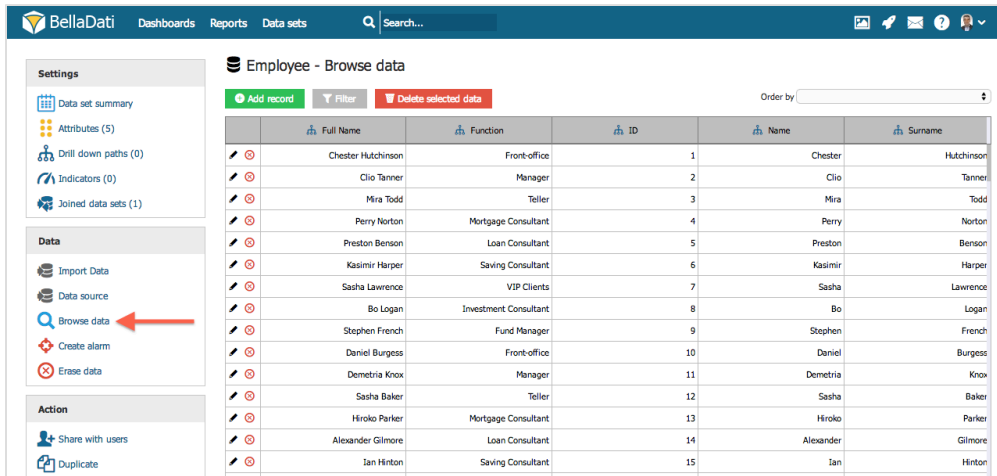
When the on-demand synchronization is enabled, the only way to trigger the import is in the data set by using button **Refresh**:



 Only report editors who have access to the source data set (can view or edit), are allowed to trigger the synchronization.

Browsing Data

Browsing existing data within data set is available after clicking to "Browse data" in left sub-menu.



The screenshot shows the BellaDati interface with the 'Employee - Browse data' view. The left sidebar has a 'Data' section with 'Browse data' highlighted. The main table displays the following data:

	Full Name	Function	ID	Name	Surname
✎	Chester Hutchinson	Front-office	1	Chester	Hutchinson
✎	Clio Tanner	Manager	2	Clio	Tanner
✎	Mira Todd	Teller	3	Mira	Todd
✎	Perry Norton	Mortgage Consultant	4	Perry	Norton
✎	Preston Benson	Loan Consultant	5	Preston	Benson
✎	Kasimir Harper	Saving Consultant	6	Kasimir	Harper
✎	Sasha Lawrence	VIP Clients	7	Sasha	Lawrence
✎	Bo Logan	Investment Consultant	8	Bo	Logan
✎	Stephen French	Fund Manager	9	Stephen	French
✎	Daniel Burgess	Front-office	10	Daniel	Burgess
✎	Demetria Knox	Manager	11	Demetria	Knox
✎	Sasha Baker	Teller	12	Sasha	Baker
✎	Hiroko Parker	Mortgage Consultant	13	Hiroko	Parker
✎	Alexander Gilmore	Loan Consultant	14	Alexander	Gilmore
✎	Ian Hinton	Saving Consultant	15	Ian	Hinton

Data browser allows you to perform following actions:

- Edit data - individually per record (row)
- Delete row
- Order displayed data by: date or any other attribute
- Filter displayed data:
 - by date interval
 - by attributes members filtering
- Add new record (row)
- **Data export:** Save data in CSV: Data can be saved in CSV or XLSX directly or in ZIP format to save download time and space on your storage.



Please note, due to browsers' performance issues, maximum 2000 rows of data is displayed in data browsing window. Use filters to display only desired subset of all data stored.




Deleting one row is performed immediately after clicking corresponding icon! There is no confirmation message displayed before.

Managing Indicators

Managing the [indicators](#) is a part of BellaDati's **data warehouse modelling**. Values of particular indicators represent the point of the examination interest. Indicators have usually numeric values (wages, costs, incomes etc.).

We are distinguishing two types of indicators:

- **data set indicators** are defined within the data set and are available as musters for report indicators. Another settings (like aggregations, appearance, etc.) are not supported. Simply said, the data set indicators represents a **raw numerical value** (in the OLAP language it is a fact) with basic attributes - name, unit and rounding mode. Values of these indicators are straightforwardly stored in BellaDati's data warehouse directly from imported data.

 Each **data set indicator** is represented by its code (unless defined by formula). This code always begins with "M_" prefix and serves as unique identification for usage in formulas. *Note: Indicator code is not editable and is assigned during it's creation.*

- **report indicators** are created in the report from the data set indicators or ad hoc. Unlike the data set indicators, report indicators are supporting wide range of various settings - **aggregations**, **appearance**, **conditional formatting**, extended **formula support** with **Report Variables and Filters components** etc. Report indicators can contain also **non-numerical values**.

BellaData

Dashboards Reports Data sets Users

Search...

Settings

Data set summary

Attributes (10)

Drill down paths (0)

Indicators (12)

Joined data sets (0)

Data

Import Data

Data source

Browse data

Create alarm

Erase data


Campaigns - Indicators

Create indicator

Create a group of indicators

	Code / Formula	Action
ActualCost	M_ACTUALCOST	
AmountAllOpportunities	M_AMOUNTALLOPPORTUNITIES	
AmountWonOpportunities	M_AMOUNTWONOPPOTUNITIES	
BudgetedCost	M_BUDGETEDCOST	
ExpectedResponse	M_EXPECTEDRESPONSE	
NumberOfContacts	M_NUMBEROFCONTACTS	
NumberOfConvertedLeads	M_NUMBEROFCONVERTEDLEADS	
NumberOfLeads	M_NUMBEROFLEADS	
NumberOfOpportunities	M_NUMBEROFOPPORTUNITIES	
NumberOfResponses	M_NUMBEROFRESPONSES	
NumberOfWonOpportunities	M_NUMBEROFWONOPPOTUNITI	
NumberSent	M_NUMBERSENT	

- **Formula indicators:** Their value is calculated according assigned mathematical formula or another operation. Values of indicators created by formula are not imported to data set, they are evaluated from formula. Formula indicators can be defined on report level as well. See [Formula Reference Guide](#) for details.

 Data set indicators contain numeric values only. Formula indicators on the other side can contain also textual values or member counts.

Creating data set indicator

New indicator can be created by following ways:

- Within data set (data set indicators)
- During the import
- Transforming an attribute into a data set indicator
- When inserting data row manually

Data set indicator settings

When you click on the name of particular indicator name in the data set, you can adjust a lot of different parameters of this indicator. You can define or change:

- Indicator name
- Indicator unit
- Decimal format (see chapter below)
- Type of rounding
- Assign indicator to group
- [Formula](#). This function is available only when the indicator is defined by formula or when creating new one.

Those adjustments will take effect in each occurrence of this indicator (data sets, reports, dashboards).



When the data set is a part of [join](#), you can choose if you want to propagate new indicator to joined data set (by checking the propagate option in dialog window).

Decimal format

Decimal format is a useful tool to:

- insert separate characters into indicator
- insert additional characters into unit
- adjust number of decimal positions

Indicators without adjusted decimal format are displayed with a comma after each three positions of digits (thousands, millions etc.) and with dot between whole number and decimal places. You can use prepared help window in BellaDati for some examples, how to define the decimal format. Basic description is also in the table below:

Code	Meaning
#	Represents one digit
,	Separator of digit places (thousands, millions etc.). It's displayed as a comma in English localization.
.	Decimal separator. It's displayed as a dot in english localization.
%	Multiply value by 100 and add % symbol
"	If you want to display any of described operational character (., #), you have to place it between " characters. It's a useful function, when you deal with shortcuts which ends with "." character.

Decimal format examples:

Actual value	Display	Code
1234	1234	#
1234	\$1234	\$#
1234.56	\$1 235	\$\$,###
1234.56	1 234,6	#,###.#
1234.56	\$1 234, 56	\$\$,###.##
1234	\$1 234 k	\$\$,### k
0.56	56 %	# %
0.5612345	56.12345%	#.#####%



You can display brief decimal format help by clicking on "Show help" link in the popup window.

Removing Indicator

Click on the cross icon (Remove link) at the end of corresponding row in indicators' list. Then confirm the removal in popup.



Indicator removal will affect all reports and dashboards where the indicator has been used!

Grouping Indicators

Indicators grouping enables better structuring and organizing of your indicators. Each indicator can be assigned to the indicator group and each group can be nested to another group.

The main advantage of indicator groups is that you handle all contained indicators as a single object.

Each group contains embedded functions like **sum** or **average** from the nested indicators or sub-groups.




Indicator groups should contain indicators with similar characteristics - eg. financial indicators, performance indicators etc.

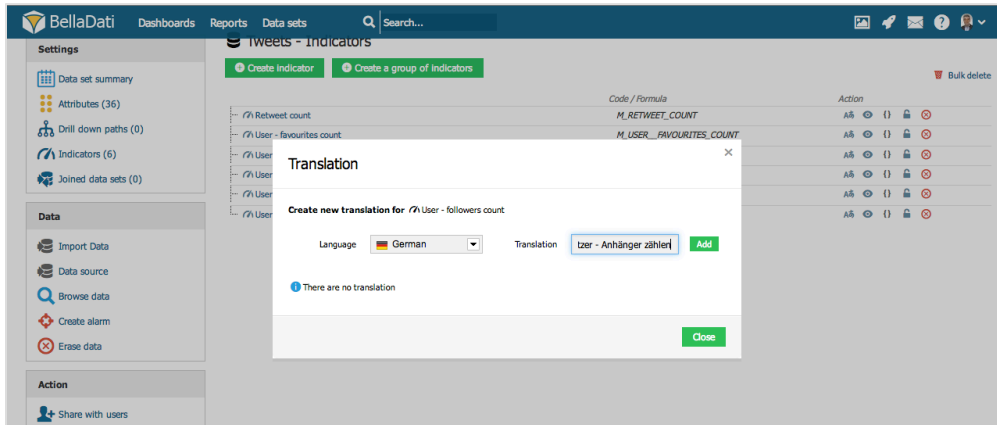
There are three ways how to create indicator groups:

- manually on the data set indicators page
- in the [indicator settings](#) dialog window
- within the [report](#)

Translating Indicators


Translation is useful tool when you have imported data in one language but then you want to display your data in another language - eg. for presentation to managers. Then if you change the language in user profile or switch it in your browser, all the translated indicators will be displayed in new alternative language. Indicators without translation remain displayed the same way (unchanged).


 Logout and repeated login to BellaDati may be needed to reflect language change.



Go to desired data set via data set menu.

1. Select "Indicators" in the left submenu.
2. Click on the translation icon at next to the name of selected indicator. A popup will appear. All existing translations of actual are displayed in the table below.
3. Select the target language in "Language" drop-down box.
4. Type translated name into "Translation" field.
5. Click "Add" button. The new translation will be added to the table below.
6. Choose another language and repeat the process or click "Close" to close popup.

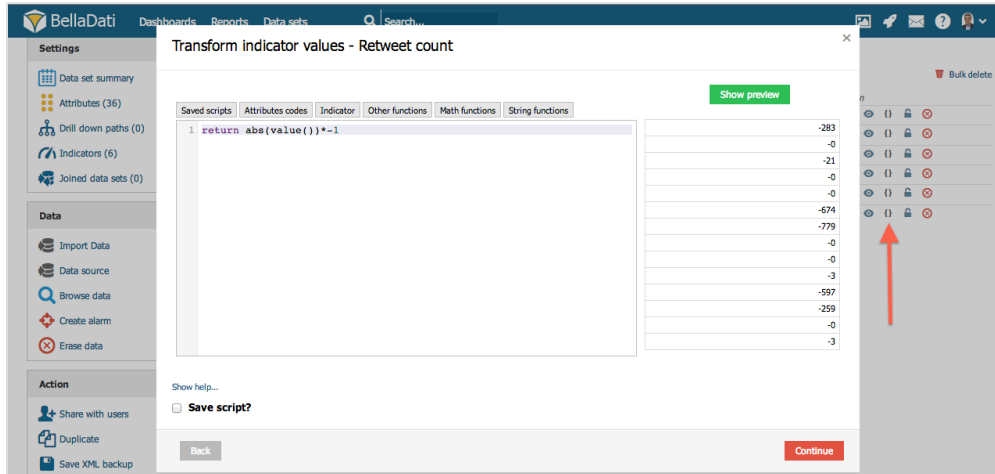
 To **edit existing translation**, please remove the actual translation first by clicking the cross icon at the end of corresponding row. Then add the translation again according instructions above.

 **Indicator traslation will not be reflected in views, where the indicator has been renamed before!** See [report indicator editing](#) for details.

Transforming Indicator values

Transforming indicator values using script

You can transform the indicator values using the transformation script.



How to access the column values?

Accessing the value we want to process is a key issue. Scripts provide a function `value()` which returns the current value. There are more advanced possibilities to access values:

`String value()` - returns the current value

`String value(String columnName)` - returns the value of the specified column

`LocalDate rowDate()` - returns date of the current row

`LocalTime rowTime()` - returns time of the current row

To get more information about the transformation, visit the developers section [Transformation scripting](#).

Build indicators with formula



Visit [Formula Reference Guide](#) page.

Adding Permissions to Indicators

BellaDati let you limit access to **Indicators** only for selected **users** or **user groups**.

Adding Access Rights

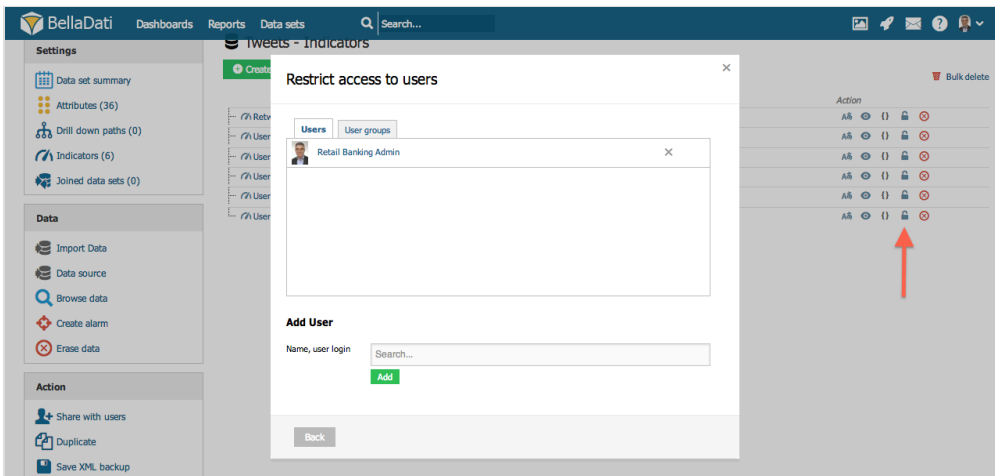


Note

Note, that by default Indicators have **Global Permission**. Every user with access to data set has access to its indicators.

In order to assign access rights, navigate to **Data Set** and select **Indicators** in **Settings**. Click on **lock icon** in row indicating the indicator.

- Switch between **User** and **User Groups** tab.
- Search **User** or **User Group** name(s).
- Click **Add**.



Note

When Indicator has assigned access rights at least to one user, it loses **Global Permission** and becomes restricted to other users.

Removing Access Right

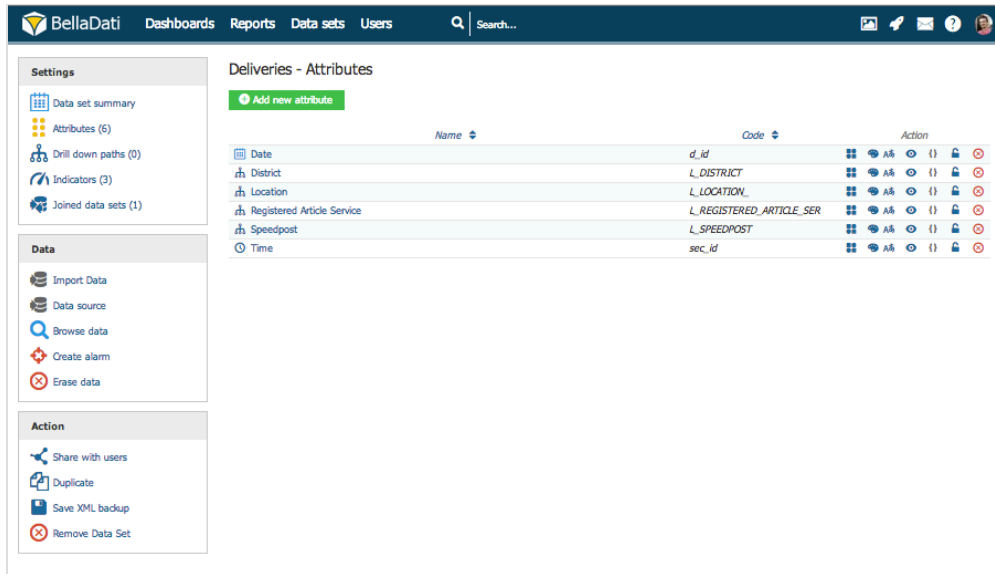
To remove access right for particular user:

- Switch between **User** and **User Groups** tab.
- Click on **cross icon** next to User name.
- [Build indicators with formula](#)

Managing Attributes

[Attributes](#) definition is a part of the BellaDati's **data warehouse modelling**. All attributes have an unique code beginning with "L_" prefix (L as level) that serves as unique identifier (eg. for counts, filtering, custom members definition, etc.).


Instance of particular attribute is called [member](#).




Creating Attribute

Attributes can be created in three different ways:


- during the [data import](#)
- manually on the data set attributes page
- during [drill-down path definition](#)

 Attribute name must be unique in the whole data set. Otherwise warning message will be displayed and the attribute will not be saved.

 When the data set is a part of [join](#), you can choose if you want to propagate new attribute to joined data set (by checking the propagation option in dialog window).


Editing Attribute

You can modify the attribute by clicking on it's name in the list.

- 
1. All changes in attributes name will influence immediately all existing [Reports](#) and [Dashboards](#) containing affected attributes.
 2. Changing of attribute name does not affect it's code
 3. Changing of attribute name will not affect attribute's names in joined data sets.


Deleting Attribute

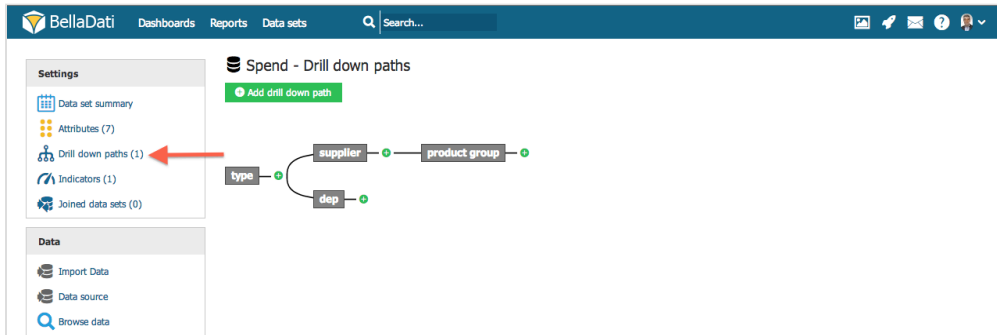
The attribute will be deleted with all corresponding data.

 Attribute removal will influence immediately all existing [Reports](#) and [Dashboards](#) (including filters and custom members). All data assigned to this attribute will be deleted.

Defining Drill-down Path

Predefined drill-down path specifies the meaningful sequence might by useful for further drill-down operation in report (see attached video tutorial below). Each data set can contain more drill-down paths. The simplest drill-down path is linear, however BellaDati supports more complex structures like trees.

 Predefined drill-down paths are available int the report when performing ad-hoc drill-down (by clicking on + in the table), or in the drill-down settings.




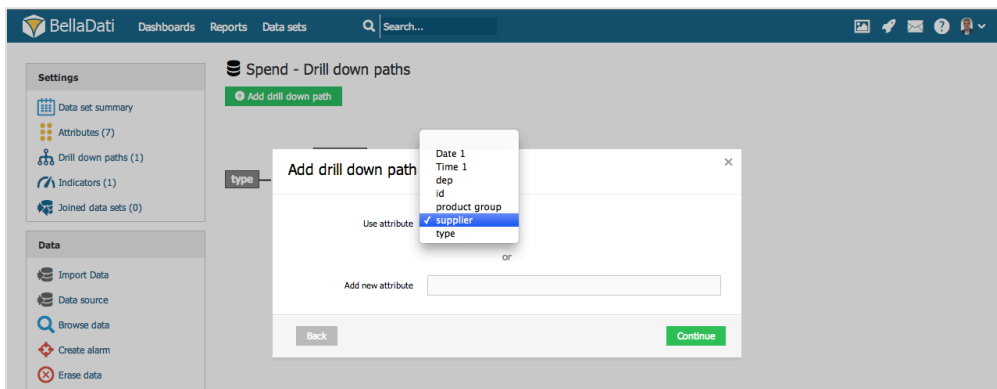
1. Go to desired data set via data set menu.
2. Select "Attributes" in the left submenu.


 Predefined drill-down paths significantly simplifies [creating reports](#) by analysts and performing drill-down operation in reports and on dashboards. Therefore we recommend defining these paths on the data set level.

Adding new drill-down path

1. Click "Add drill down path" at the bottom of the page. A popup will appear.
2. Select first attribute in "Use attribute" drop-down box and click on "Save" button.
3. Click the green plus icon on the right of existing attribute in drill-down path to add another attribute and repeat steps above to create whole drill-down path.

 By clicking on green plus icon in the middle of existing drill-down path, you can branch out current drill-down path to more sub-paths.



 You can create an new attribute during drill-down path definition by filling in the field "Add new attribute" in the popup. See [managing attributes](#) for details.

Editing drill-down path

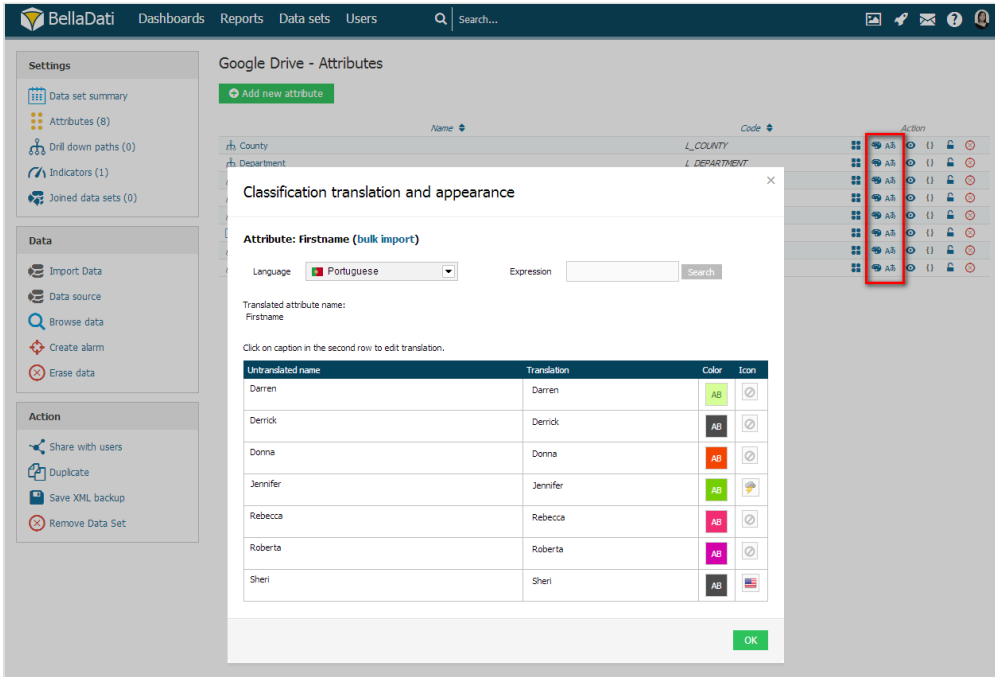
- Removing attribute: Hoover over the desired attribute, click on the right top cross symbol and confirm this action in popup.

Members Appearance

This function allows you to adjust following appearance options for individual attribute [members](#):

- color
- icon

Both can be assigned to particular member simultaneously.



1. Go to desired data set via data set menu.
2. Select "Attributes" in the left submenu.
3. Click on the translations and appearance icon at the end of the row of selected indicator. A popup will appear. Actual color and icon assignation is displayed in the table below.
4. **Color:** Click icon in "Color" column in the row of corresponding member. A color selection tool will appear. Click on desired color. Note: Clicking on "Default" field will reset member color to default (none).
5. **Icon:** Click icon in "Icon" column in the row of corresponding member. An icon selection tool will appear. Select desired category in drop-down box and then click required icon. Note: Clicking on left top symbol will reset member color to default (no icon).
6. Click "OK" to close popup.



When there are lot of members, you can filter them by using "Expression" field. Just type in the part of the names of requested members and then click "Update" button. Empty field means no member filter is active.



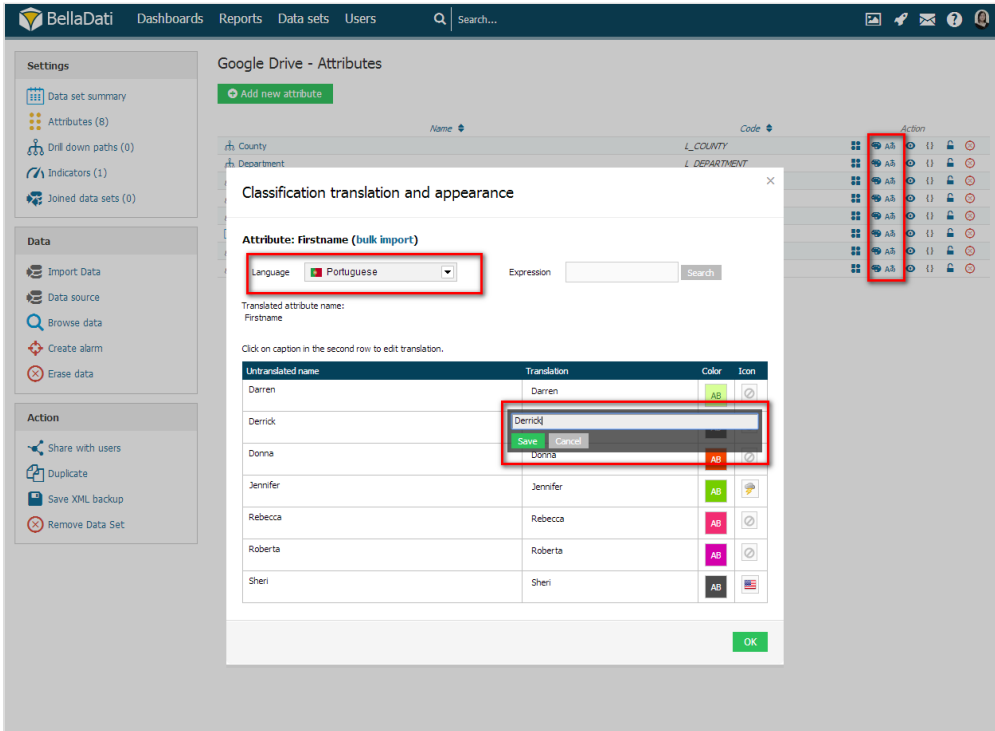
All changes in member appearance will influence immediatelly all existing [Reports](#) and [Dashboards](#) displaying these members.

Translating Attributes and Members

This function allows you to add or edit translations for:

- [Attributes](#)
- Attribute [members](#)

Translation is useful tool when you have imported data in one language but then you want to display your data in another language - eg. for presentation to managers. Then if you change the language in [user profile](#) or switch it in your browser, all the translated attributes and members will be displayed in new alternative language. Attributes and members without translation remain displayed the same way (unchanged).



1. Go to desired data set via data set menu.
2. Select "Attributes" in the left sub-menu.
3. Click on the translations and appearance icon at the end of the row of selected attribute. A popup will appear.
4. Select the target language in "Language" drop-down box. Now you see actual attribute and its' members translations in the selected language (indicated also in drop-down box - With translation section).
5. Edit the attribute translation by clicking its name in "Translated attribute name:" row via [in-line edit function](#) (bold). Then click "Save" button.
6. Edit members translation by clicking their names on corresponding row in "Translation" column and in-line editing. Then click "Save" button.
7. Click "OK" to close popup.



Attributes and members translation is available in reports as well.



When there are lot of members, you can filter them by using "Expression" field. Just type in the part of the names of requested members and then click "Update" button. Empty field means no member filter is active.



You can download existing translations by clicking on button Save as CSV. These translations can be imported to different data set. For this feature should be used [Bulk import](#) feature.



All attribute and member translations will influence immediately all existing [Reports](#) and [Dashboards](#) using these attributes and members.

If there should be used attribute appearance of data sets which are joined in the joined data set, select option *Use translation and appearance settings from the source data set* in the joined data set.

BellaDati

Folders

Dashboards

Reports

Data sets

Users

Search...

Settings

Data set summary

Attributes (0)

Diff down paths (0)

Indicators (0)

Joined data sets (0)

Data

Browse data

Create alert

Action

Share with users

Save XML backup

Import report from XML

Remove Data Set

Branch Visits join - Attributes

Show/Hide ML names

☐ Use translation and appearance settings from the source data set

Name	Code	ETL column name	Action
City	L_CITY		
Close	L_CLOSE		
Customer Since	L_CUSTOMER_SINCE		
Date	L_DATE		
Date	L_DATE_1		
Date 3	L_DATE_3		
Full Name	L_FULL_NAME		
Full Name	L_FULL_NAME_1		
Function	L_FUNCTION		
ID	L_ID_1		
ID	L_ID_1_2		
ID	L_ID_1_2_3		
Location	L_LAT		
Manager	L_MANAGER		
Name	L_NAME_1		
Name	L_NAME		
Open	L_OPEN		
Street	L_STREET		
Subject of visit	L_SUBJECT_OF_VIST		
Surname	L_SURNAME		
Surname	L_SURNAME_1		
ZIP	L_ZIP		
branch_id	L_BRANCH_ID		
customer_id	L_CUSTOMER_ID		
ID	L_ID		

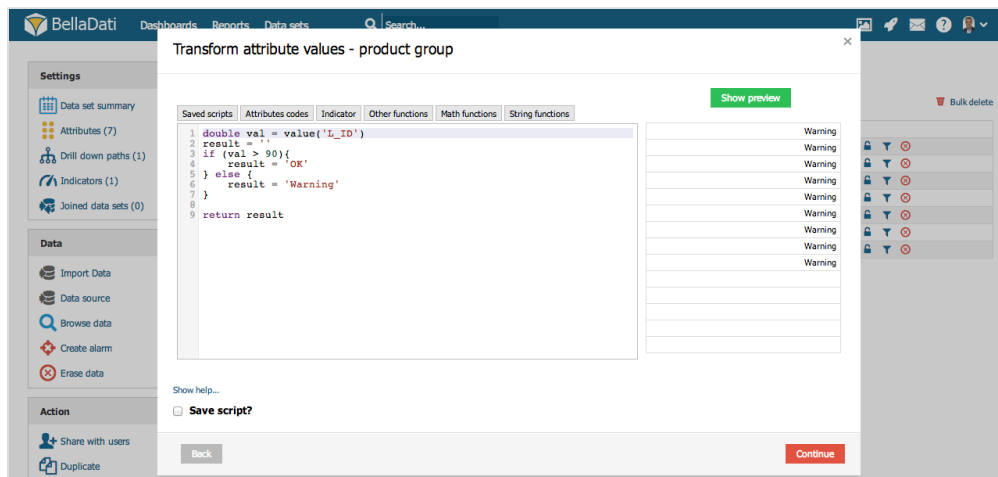
1

2

Transforming Attribute Values

Transforming attribute values using script

You can transform the attribute values using the transformation script.



How to access the column values?

Accessing the value we want to process is a key issue. Scripts provide a function `value()` which returns the current value. There are more advanced possibilities to access values:

`String value()` - returns the current value

`String value(String columnName)` - returns the value of the specified column

`LocalDate rowDate()` - returns date of the current row


`LocalTime rowTime()` - returns time of the current row

To get more information about the transformation, visit the developers section [Transformation scripting](#).

Creating Attribute Subsets

Attribute **Subset** is a virtual copy of attribute allowing you to:

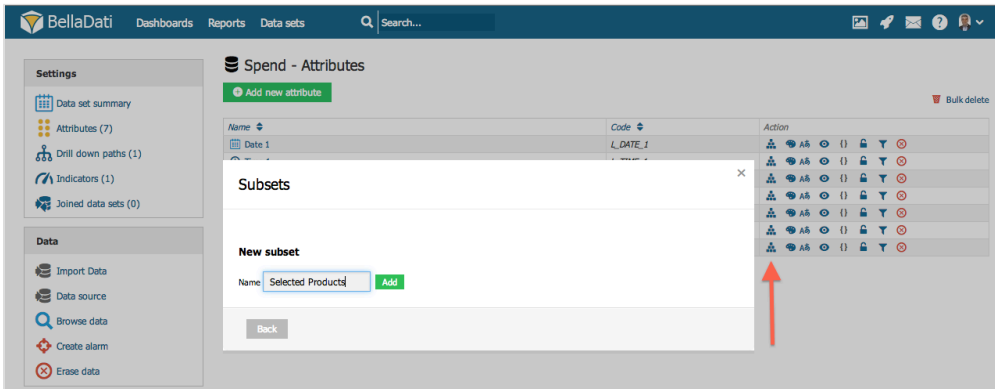
- select and use only desired members (include/exclude members)
- define custom order of members

 Into the subset can be added both, attribute and attribute custom members.

Creating Subset

Navigate to **Attribute management** by selecting **Attribute** in **Settings** panel of Data Set. Click on the **Subset** icon of particular attribute. *Subset editor* window will be opened.

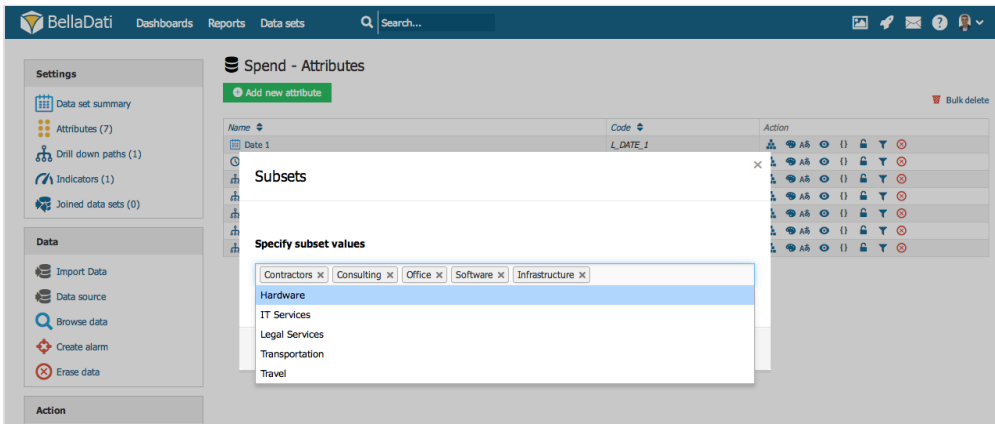
1. Provide subset **name**
2. Hit **Add** button



New **Subset** will be created.

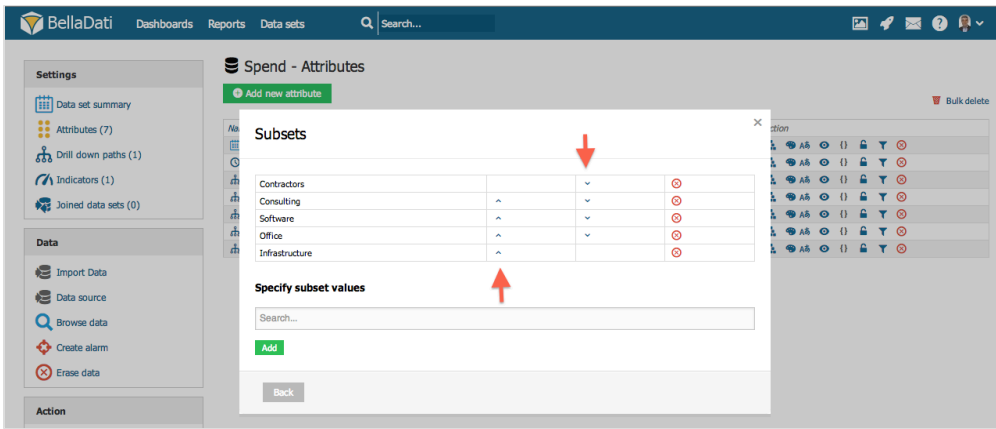
Selecting Members

Use BellaDati Search input to **Specify subset values** and click **Add**.



Specifying Order

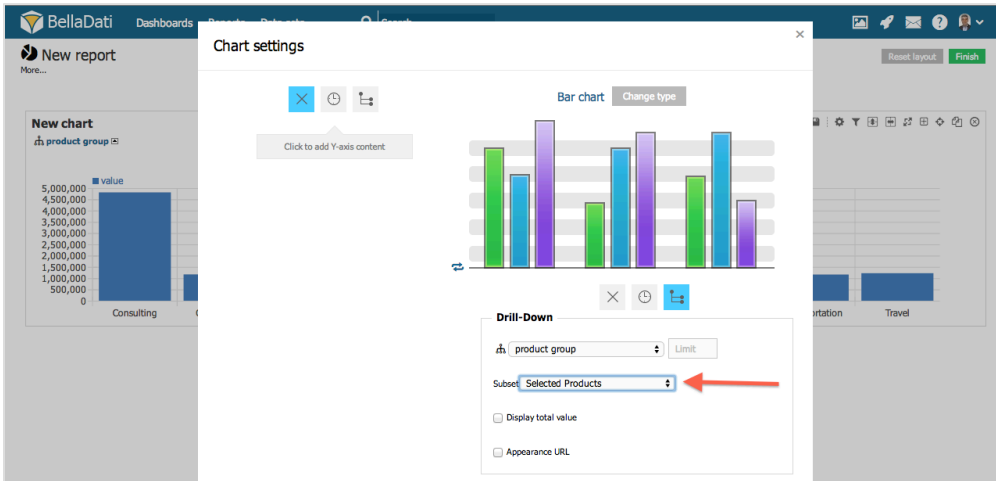
Use **Up/Down** arrow to define custom order of **Members** within the **Subset**.



Using Subsets

You can use existing **Subsets** instead of default **Attributes** and **Drill-down** paths in Reports. To apply Subset:

- Choose parent attribute of the **Subset** in **View Settings**
- Pick form the offered **Subsets**
- **Confirm** selection



Adding Permissions to Attributes

BellaDati let you limit access to **Attributes** only for selected **users** or **user groups**.

Adding Access Rights



Note

Note, that by default Attributes have **Global Permission**. Every user with access to data set has access to its attributes.

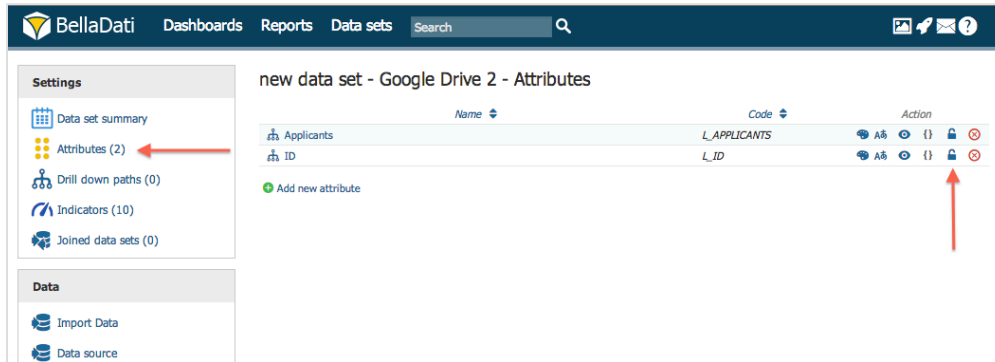
In order to assign access rights, navigate to **Data Set** and select **Attributes** in **Settings**. Click on **lock icon** in row indicating the attribute.

- Switch between **User** and **User Groups** tab.
- Search **User** or **User Group** name(s).
- Click **Add**.



Note

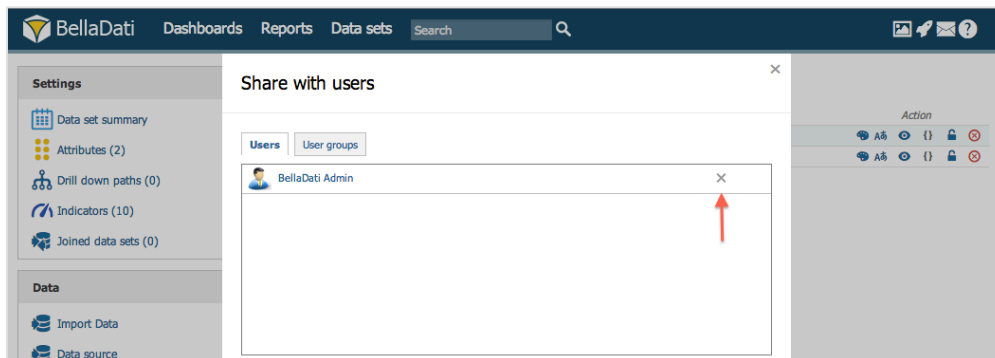
When Attribute has assigned access rights at least to one user, it loses **Global Permission** and becomes restricted to other users.



Removing Access Right

To remove access right for particular user:

- Switch between **User** and **User Groups** tab.
- Click on **cross icon** next to User name.




Setting Permission Filter (Lookup Table)

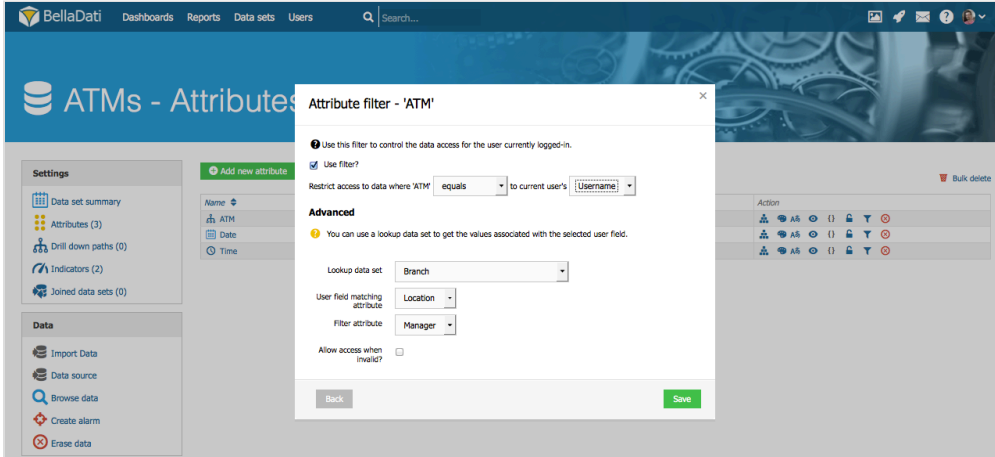
Permission filters allow you to define portion of the data (filter) which will be available to logged in user. Permission filters are always set up on particular **attribute** and has to correspond with user profile information.

In order to create Permission filter, click on **Filter** icon next to desired attribute.

1. Check **Use filter**
2. Select filtering **condition** (equals / not equal to)
3. Select user **profile parameter** (username / email / name / surname)
4. Choose **lookup table** if you want to filter by advanced parameters stored in dataset
5. Pick **matching attribute** and **filter attribute**

 In this case, there should be created different data set which contains **profile parameter** (username / email / name / surname) and **filter attribute** (g.e. department, company, etc.)

For example scenario with detailed steps how to create attribute filter proceed to [Setting Permission Filter - Use Case](#).



The screenshot shows the BellaDati interface with a modal dialog titled "Attribute filter - 'ATM'". The dialog contains the following elements:

- Header:** "Attribute filter - 'ATM'" with a close button.
- Instructions:** "Use this filter to control the data access for the user currently logged-in."
- Use filter?:** A checked checkbox.
- Restrict access:** "Restrict access to data where 'ATM' equals to current user's 'Username'".
- Advanced section:**
 - Tip: "You can use a lookup data set to get the values associated with the selected user field."
 - Lookup data set:** "Branch"
 - User field matching attribute:** "Location"
 - Filter attribute:** "Manager"
 - Allow access when invalid?:** An unchecked checkbox.
- Buttons:** "Back" and "Save".

The background interface shows a sidebar with "Settings" and "Data" sections, and a main area with a table of actions.

Setting Permission Filter - Use Case

In case you need to share one data set with users from multiple departments and companies there should be created Lookup Data Set which will set which user should see which values. This use case will guide you, how to create mapping data set and set filtering options.

First, let's create mapping data set. Mapping data sets defines, which user belong to entity that should be filtered (g.e. company, department, etc.). In our case, we will have user email and name of the company like following:

User email	Company
user1@user1.com	Company1
user2@user2.com	Company1
user3@user3.com	Company2
user4@user4.com	Company2

Now, we will need to go to the data set which should be filtered, go to Attributes. Select attribute which contains the same values as column Company in the mapping data set, hover your mouse over button Action and select Filter.

The screenshot shows the 'Invoice Table - Attributes' window in BellaDati. The left sidebar contains navigation options like 'Settings', 'Data', and 'Action'. The main area displays a table of attributes. The 'Company Code' attribute is highlighted in yellow. A red arrow points from the 'Action' column of 'Company Code' to the 'Filter' option in the right-hand menu.

Name	Code	ETL column name	Action
Account Assignment	L_ACCOUNT_ASSIGNMENT	ACASS	
Account Number of Vendor or Creditor	L_ACCOUNT_NUMBER_OF_VEND	UNNR	
Accounting Document Number	L_ACCOUNTING_DOCUMENT_NUM	BEINR	
Additional Information (Customer Fields)	L_ADDITIONAL_INFORMATION	ADDINF	
Annotation	L_ANNOTATION	NOTE	
Application	L_APPLICATION	APPL	
Assignment Number	L_ASSIGNMENT_NUMBER	DIQNR	
Barcode	L_BARCODE	BARCODE	
Baseline Date for Due Date Calculation	L_BASELINE_DATE_FOR_DUE	ZBDDT	
Business Area	L_BUSINESS_AREA	GSSER	
Business Transaction	L_BUSINESS_TRANSACTION	BUSCS	
Calculate Tax Automatically	L_CALCULATE_TAX_AUTOMATIC	RAWTI	
Classification	L_CLASSIFICATION	CLASS	
Client	L_CLIENT	MANDT	
Company Code	L_COMPANY_CODE	BUKRS	
Currency Key	L_CURRENCY_KEY	WAERS	
Customer Number 1	L_CUSTOMER_NUMBER_1	KUNNR	
Debit/Credit Indicator	L_DEBITCREDIT_INDICATOR	SWZG	
Deletion Flag	L_DELETION_FLAG	XDELT	
Document Category	L_DOCUMENT_CATEGORY	DOCAT	
Document Date in Document	L_DOCUMENT_DATE_IN_DOCUM	BLDAT	
Document Header Text	L_DOCUMENT_HEADER_TEXT	HRXT	
Document Type	L_DOCUMENT_TYPE	BLART	
Entry Time Stamp - Date	L_ENTRY_TIME_STAMP_DATE	ENTRY	
Entry Time Stamp - Time	L_ENTRY_TIME_STAMP_TIME	ENTRY	

In newly opened window need to be set filter settings.

- Select option **Use attribute filter**.
- Select if values should equal/not equal to selected user field. In our case, there should be selected values that **equal** to current user's **Email**.
- Select mapping data set in the field **Data set with additional user's data**.
- Select which column of the mapping data set will be used as **Attribute corresponding to the selected signed user field**. It will be column User in our case (in this column is stored user's email address).
- Select which column will be used as **Attribute for filtering**. It will be column Company in our case.
- Optionally can be selected option **Allow access when there are no matching values or is invalid?** to allow access for users who don't have any matching department.
- Optionally can be selected option **Allow access when there is a value equals to**. Selecting of this option allows you to set parameter that allow selected user access all the values. Example: this parameter is set to: "***". When there is user who has assigned company called "***" in the mapping table, this user will be able to access all the values.


Creating attribute custom members

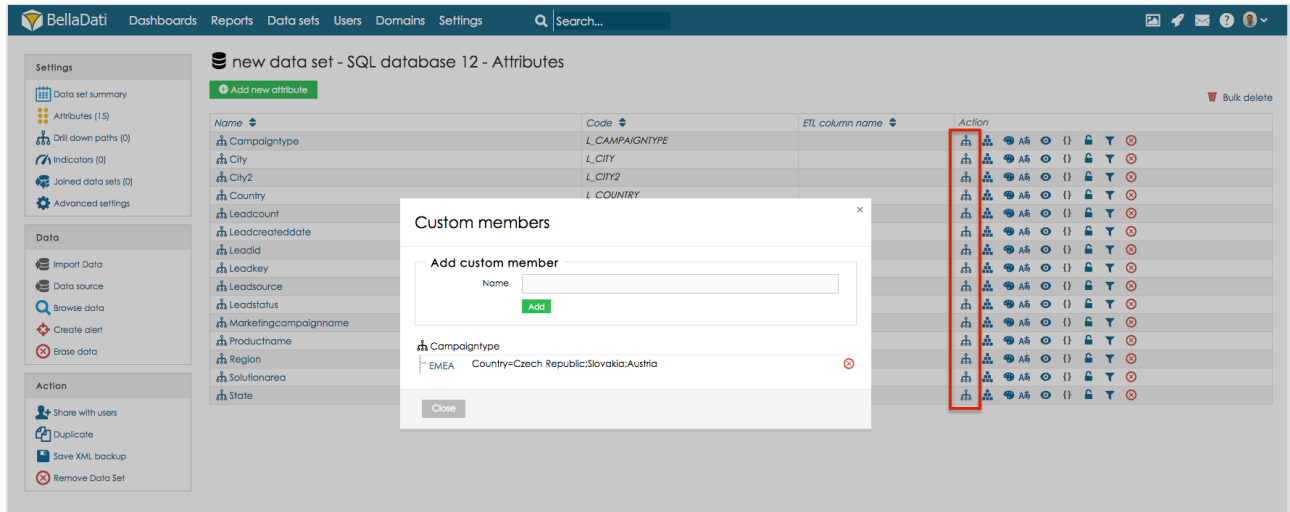
Custom member allows you to add your own nodes into drill down paths. You can add Custom member from *Data set's Attribute page* after clicking on **Custom members** icon in column Action.

Custom member dialog allows you to:






































































































































































- Specify **Name** of the custom member.
- Select **Level** for a custom member.

Every new node requires additional definition. *Custom member definition* dialog allows you to determine attribute values from particular levels which will be aggregated into custom member.

 You can delete created nodes in *Custom member* dialog.



The screenshot shows the BellaDati interface with the 'new data set - SQL database 12 - Attributes' page. A 'Custom members' dialog is open, allowing the user to add a custom member. The dialog has a 'Name' field and an 'Add' button. Below the dialog, a table lists attributes with their codes and ETL column names. The 'Action' column for each attribute contains icons for adding, deleting, and other actions. A red box highlights the 'Action' column icons for the 'Campaigntype' attribute.

Name	Code	ETL column name	Action
Campaigntype	L_CAMPAIGNTYPE		          
City	L_CITY		          
City2	L_CITY2		          
Country	L_COUNTRY		          
Leadcount			          
Leadcreateddate			          
Leadid			          
Leadkey			          
Leadsource			          
Leadstatus			          
Marketingcampaignname			          
Productname			          
Region			          
Solutionarea			          
State			          

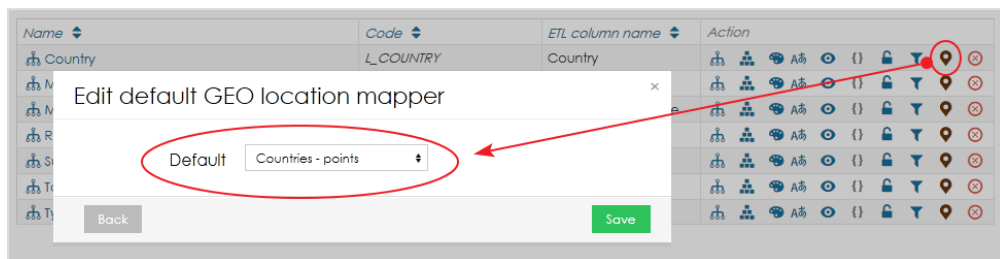
Editing default GEO location mapper

As in chart and tables, you can set **another drill down** to change the granularity of map. To do so you have to specify **default GEO location mapper**.



To get more information about setting custom drill down continue by [Creating Geo Maps](#)

To edit default GEO mapper, click on GEO point icon and select mapper of your choosing.



You can download video of editing default GEO location mapper [here](#).

Joining Data Sets

Joined data sets allow you to analyze data from more data sets together and therefore use data from more [data sources](#) in a single report.

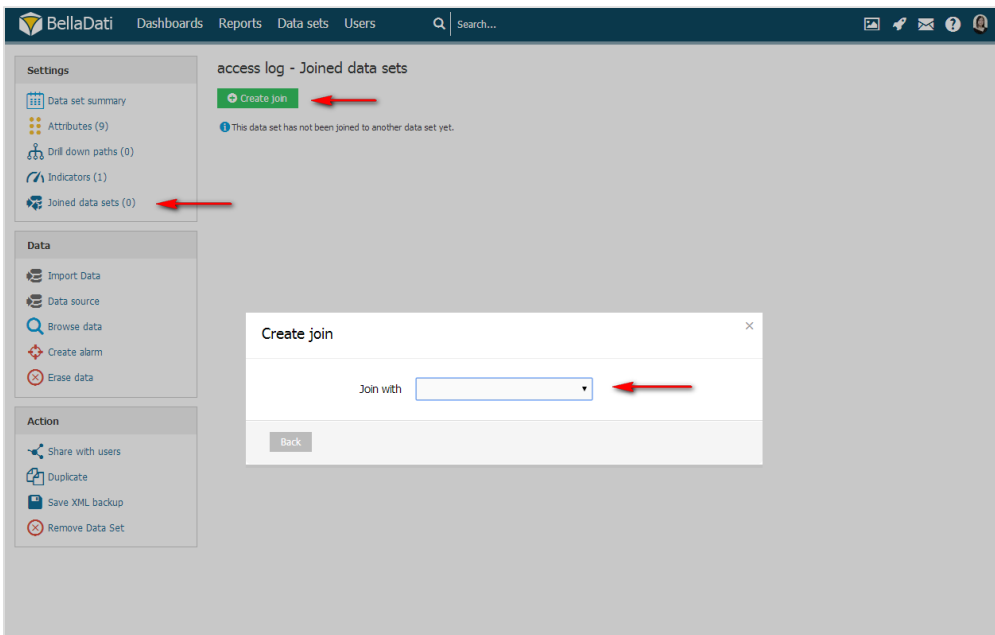


The principle is similar to joining SQL database tables.

Creating Join

To create join in BellaDati:

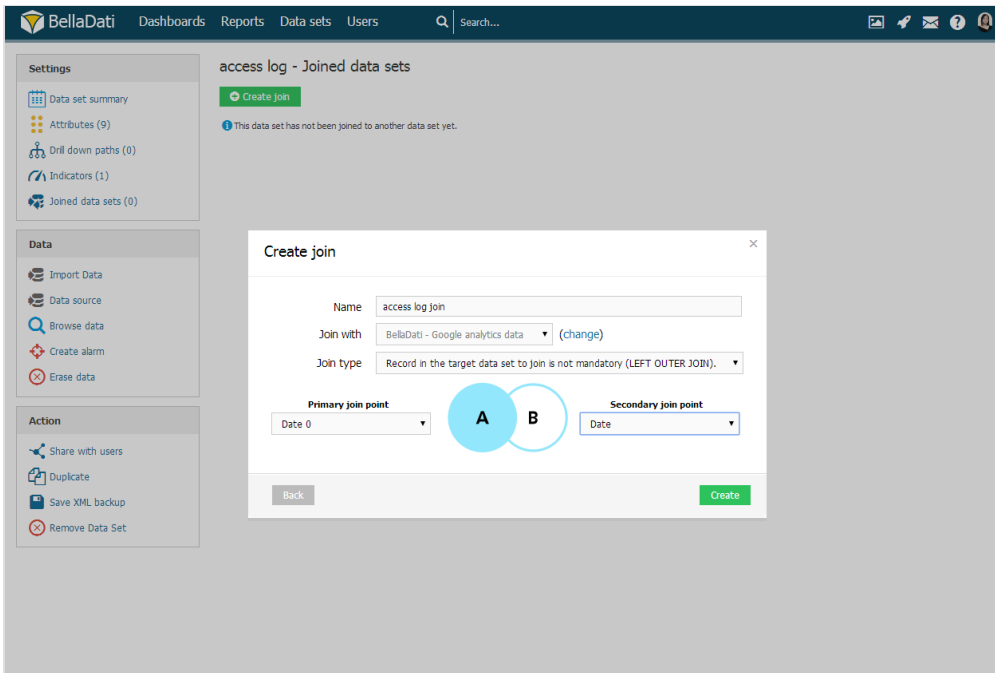
1. Navigate to one of the **Data Sets** you want to use in join
2. Click on **Joined Data Sets** in **Settings** section of the left action bar
3. Choose **Create Join**
4. Select **Data Set** to be joined with



Join condition

Each join has to have specified following parameters:

- **Name:** Joined data set name.
- **Join with:** Name of the source data set to join with.
- **Join condition:** Date/time in both records are taken in account or not.
 - With Time Match: Date/time in both records are taken in account.
 - Without Time Match: Date/time in both records are **not** taken in account.
 - Custom: Cross join with option to specify **own condition**.
- **Join type:** Standard join types available - match always depends on **attributes**.
 - Left outer join: Record in the target data set is not mandatory.
 - Inner join: Record in the target data set is mandatory.
 - Cross join: No attribute match required.



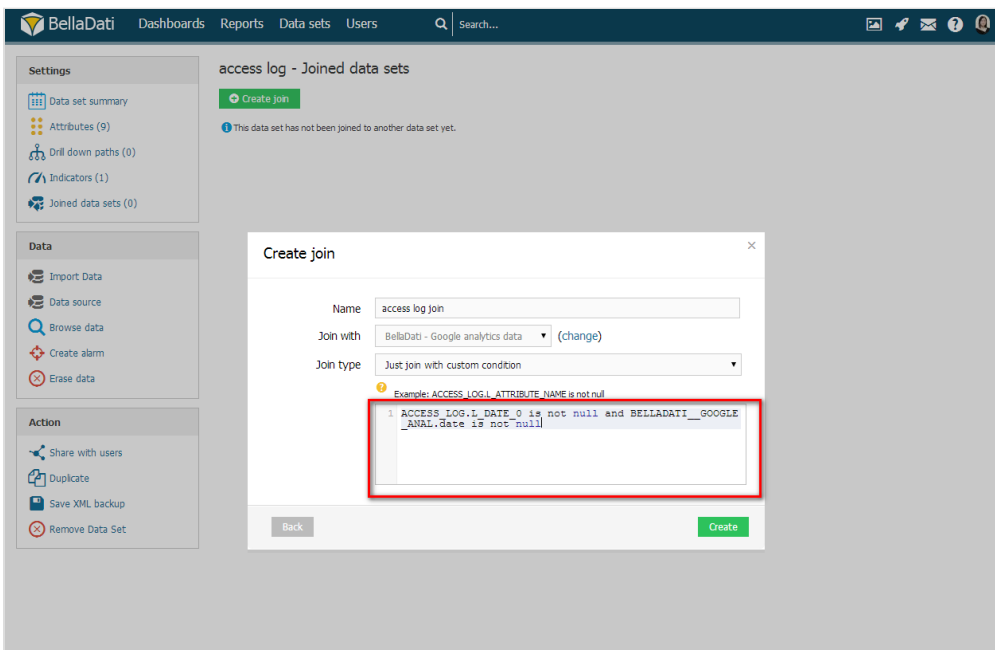
The resulting joined data set will contain all the attributes and indicators from all source data sets.

Custom Join Condition

BellaDati allows you to define you own **joining condition**. To do so:

1. Select **Custom** in **Join Condition**
2. **Cross join** will be automatically applied
3. Type **joining condition** to restrict the output

BellaDati offers you **autocomplete** to easily construct joining conditions.



Multiple Join Points

One data set could be joined with more data sets (eg. join by department ID to get department full name, address, country, total sales and then join by product ID to get products name, price, weight and mass). Each join is called **join point**.

To add another **joining point** into existing join:

1. Click on **add joining item**
2. Select desired **Data Set**
3. Continue with **Join Conditions**

BellaData
Dashboards Reports Data sets Users
Search...

Settings

- Data set summary
- Attributes (16)
- Drill down paths (0)
- Indicators (6)
- Joined data sets (1)

Data

- Browse data
- Create alarm

Action

- Share with users
- Save XML backup
- Remove Data Set

access log join - Joined data sets

⚠ Data set has been changed and needs to be rebuild. (run now)
Rebuild will start shortly
Disable automatic rebuild

ℹ This data set is a join of access log and following data sets:

+ Add joining item
- Delete all joins

Target data set	Primary join point	Secondary join point	Join type	Joining condition
BellaData - Google analytics data	⚠ Attributes match is not required		<div style="text-align: center;"> <div style="border: 2px solid lightblue; border-radius: 50%; width: 60px; height: 60px; display: flex; align-items: center; justify-content: center; margin: 0 auto;">A</div> <div style="font-size: 2em; margin: 0 10px;">+</div> <div style="border: 2px solid lightblue; border-radius: 50%; width: 60px; height: 60px; display: flex; align-items: center; justify-content: center; margin: 0 auto;">B</div> </div> <p>Just join with custom condition</p>	ACCESS_LOG_ID_DATE_0 is not null

Data preview

Records count: 10000

[Date] [BellaData - Google analytics data]	Date 0 [access log]	Time 0 [access log]	Browser [BellaData - Google analytics data]	City [BellaData - Google analytics data]	Country [BellaData - Google analytics data]
1/1/2011	8/29/2012	11:58:54 PM	Firefox	Prague	Czech Republic
1/1/2011	8/29/2012	11:58:54 PM	Internet Explorer	Dublin	Ireland
1/1/2011	8/29/2012	11:58:54 PM	Safari	Prague	Czech Republic
1/2/2011	8/29/2012	11:58:54 PM	Firefox	Prague	Czech Republic
1/2/2011	8/29/2012	11:58:54 PM	Firefox	Prague	Czech Republic
1/3/2011	8/29/2012	11:58:54 PM	Chrome	Prague	Czech Republic
1/3/2011	8/29/2012	11:58:54 PM	Chrome	Prague	Czech Republic
1/3/2011	8/29/2012	11:58:54 PM	Firefox	Prague	Czech Republic
1/3/2011	8/29/2012	11:58:54 PM	Firefox	Prague	Czech Republic
1/3/2011	8/29/2012	11:58:54 PM	Internet Explorer	Prague	Czech Republic
1/4/2011	8/29/2012	11:58:54 PM	Chrome	Prague	Czech Republic
1/4/2011	8/29/2012	11:58:54 PM	Chrome	Prague	Czech Republic
1/4/2011	8/29/2012	11:58:54 PM	Chrome	Prague	Czech Republic

Joining Facts and Restrictions

- 1. Data in the joined data set will be the intersection of all joins.
- 2. Data in joined data sets are updated automatically when data are changed or imported to the source data sets. You cannot import data directly to the joined data sets.

1. Please note, that the permissions setting based on data filter has to be set for both (source and joined) data sets!
2. Data join is also available on data source level (database). Please consider this option when you plan to analyze millions records of data. Joining on database level may provide better performance.

Joined data set supports these functions the same way like in ordinary data set:

- Data browsing and export
- Data changes watching
- Structure backup

Changing join point

You are allowed to perform following actions on existing joined data set:

- Changing of join type
- Changing of date/time matching condition
- Deleting the join point (or deleting all join points)
- Adding new join point
- Disable/Enable automatic rebuild of data set

BellaDati

Dashboards

Reports

Data sets

Users

Search...

Settings

Data set summary

Attributes (25)

Drill down paths (0)

Indicators (4)

Joined data sets (2)

Data

Browse data

Create alert

Action

Share with users

Save XML backup

Remove Data Set

Shared Wallet Survey join - Joined data sets

This data set is a join of Shared Wallet Survey and following data sets:

Add joining item

Delete all joins

Disable automatic rebuild

Target data set	Primary join point	Secondary join point	Join type	
<div>Product_xzGAIL</div>	<div>product_id [Shared Wallet Survey]</div>	<div>product_id [Product_xzGAIL]</div>	<div><div><div>A</div><div>B</div></div><div>Record in the target data set to join is not mandatory (LEFT OUTER JOIN).</div><div>Change to: <div>Inner Join</div> <div>Cross Join</div></div></div>	<div></div>
<div>Customers with complete CRM</div>	<div>customer_id [Shared Wallet Survey]</div>	<div>ID [Customers with complete CRM]</div>	<div><div><div>A</div><div>B</div></div><div>Record in the target data set to join is not mandatory (LEFT OUTER JOIN).</div><div>Change to: <div>Inner Join</div> <div>Cross Join</div></div></div>	<div></div>

Data preview

All changes will be propagated to joined data and related reports or dashboards immediately.

BellaDati Documentation, Version 2.7.11.5

Page 134 of 370

Building joined data set

Building process of data sets is triggered by the data or structure change in the underlying data sets (when automatic rebuild is enabled) or manually by user. During the building process, all referenced data sets are locked for performing changes.

Disabling the building process

Disable process for particular data set

There are situations, when we don't want to start the building process automatically, especially in the "big" data sets. After the building process has been triggered, you can disable it in the information box on the top of the data set overview page:

Target data set	Primary join point	Secondary join point	Join type
Branch	Age Group [Customers with complete CRM]	City [Branch]	Inner Join
Customers with complete CRM	customer_id [Shared Wallet Survey]	ID [Customers with complete CRM]	Inner Join

Disable process for all data sets in specific time interval

This feature is available in On-Premise version only

In specific cases, for example if there are many joined data sets build on several daily updated data sets, each change of the underlying data set triggers the building process. It may cause "locking" errors when the system will try to import data into another referenced data set which is part of the join. For this situations, you can disable the building process for the specific time interval:

Configuration

General

Property Name

- Maximum upload file size (in bytes)
- Max failed logins count
- Suspended materialization interval (H:mm)
- Email sender/recipient
- Application URL
- Google login enabled

Property Value

- localhost

Edit

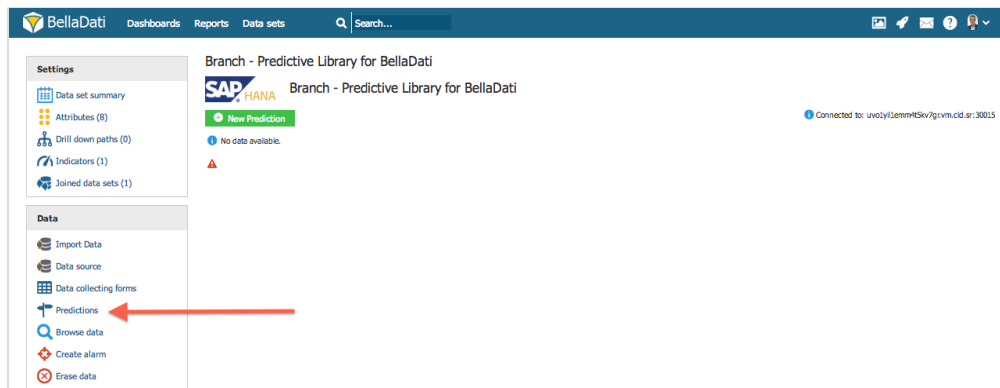
Suspended materialization interval (H:mm)

Back Save

Working with SAP HANA Predictions Manager

BellaDati allows you to create predictions leveraging [SAP HANA PAL](#) library.

Predictions are created and managed in **SAP HANA Predictions Manager**. To enter **Manager** navigate to appropriate Data Set ale select **Predictions**.



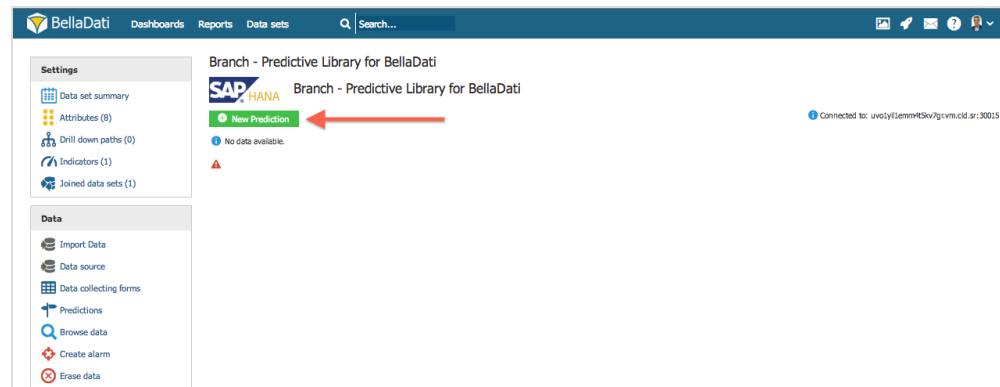
Creating Predictions

Click New Prediction to open dialog window. It allows you to define prediction's:

- Name
- Function
- Parameters (depends on selected Function)
- Execution

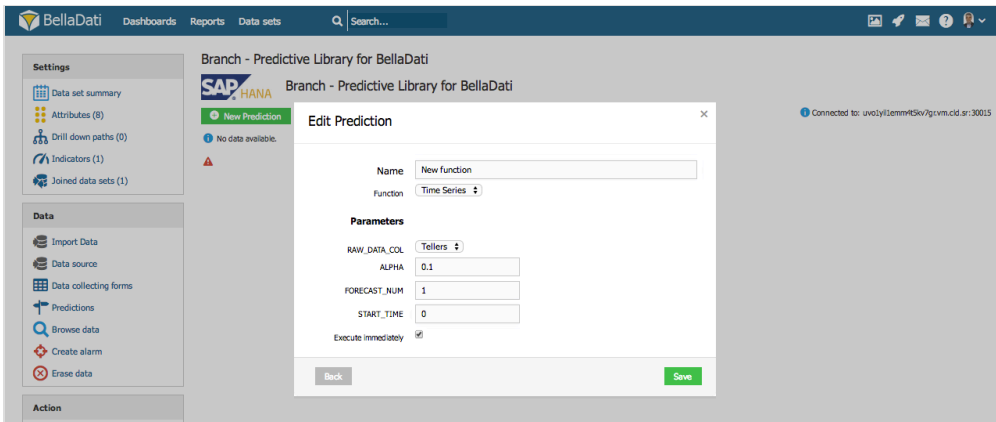


Please refer to [SAP HANA PAL](#) documentation for explanation of particular parameters and allowed values.



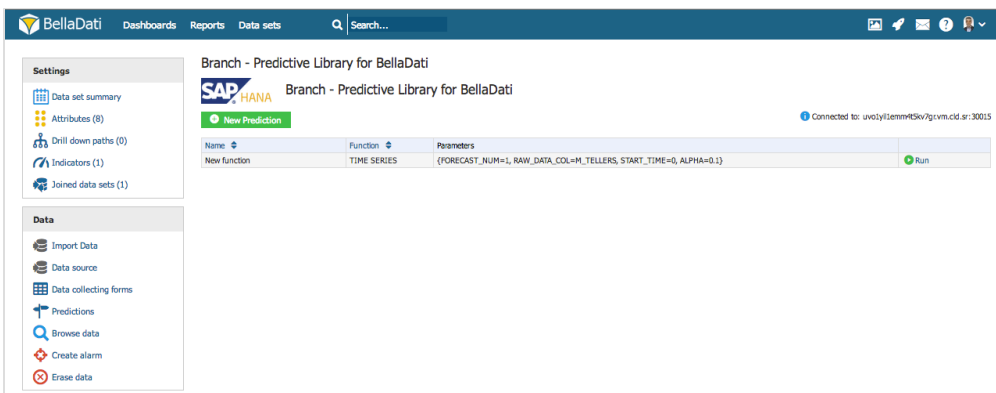
Executing Prediction

You can execute prediction immediately by checking **Execute immediately** or manually from **Predictions Manager** by hitting **Run** button.



Editing Prediction

To edit existing prediction, hover over its name and click **Edit** button. It opens prediction dialog with defined parameters.



Structure Backup

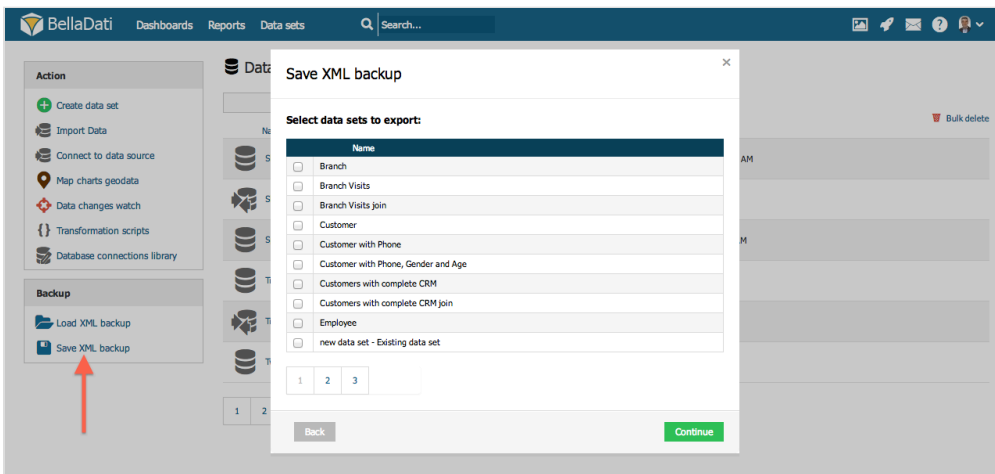
This function allows you to backup existing structures in BellaDati and migrate them to another instance (eg. from Cloud to On-Premise).

There are two types of BellaDati structures backup:

- [Backup of the whole domain structure](#)
- Data set and report structure backup - described below
- [Report backup only](#)

The following structures are included in this backup:

- **Data sets:** attributes, indicators, indicator groups and their settings (appearance, format, formulas), data set [owners](#), sharing settings
- Data source settings
- Alarms
- Joins
- **Reports:** Views (tables structure, chart types and their structure), report indicators, report [owners](#), sharing settings.



There are two ways of this structure backup:

- Selected data set and reports based on its data backup.
- Bulk backup of more selected data sets, their settings and related reports.



When exporting data sets created by joins, all basic data sets will also be automatically exported.

Exporting XML Structure



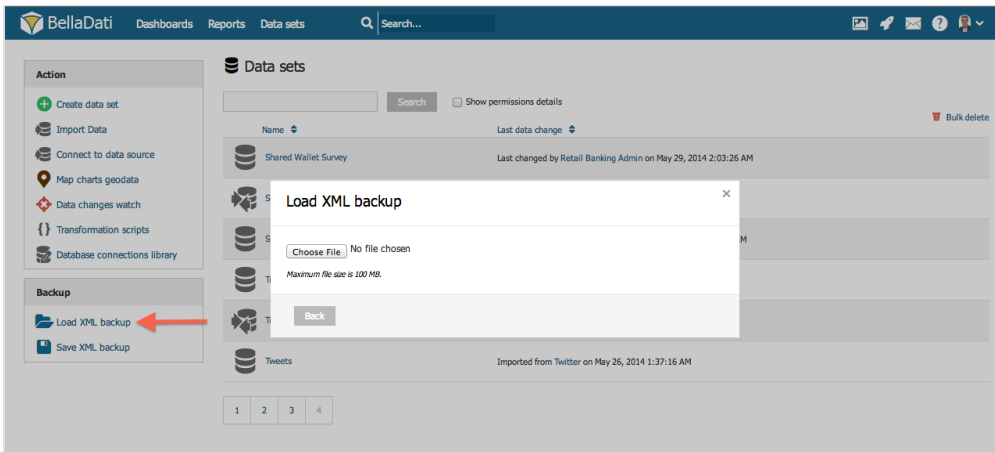
Data are not exported. Use separate [exporting data](#) function to backup data!



Import settings (templates), dashboards, users and user groups are not included in XML structure backup. We recommend to create users, user groups and assigning roles to them before importing XML backup. See also [user import](#) feature.

Importing XML Structure


A wizard is available during XML structure import. The preview of imported data sets and report parameters is displayed.



Several checks are performed during XML structure import:

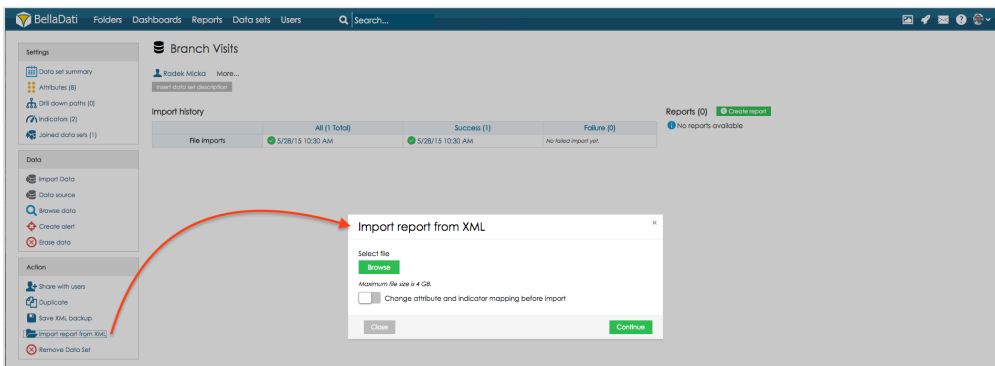
- User and user groups: You can select another users or user groups from the existing accounts in the domain. *Note: All data sets and reports must have owners!*
- Indicator and attribute assign.

You can manually exclude some data sets and reports from the imported XML structure.

 Existing XML structure can be modified - see [Setup Data Model using XML](#) for details.

Importing report

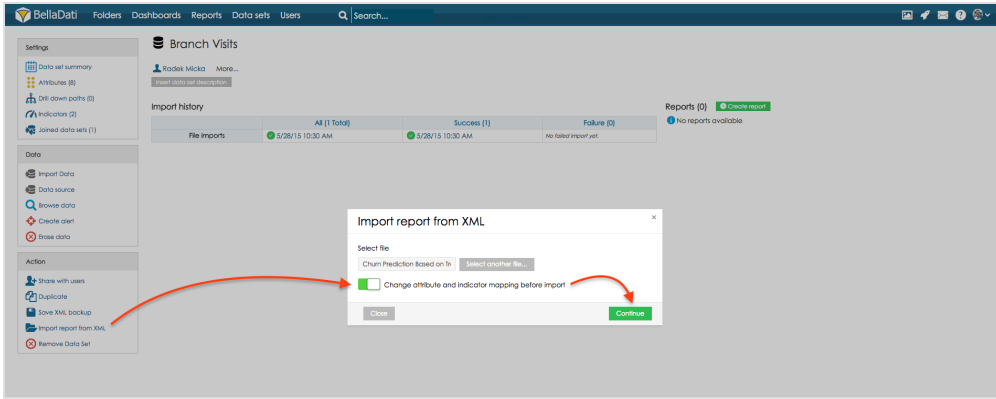
Open data set on which imported report was created. Select option *Import report from XML* to display import wizard.



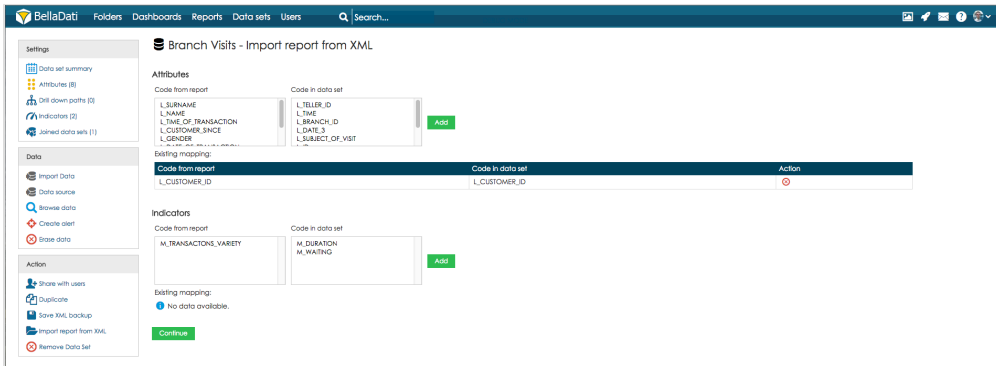
Select XML file with report export and click *Continue* to import report.

Importing report based on different data set

Open data set to which the report will be imported. Select option *Import report from XML* to display import wizard. Select XML file with report export, select option *Change attribute and indicator mapping before import*, click button *Continue* to proceed to mapping options.



Map existing attributes and indicators from the report to attributes and indicators from the data set. To do this, select attribute (or indicator) code from the column and relevant code from the data set. By clicking button Add, there is created reference between these two codes. When there is created reference for all the attributes and indicators from the report, click button Continue to import the report.



Sharing Data Sets



Data set sharing functions are only available for the owners of the particular data set.

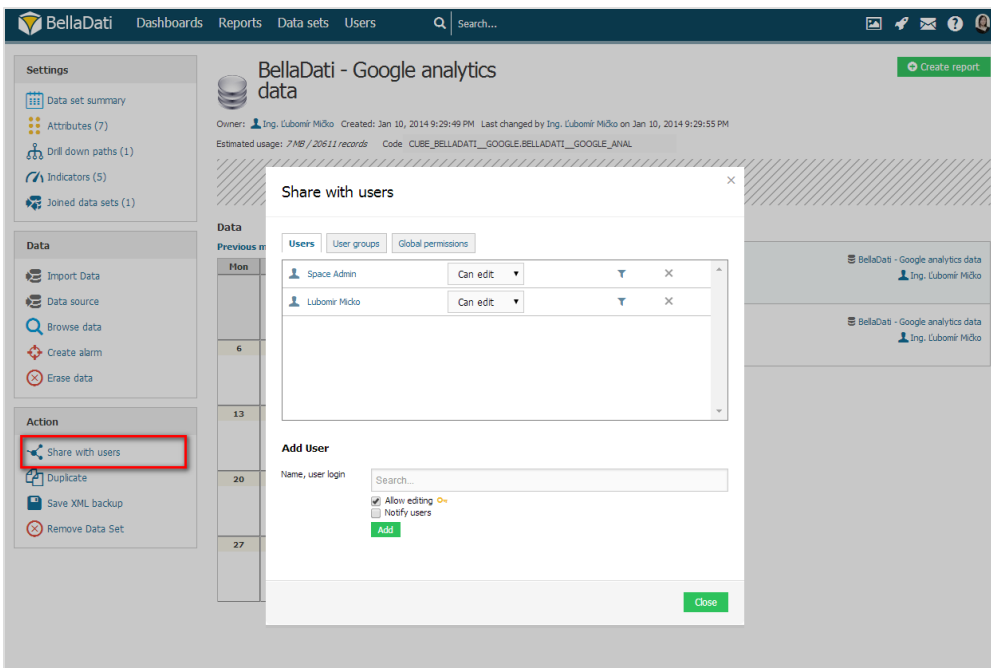
Data set sharing functions allows you to perform following actions:

- Grant access to data for selected [users](#) or [user groups](#)
- Allow access to data for all users in the domain globally
- Optionally notify users about granted access to data sets
- Restrict access to shared data at data level by defining the data filter for users and user groups

When you are setting up data set sharing for individual users or user groups, please distinguish following two [access levels](#):

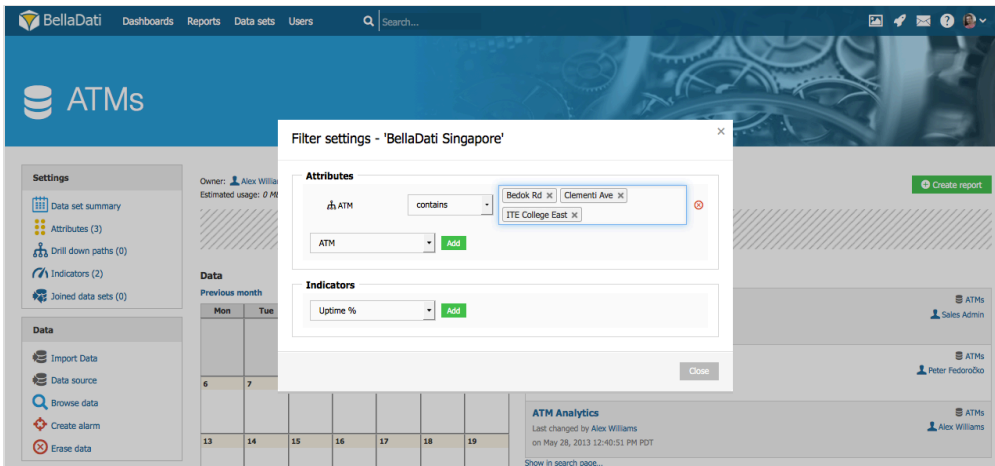
- **Read-only access:** Only reports or dashboards can be created.
- **Full access:** All functions except sharing and data set deleting are available.

As soon as the data set is shared with the user, he is able to create report and then dashboards based on shared data. In addition to that, users with full access can also manage the data set in the same way as it's owner except sharing settings or complete removal.



Data Set Filter

You can restrict access for selected users or user groups at data level. Eg. large companies can have more SBUs requiring the same data however each SBU management should have access only to the data concerned with their SBU. On the other side general management can still access aggregated data for all SBUs. Data set filter function allows you to restrict sharing only to records that contains selected members (eg. SBU or department name). All aggregations for these users is then available only on these data.



Data set filter functions exactly the same as for [report](#).



Data set member filter is not available for global data set sharing.
Always thoroughly plan member filter settings in connection with [user roles and permissions](#)! Otherwise data leakage may occur when sharing reports or dashboards based on this data set.

Data set sharing via Sharing console

BellaDati also allows you to see the list of data sets with users and user groups the data set is shared with. The list contains separate columns for data set owner, editors, read-only users, group editors, read only groups.



Only [Domain administrator](#) is allowed to access the Sharing console.

You can find Sharing console in the Users section. To access Sharing console:

1. Point to the Users in the main menu and click the Sharing console item.
2. Choose Data sets in the Actions box

To change the sharing settings click button Edit for selected data set.

Name	Owner	Editor	Read-only users	Editors groups	Groups for read-only	View-access for everybody in domain	Action
123	Radek Micka2		James Bond				Edit
12345	Radek Micka2						Edit
1.Booking	Radek Micka2						Edit
1.Booking_Mingli	Radek Micka2						Edit
1.Partner	Radek Micka2						Edit
1.Partner_atmPM	Radek Micka2						Edit
1.Product	Radek Micka2						Edit
1.Product_gOnq	Radek Micka2						Edit
1.Step Conversion	Radek Micka2						Edit
1.Step Conversion_JAAXq	Radek Micka2						Edit
3.Amenities	Radek Micka2						Edit
3.Bookings	Radek Micka2		tester testovic				Edit
3.Bookings_cuFOZ	Radek Micka2						Edit
3.Bookings with Customers and hotels	Radek Micka2						Edit
3.Bookings with Customers and hotels_10aG	Radek Micka2						Edit
3.Customer Requests	Radek Micka2						Edit
3.Customer Requests join	Radek Micka2						Edit
3.Customers	Radek Micka2						Edit
3.Customer Segmentation	Radek Micka2		tester testovic				Edit
3.Customer Segmentation_gkXfkg	Radek Micka2						Edit
3.Customers_voGfc	Radek Micka2						Edit
3.hotels	Radek Micka2						Edit
3.hotels_GEaEd	Radek Micka2						Edit
000	Radek Micka2						Edit
AAA-EngagementPortfolio	Radek Micka2						Edit
AAA-ProjectList	Radek Micka2						Edit
AAA-ProjectList_wllqz	Radek Micka2						Edit

Watching Data Changes

This function is also known as **alarms**. Data watching function allows to monitor actual data in the data set and launch alerts when the values fulfil predefined conditions.

One data set can contain more alarms.

Alarm parameters:

- **Alert Name**
- **Date attribute:** Data will be monitored by this date attribute.
- **Date interval:** Data for this period are aggregated and checked against the condition.
- **Alert conditions:** Alarm condition.
 - Indicator and it's aggregation.
 - Condition: equal, lower, greater, decrease , increase.
 - Value: Comparison to this absolute value.
- **Send e-mail:** When alert is launched, user will be also notified by e-mail. By standard, user is notified in alert actions list or via recent changes dashlet on dashboard.
- **Check on data change:** Alert will be checked on every data change.

Following actions are available in alarms list:

- **Check now:** Allows user to execute condition checking for the particular alarm manually (see also "Continuous check above").
- **Filter:** Set values of the attributes and indicators which will be monitored.
- **Settings:** Allows user to change alarm settings.
- **Delete:** Completely removes alarm from data set.
- **Deactivate watching:** Condition checking and alerts will be disabled when the alarm is suspended.
- **Activate watching:** Changes suspended alarm to active state - condition checking will be restored.

The screenshot displays the BellaDati application interface. A modal window titled "Create alert" is open, allowing the user to configure a new alert. The alert name is "Sales are too low". The date interval is set from 11/1/2014 to 11/30/2014. The alert condition is configured to trigger when the sum of sales is lower than 100,000. The user has opted to receive email notifications and to have the alert checked on every data change. The background shows the main dashboard with a sidebar containing navigation options like Settings, Data, and Action. The "Create alert" option in the Action section is highlighted with a red box. A report titled "Coffee" is also visible in the background.

Managing GEO Data



Only users with data manager role can manage GEO points, regions and map providers. See [BellaDati permissions and roles](#) for details.

BellaDati supports data visualisation on geo maps by using two options:

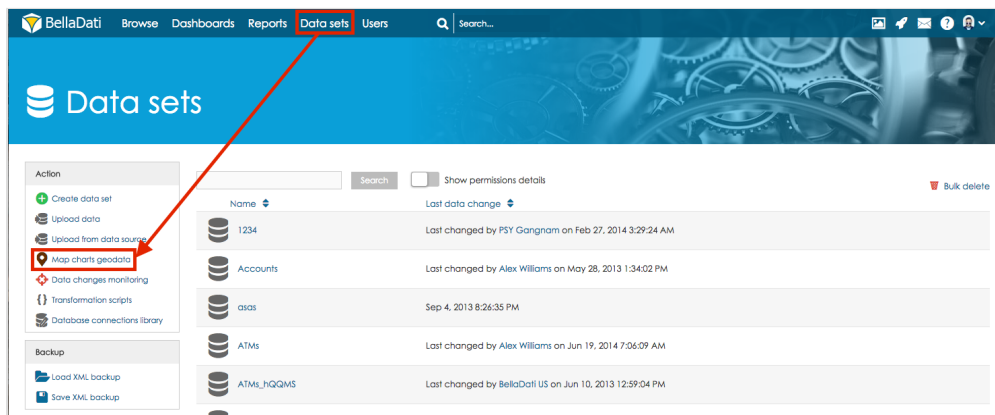
- GEO Points
- GEO Shapes

BellaDati supports data visualisation on geo maps by using two options:

- GEO Points
- GEO Shapes

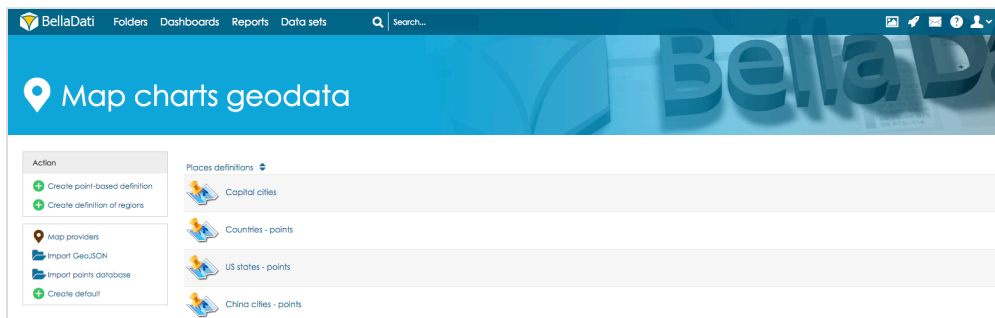
Map charts geodata screen

In the **Data set menu** screen click on the **Map charts geodata** to manage GEO points, regions and map providers.



Available actions on **Map charts geodata** screen are:

- **Create point-based definition**
- **Create definition of regions**
- **Map providers:** Use your custom GEO map provider.
- **Import GeoJSON:** Allow to import predefined geo regions from file using [GeoJSON format](#).
- **Import points database:** Allow to import predefined geo points from CSV file in structure: point name, longitude, latitude, additional names
- **Create default:** Will create default set of regions and points (countries, world capital cities, US states). That are delivered by default as part of BellaDati installation.

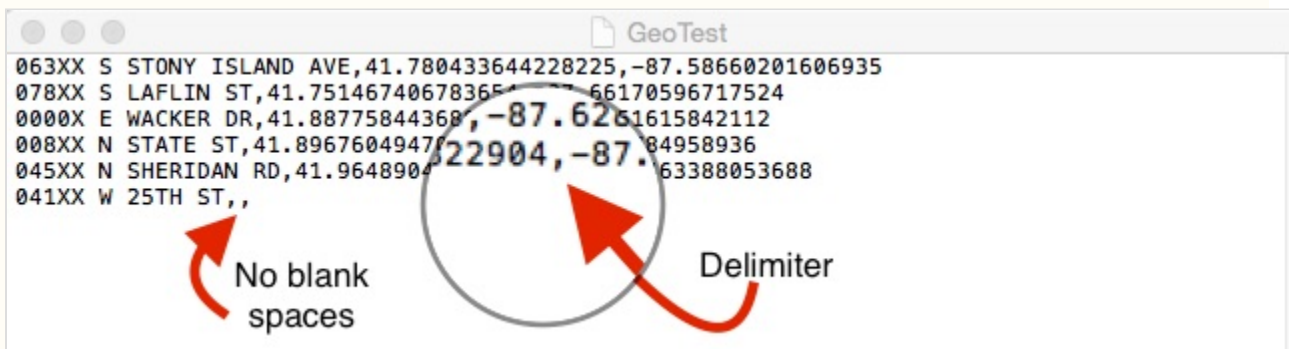


Uploading GEO Points definition



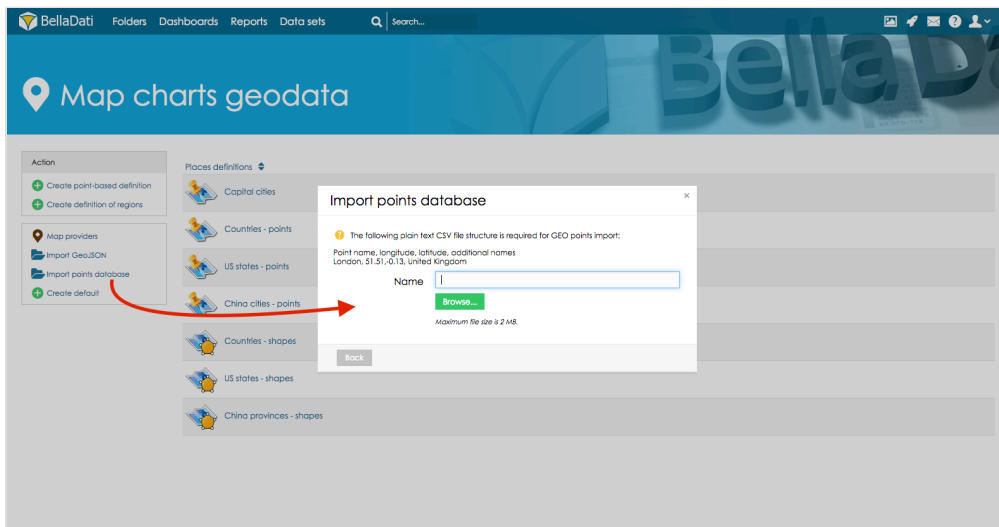
GEO Points file format

Your CSV file with GEO location points must have exact format on picture below. **Please note the delimiter type and no blank spaces are accepted.** Each GEO point is defined by its latitude and longitude coordinantes. **You have to define associated drill-down values to match drill-down members in the report view.** You can associate more drill-down values to single GEO point definition. Upper and lower case are distinguished (eg. New York, new york, NY are different values).



This picture includes GEO POINTS for streets of Chicago city as following - **Street Name, longitude, latitude (longitude and latitude exactly in this order)**

1. In the **Map charts geodata** screen click on the **Import points database** and import CSV file in format described in the note section



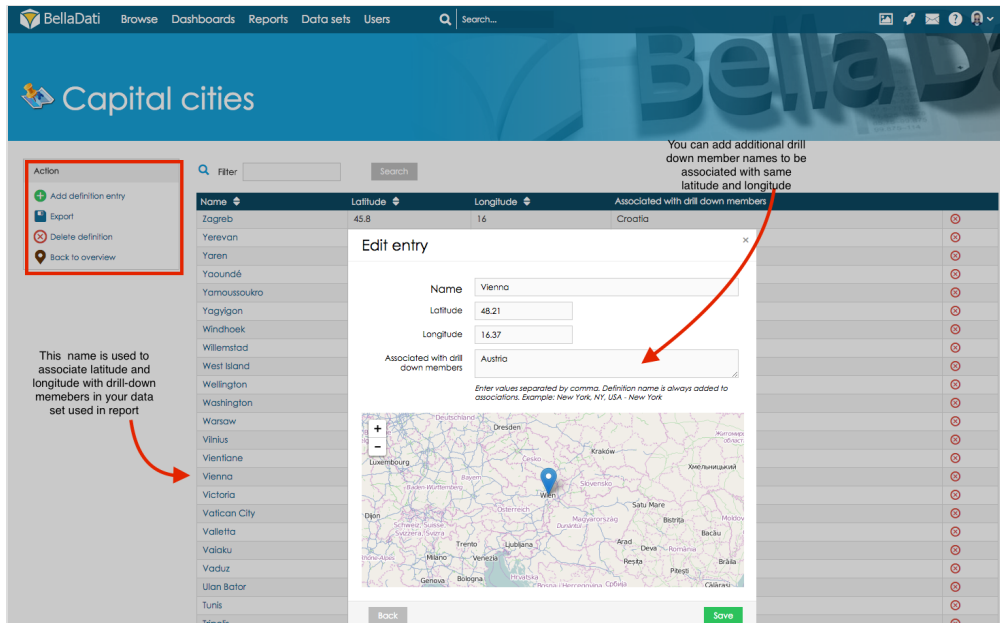
2. Click on the imported points data to edit data or proceed to report to create New Map

Following options are available

- Add definition entry (to add new latitude and longitude definitions into the exiting definition)
- Export option will export existing definition into the NameOfExistingDefinition.csv
- Delete definition will delete existing definition
- Back to overview will return user to overview of all available definitions

On the screen of the particular GEO points definition user can edit following parameters

- Name,
- Latitude
- Longitude
- Associated drill-down values (multiple parameters can be inserted by using , as an delimiter)



Associate with other drill-down members

Name of the GEO point will be associated with drill-down members in your data set used to create new map in report. Using **Associated with drill-down members** section you can associate latitude and longitude as well with other drill-down member. Example - City of Vienna has got associated latitude and longitude in Geo Points definition. If your data set includes as well state Austria. You can define it in **Associated with drill-down members** section and same latitude and longitude will be associated with Austria.



Tip how to upload geo data other way

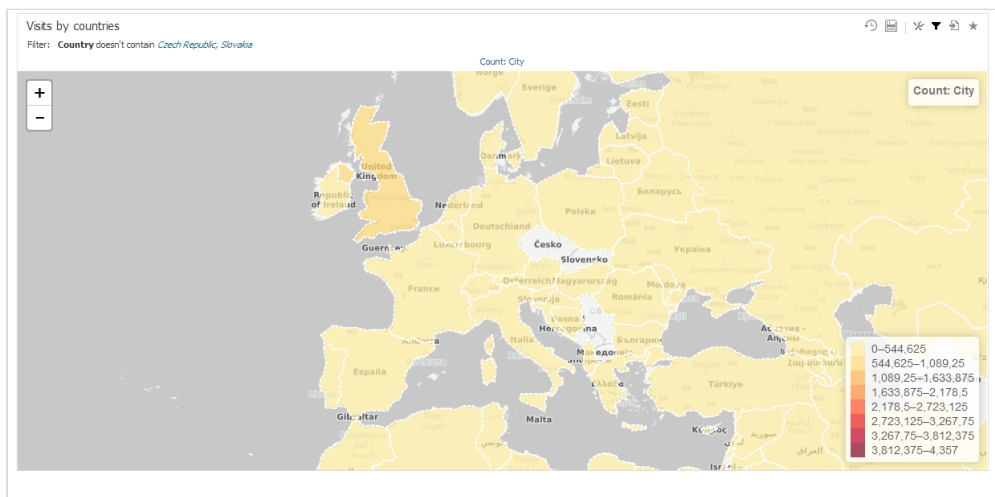
Include the GEO point definition in the regular data import mapping the longitude/latitude to the GEO Point attribute type. But if you need to use **Show details for** option during creating new maps in report import GEO data through Map Charts Geo Data section as described below.

GEO Regions

Each GEO region is defined by three or usually more points specified by latitude and longitude coordinates. Standardized GeoJSON format is supported to simplify importing these definitions. **You have to define associated drill-down values to match drill-down members in the report view.** You can associate more drill-down values to single GEO region definition. Upper and lower case are distinguished (eg. Canada, canada, CAN are different values).

Following parameters are manageable for GEO regions:

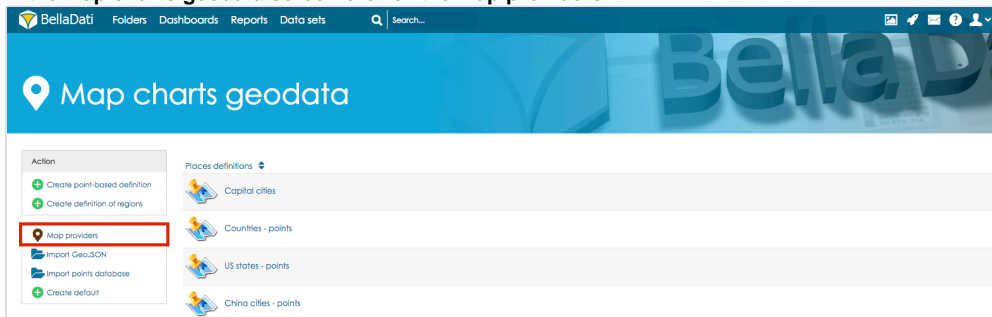
- Definition name
- Region parameters: Name, GeoJSON coordinates, Associated drill-down values



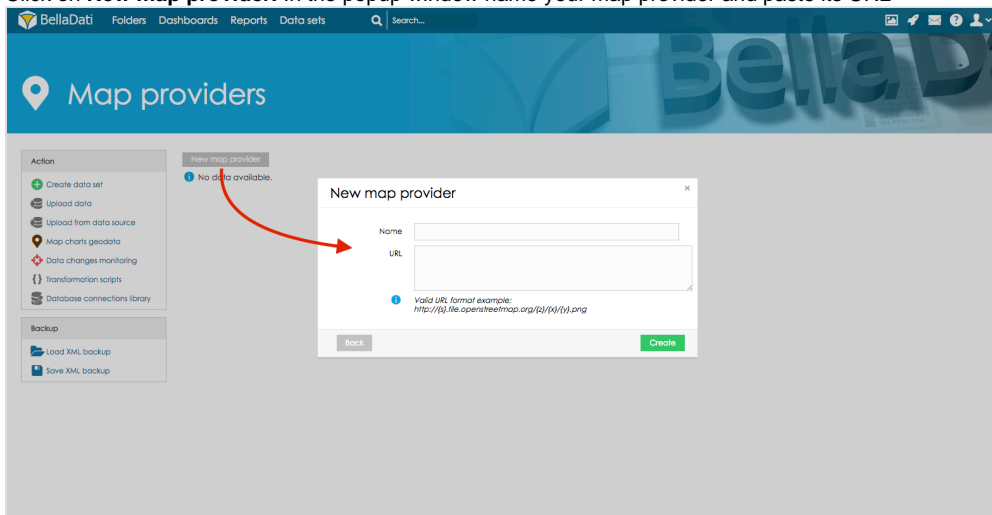
Map providers

Custom map providers are supported. You can add your own map provider to use in map charts.

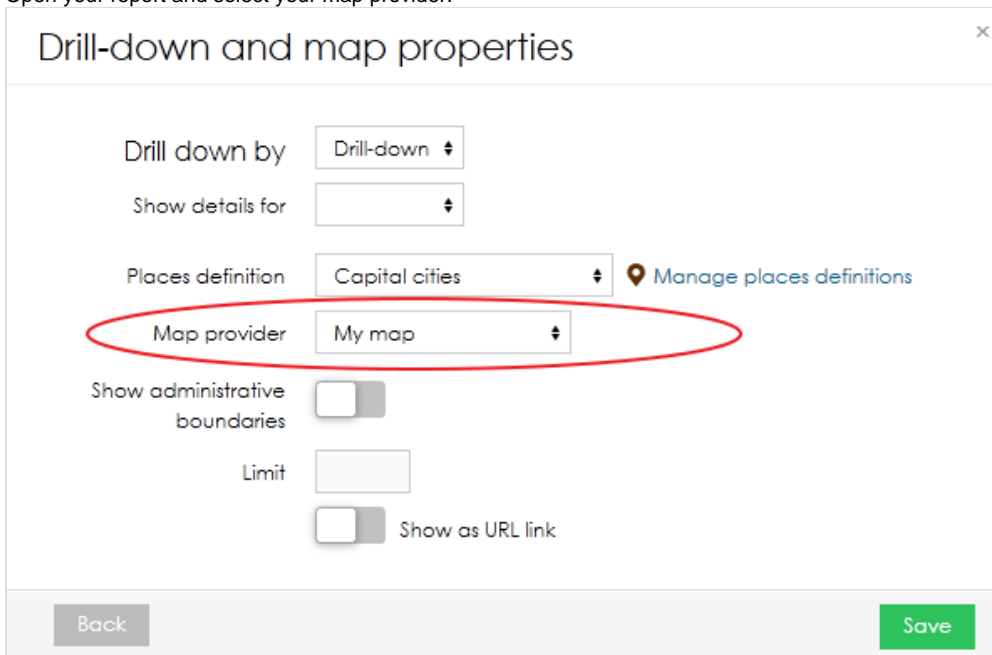
1. In the **Map charts geodata** screen click on the **Map providers**



2. Click on **New map provider**. In the popup window name your map provider and paste its URL



3. Open your report and select your map provider.



Follow the valid URL format example using the OSM standard. You can refer to [OpenStreetMap wiki](#) for more information (section "Tile servers"). In case you are using only one server, parameter **{s}** is not used.

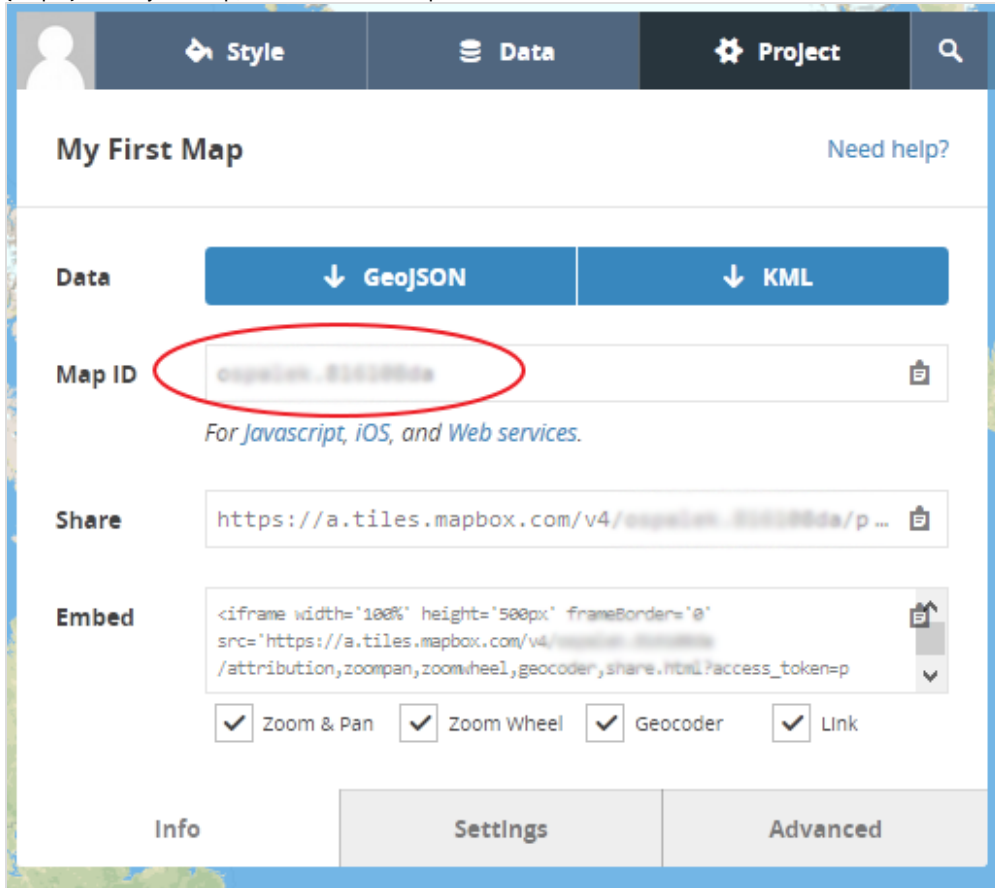
Using Mapbox as tile server

Mapbox is a mapping platform which can also be used as a map tile server in BellaDati.

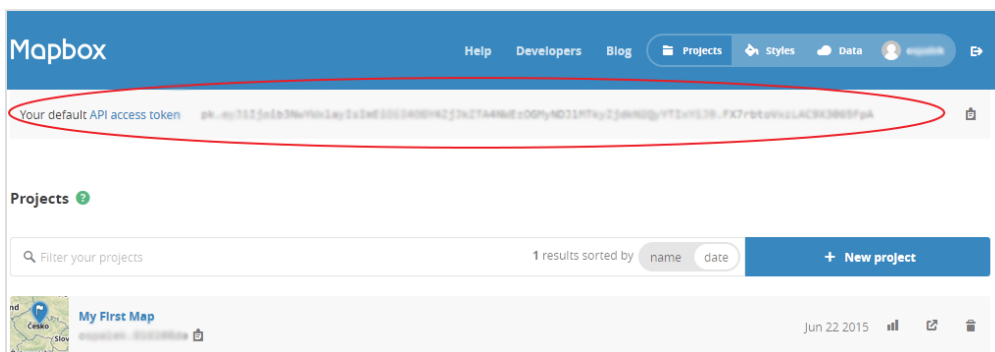
1. Create new account [here](#).
2. Create new map with desired settings.
3. In BellaDati click on **New map provide** and this URL:

`http://api.tiles.mapbox.com/v4/{mapid}/features.json?access_token=<your access token>`

- a. {mapid} - ID of your map, can be found in map info



- b. <your access token> - access token that can be found on Projects page



Creating own map tile server



Own map tile server is very resource-demanding and requires powerful server with big storage. In most cases it is better to use default map provider.

We suggest using this [tutorial](#) for installing and configuring your map tile server. It is compatible with Ubuntu (tested on version 14.04). More info can be found [here](#):

- <https://switch2osm.org/serving-tiles/using-an-all-in-one-solution/>
- <https://switch2osm.org/serving-tiles/manually-building-a-tile-server-14-04/>
- <https://switch2osm.org/serving-tiles/building-a-tile-server-from-packages/>

Maps can be downloaded from these links:

- <http://planet.openstreetmap.org/> - map of the whole planet - 28 GB
- <http://download.geofabrik.de/index.html> - maps of regions. In case only one country is needed, clicking on region name will open list of countries.

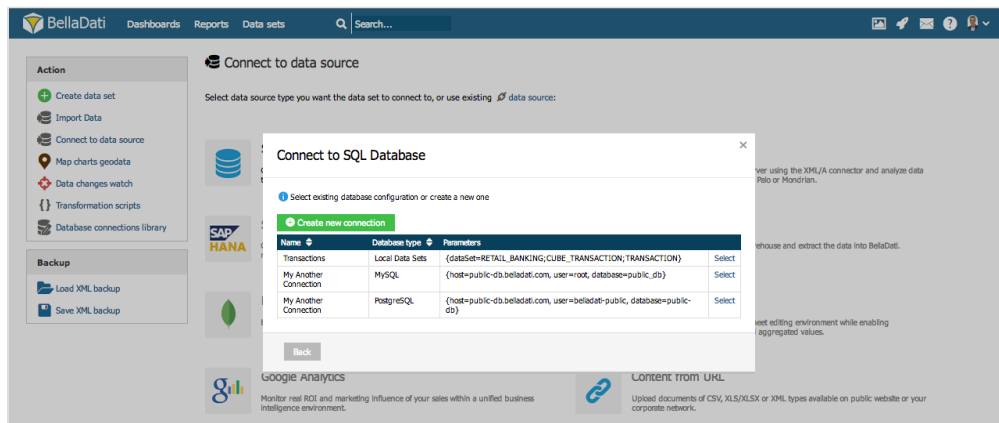
After successful start of server, copy the URL (or IP adress) and paste it in BellaDati in correct format (see [Map Providers](#) above)

Database Connections Library

BellaDati enables you to **invoke existing connection** when connecting to [SQL Data Sources](#).

After selecting SQL Data Source, you can either:

- create new connection by clicking on **Create new connection**
- use existing connection by clicking **Select** link (connections existing in the domain are available to all domain admins)



Reports

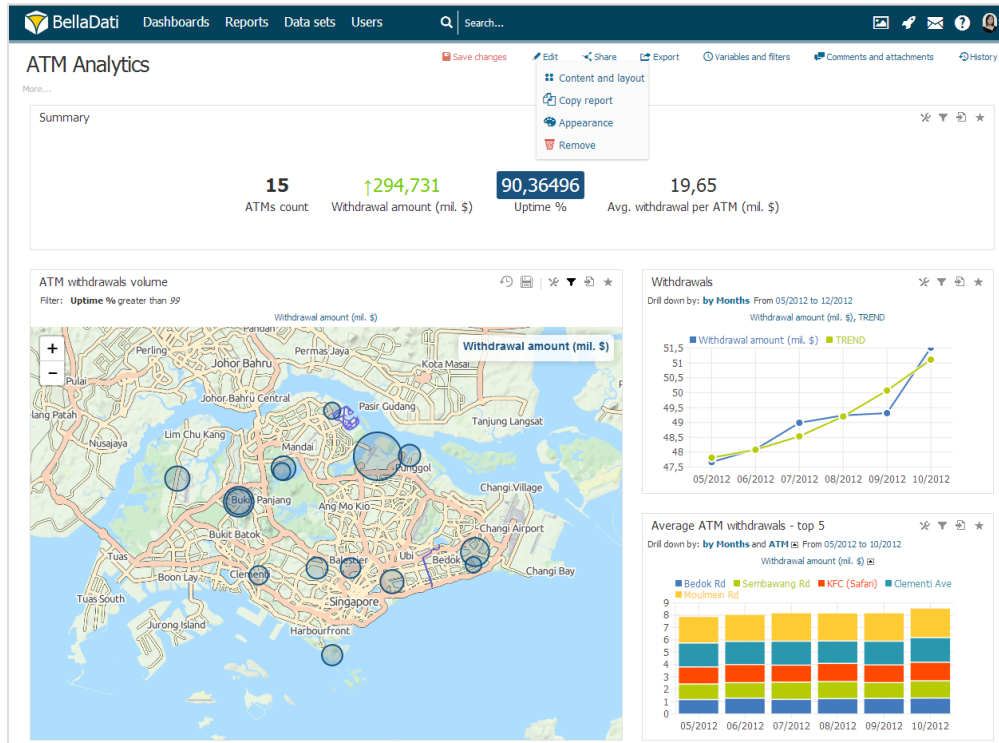
Reports in BellaDati serve for thorough data and trend analysis. Each report is composed of tables, charts, maps and KPI labels displaying data stored in data sets using various aggregations. Each report can also contain custom content, comments, attachments and can be easily shared with other BellaDati users, published to corporate intranet or public places. It is also possible to export each reports to PDF, Excel, PNG or Power Point. Additionally, regular exports sent via e-mail can also be scheduled.

The main purpose to work with reports is mainly for **analysts** who intend to drill-down accross large amounts of data in detail and then prepare selected figures for company managers.

Consider [dashboards](#) for the brief and fast continuous overview of trends by **managers**. Moreover dashboards allow to visually compare data from more reports and data set on a single page.



Only users with [report editor role](#) are allowed to manage the reports. If you don't have this role, please contact your BellaDati administrator.



In reports the following objects are defined and managed:


- [Report](#)
- [Report Layout](#)
- [View](#)
 - [Table](#)
 - [Chart](#)
 - [Geo map](#)
 - [KPI label](#)
 - [Custom content](#)
 - [Formulas](#)

Following actions can be performed within reports:

- [Setting Date Interval](#)
- [Displaying Indicators](#)
 - [Adding Indicators](#)
 - [Editing Indicators](#)
 - [Indicators Appearance](#)
 - [Conditional Formatting](#)
- [Using Filters](#)
 - [Filtering by Attribute](#)
 - [Filtering by Indicator](#)
 - [Modifying Indicators](#)
- [Exporting View](#)


- [Exporting to PDF](#)
- [Exporting to PNG](#)
- [Exporting to Excel](#)
- [Publishing View](#)
- [Sharing Report](#)
- [Copying Report](#)
- [Adding Comments and Attachments](#)

Creating Report

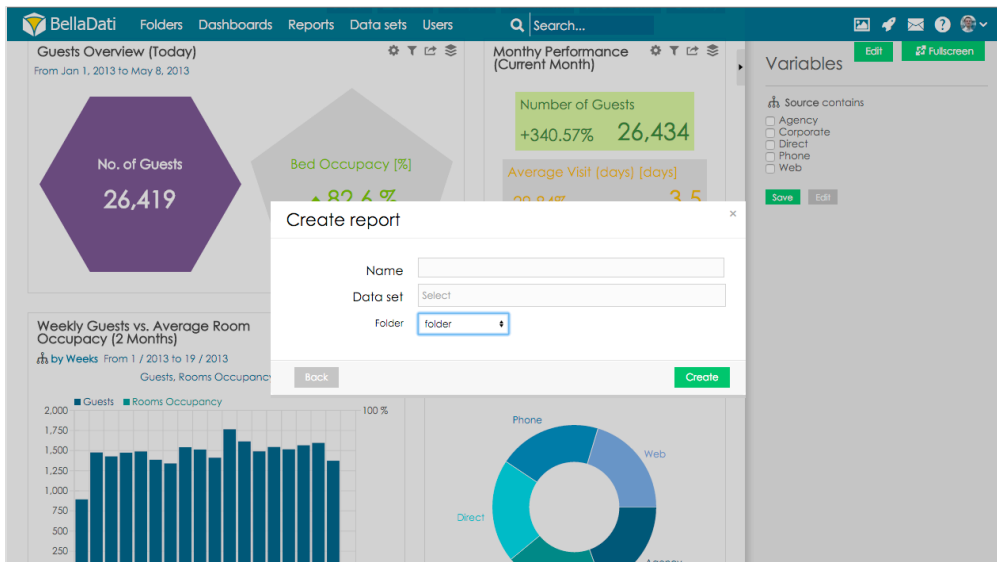
 Only users with report editor role are allowed to create and manage the reports.

Point to the **Reports** in main menu on the top of the screen and click "Create report" item which will appear.

1. Enter name of the new report.
2. Select the data set you are going to analyze data from.
3. Select Folder the report will be added to.

 New report can be created also directly from [data set summary](#) or [folders](#).

New report has no contents - continue by [Creating View](#).



Views


Each report can consist of the following elements (generally called views):

- [Creating Table](#)
- [Creating Chart](#)
- [Creating Geo Maps](#)
- [Creating KPI labels](#)
- [Adding Custom Content](#)

You can select up to three basic dimensions in each view:

- Date, Time
- Attributes (drill-down path)
- Indicators

See [Import Settings](#) or [Detailed Glossary](#) for more detailed description.

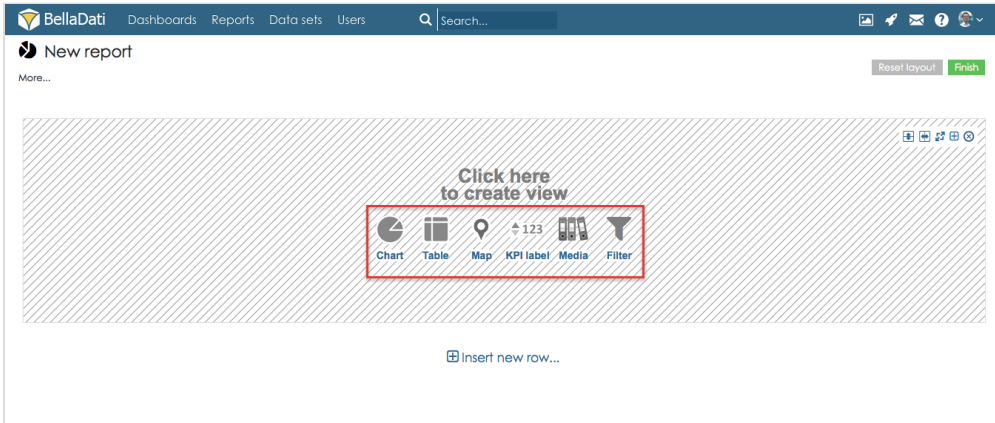
 There is a soft limit of about 8 views per one report. We do not recommend to exceed this limit to preserve good BellaDati performance running in your web browser.

Creating View



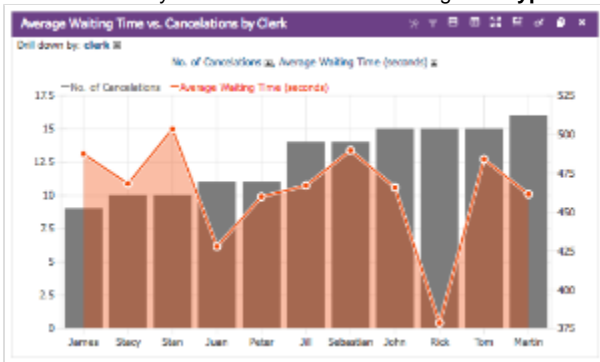
You need to be in [edit mode](#) in order to create new view. Click on "Edit" in top report menu to activate edit mode.

To add a new **View** hover over free place and click on desired **View type**. The *Add new view* dialog box will appear.



View types

BellaDati allows you to select from the following **view types**:



Chart

Chart view offers various data visualization types. You can specify:

- **Indicators** displayed in the chart.
- **Drill down path** used to categorize data in more detail.
- **Date interval** restricting time period of displayed data
- **Chart appearance**

To learn more about **Charts** continue by [Creating Chart](#).

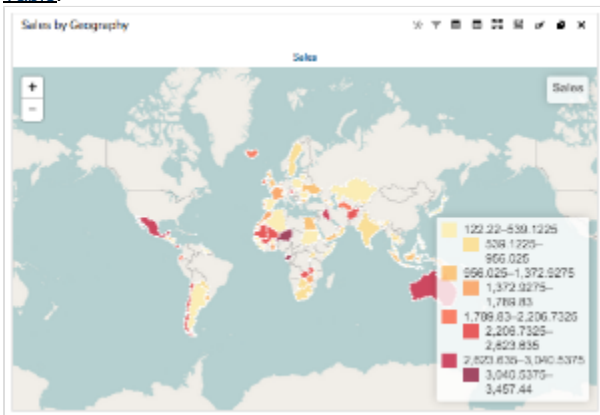
Name	Hours Played Daily				Hours Pl
	2013				
	Q1				
	February				
	25	26	27	28	
Duration (Seconds)	Duration (Seconds)	Duration (Seconds)	Duration (Seconds)	Duration (Seconds)	Duration
PETER	95,145	75,004	578,15	8,682	
JOHNS	66,46	57,131		66,682	
THOMAS	13,223				
GEORGE		1,773		58,752	
CLIVE	35,405			6,832	
JULIA				2,555	

Table

Table view allows displaying data in the crosstab grid. You can specify:

- **Indicators** displayed in the table.
- **Drill down paths** used to categorize data in more detail.
- **Date interval** restricting time period of displayed data
- **Table appearance**

To learn more about **Tables** continue by [Creating Table](#).



Geo map

Geo map view offers data visualization on the interactive map. You can specify:

- **Indicators** displayed in the map.
- **Drill down path** used to categorize data in more detail.
- **Time interval** restricting time period of displayed data

To learn more about **Geo maps** continue by [Creating Geo Maps](#).



KPI label

KPI label view allows clear monitoring of the important indicator value.

You can specify:

- **Indicator** display ed in the label.
- **Time interval** restricting time period of displayed data
- **Label appearance**

To learn more about **KPI labels** continue by [Creating KPI labels](#).

Filters

Date contains

7/7/2014 11/18/2014

Client contains

☐ Ratings
☐ Solutions

Current

United States

Filters

Filter view allows you to dynamically **modify content** of the report. You can also create custom components which allow report viewers to filter the report by just a few clicks.

Available custom components:

- Slider
- Multiple select
- List
- Check box
- Radio buttons
- Text

To learn more about filtering continue by [Creating Filters](#).

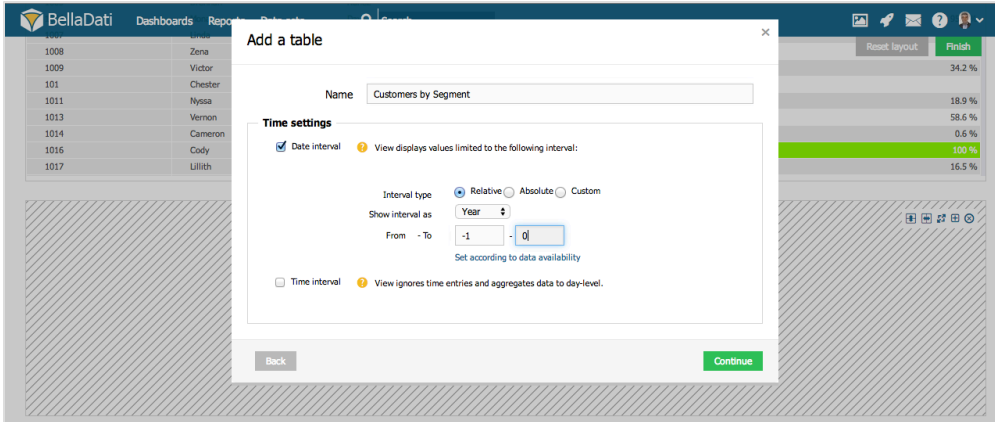
Creating Table



You need to be in [edit mode](#) in order to create new table. Click on "Edit" in top report menu to activate edit mode.

To add a new **Table**, hover over free place and click on **Table** view type. The *Add a table* dialog box will appear.

1. Enter name of the new table.
2. Check **Date interval** if you need to restrict time period of displayed data - continue by [Setting Date Interval](#).



Confirm new **Table** view by clicking on a green **Add** button. BellaDati will guide you through additional setup.

- **Indicators:** select and edit displayed indicators in the table - continue by [Displaying Indicators](#).



You can create also table without any indicators. This is useful for "static" items lists reports (especially in conjunction with more drill-downs and hiding drill-down "+" signs).

Table management

You can perform additional operations in the upper right corner of the inserted **Table** view:

- Table settings
- Indicators
- Table appearance
- Filter setting - continue by [Using Filters](#)
- Export view - continue by [Exporting View](#)
- Add to dashboard
- Move table
- Duplicate table
- Delete table



Hover over **Indicators** in a toolbox list to quickly add or remove indicators.

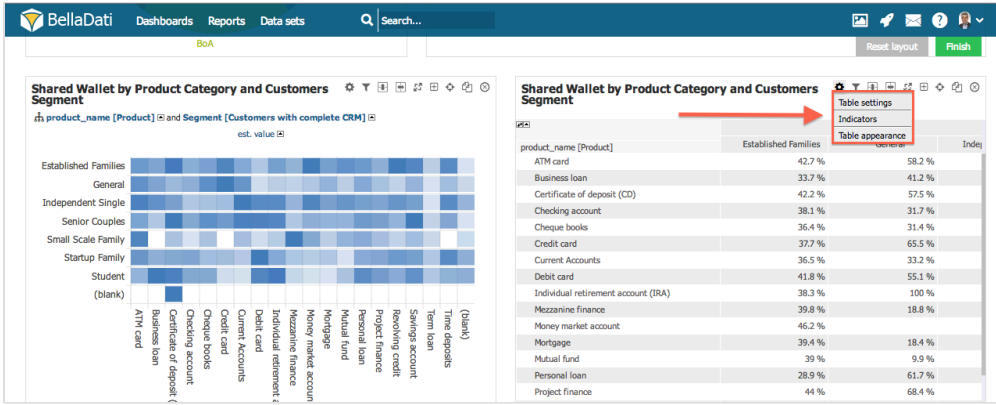
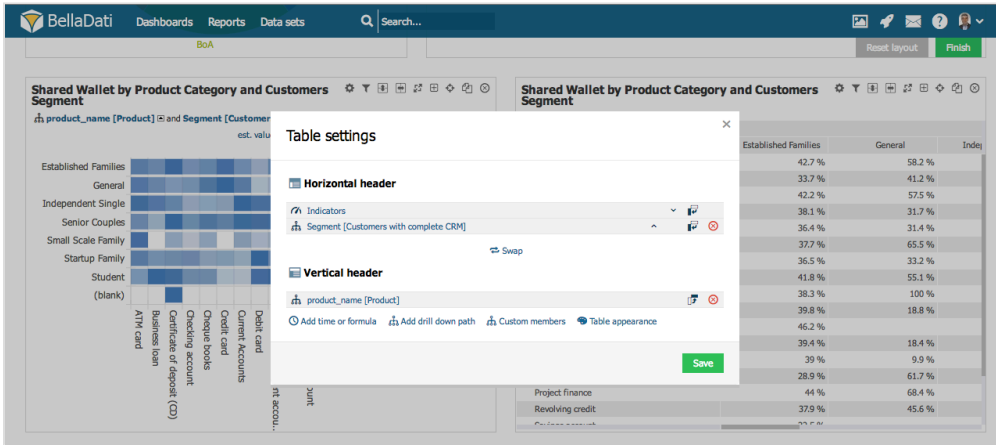


Table settings

Click on **toolbox** icon or select **Table settings** from the toolbox list to enter *Table settings* dialog.

Table settings dialog shows **Time**, **Indicators** and **Drill down paths** currently positioned at X and Y axes. BellaDati allows you to:

- **Swap** X and Y axes
- **Switch** between horizontal and vertical position.
- **Change** order within axis.
- **Remove** drill down path.
- **Enter** indicators or drill down path dialogs.
- [Editing Table Axes Content](#)
- **Add** custom members
- **Edit** table appearance.



Drill-down managing

Hover your mouse over the drill-down header to display drill-down management box. This box allows you to:

- add new drill down to the table ("+" sign)
- remove drill down from the table ("X" sign)
- change position of drill-downs in the table ("<" and ">" signs)

The screenshot shows the BellaDati interface with a table. A red arrow points to the drill-down management box, which is a small box that appears when you hover over the drill-down header. The box contains a list of indicators and drill down paths, and it also has a 'Swap' button and a 'Save' button.

Market	Major Market	Small Market	Sales	Profit
North	Major Market	Small Market	152,579	89,337
East	Major Market	Small Market	112,466	34,515
South	Major Market	Small Market	138,240	48,848
West	Major Market	Small Market	40,316	10,369
	Major Market	Small Market	37,410	15,744
	Major Market	Small Market	66,516	16,712
	Major Market	Small Market	96,892	31,785
	Major Market	Small Market	175,372	42,211


Custom member

Custom member allows you adding your own nodes into drill down paths. You can add Custom member from *Table settings* dialog after selecting **Add custom member**.

Custom member dialog allows you to:

- Select **Level** for a custom member.
- Specify **Name** of the custom member.

Every new node requires additional definition. *Custom member definition* dialog allows you to determine attribute values from particular levels which will be aggregated into custom member.

 You can delete created nodes in *Custom member* dialog.

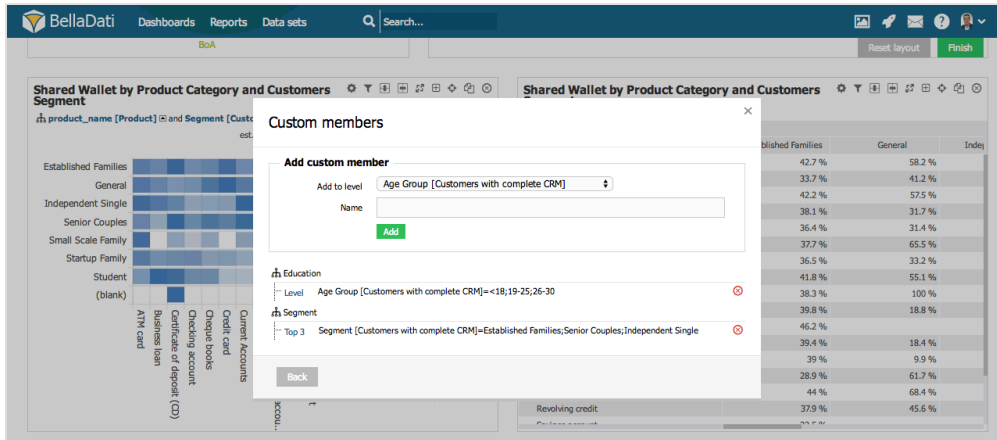



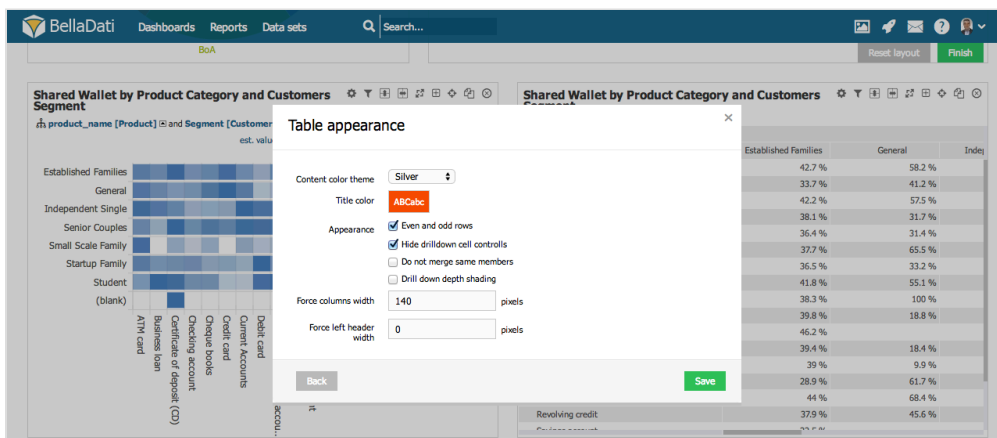
Table Appearance

You can access **Table Appearance** from the *Table Settings* dialog or the from toolbox drop down list.

Table Appearance dialog allows:

- Applying predefined **Content color themes**.
- Setting **Title Color**.
- Checking **Even and odd rows** differentiation.
- **Hide drilldown cell controls**: Drill-down "+" signs will be disabled for all drill-downs in the table (useful for "static" item lists etc.).
- Selecting **Drill down depth shading**.
- Force **column width**.
- Force **left header width**.
- Set **sort indicator** and its **aggregation**.

 Hover over **Table Appearance** in toolbox list to quickly apply **Title Color**.



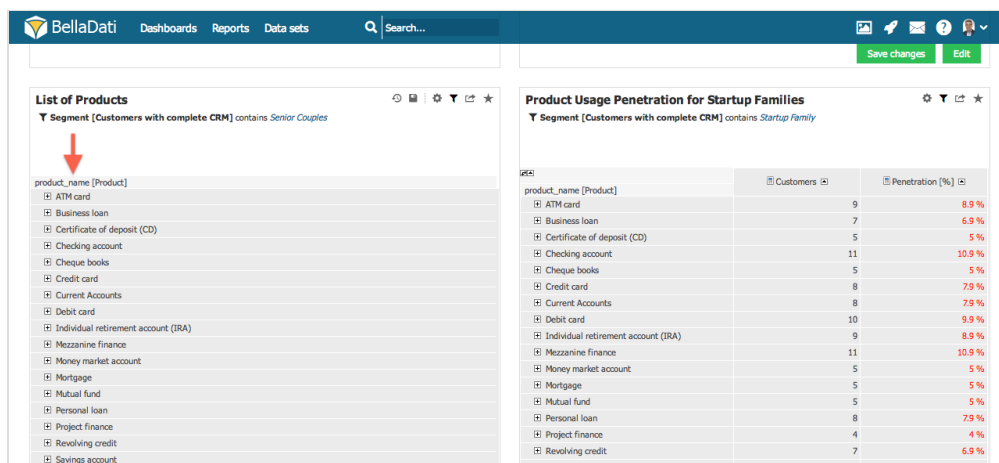
Displaying Source Data

See [how it works](#).

Static Lists

BellaDati allows you to list attribute members without any indicators in form of a **static lists**. In order to create static lists:

1. Create new **Table**
2. Select arbitrary **Indicator**.
3. Choose desired **Attributes**.
4. Remove **Indicator** from the table.
5. Optionally hide drill-down controls in **Table Appearance**.
6. Optionally prevent BellaDati from merging same members in **Table Appearance**.



The screenshot shows the BellaDati interface with two static lists. The left list is titled 'List of Products' and the right is titled 'Product Usage Penetration for Startup Families'. Both lists show a table of products with their respective attributes.

List of Products

Segment: [Customers with complete CRM] contains Senior Couples

product_name [Product]
ATM card
Business loan
Certificate of deposit (CD)
Checking account
Cheque books
Credit card
Current Accounts
Debit card
Individual retirement account (IRA)
Mezzanine finance
Money market account
Mortgage
Mutual fund
Personal loan
Project finance
Revolving credit
Savings account

Product Usage Penetration for Startup Families

Segment: [Customers with complete CRM] contains Startup Family

product_name [Product]	Customers	Penetration (%)
ATM card	9	8.9 %
Business loan	7	6.9 %
Certificate of deposit (CD)	5	5 %
Checking account	11	10.9 %
Cheque books	5	5 %
Credit card	8	7.9 %
Current Accounts	8	7.9 %
Debit card	10	9.9 %
Individual retirement account (IRA)	9	8.9 %
Mezzanine finance	11	10.9 %
Money market account	5	5 %
Mortgage	5	5 %
Mutual fund	5	5 %
Personal loan	8	7.9 %
Project finance	4	4 %
Revolving credit	7	6.9 %
Savings account	11	10.9 %

Setup export layout

To see how setup your own export layout (custom row height, column width, table header, data formatting or merged regions), proceed to [separate page](#).

Adding Date Intervals



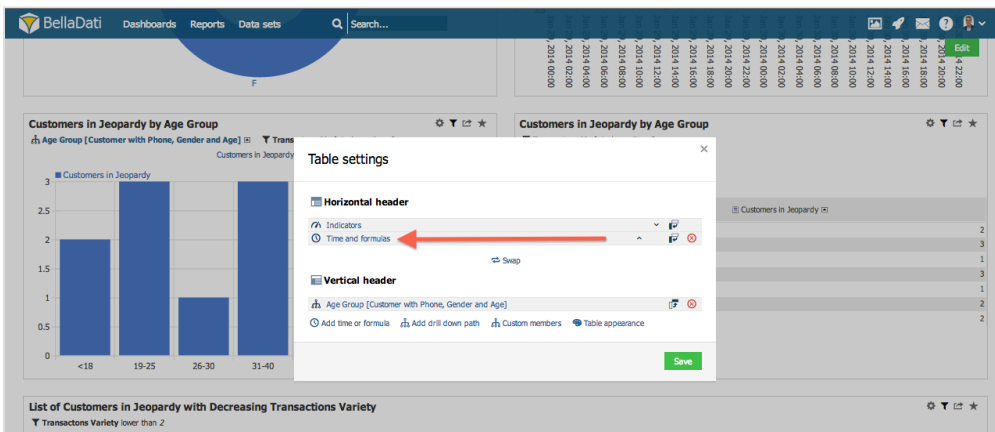
Please, make sure to get familiar with [Setting Date Interval](#) before proceeding with this section. Note, that adding multiple intervals is allowed only for **Table** views.

When creating Table view, you can define multiple **Time intervals** or write special **Table formulas**.

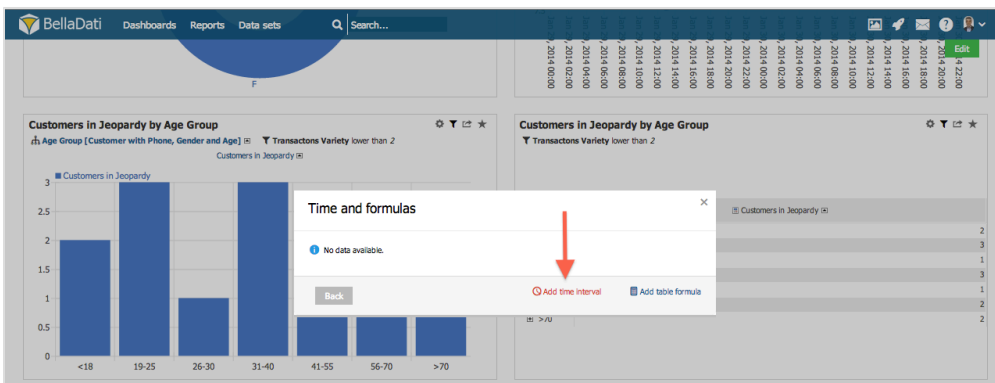
Adding time interval

To add new time interval go to **Table settings** and:

- Click **Add time definition**, if you have not applied any interval yet.
- Click **Time and formulas**, if you have already applied time interval(s).



You will see list of current time intervals. To add new time definition, click **Add time interval**. You will be prompted with popup window similar to one in [Setting Date Interval](#).

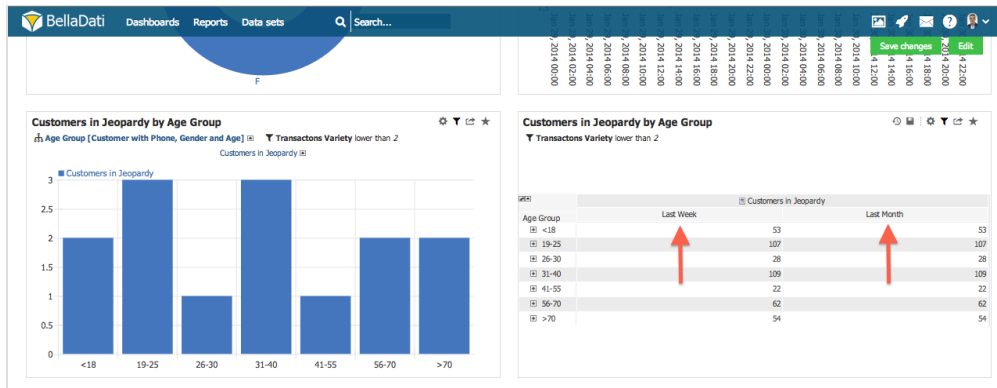


After settings confirmation, table will be extended with defined intervals.



Use time intervals to display data with various date/time granularities. For example: Display data broke-down by months and total year.

You can see table with **Current Month** and **Last Year in Months** definition below.



Date Interval Appearance

Click on **Appearance Settings** button next to **Date Interval** definition. It allows you to set up:

- Color
- Bold values
- Avoid conditional formatting
- Emphasise only

Working with Date Intervals

This user case will introduce you how to work with data intervals

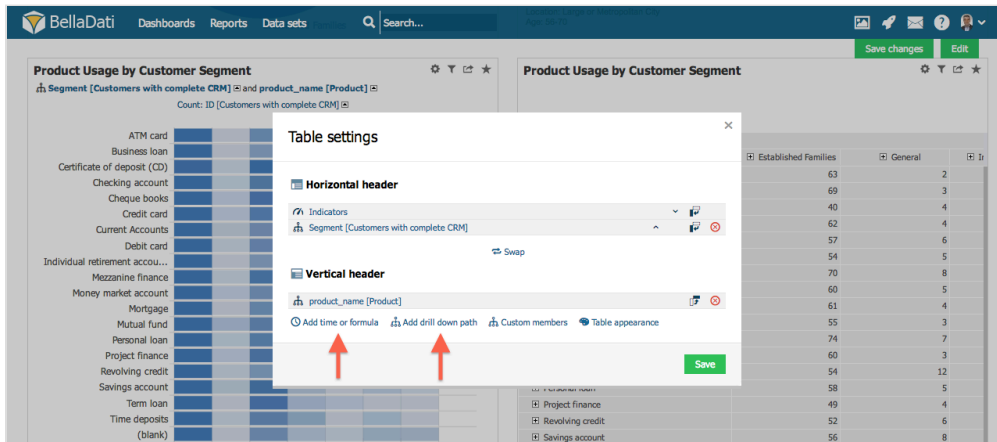
1. **Adding time interval with absolute date values**
2. **Adding time interval with relative date values**
3. **Adding table formula with absolute date values**
4. **Adding table formula with date variables**

Editing Table Axes Content

BellaDati allows you to add to each axis multiple:

- **Date Aggregation**
- **Drill-down path**

Click on **Add time or formula** or **Add drill down path** links to place them onto the axis.



Adding Date Aggregation to the Axis

To add date aggregation:

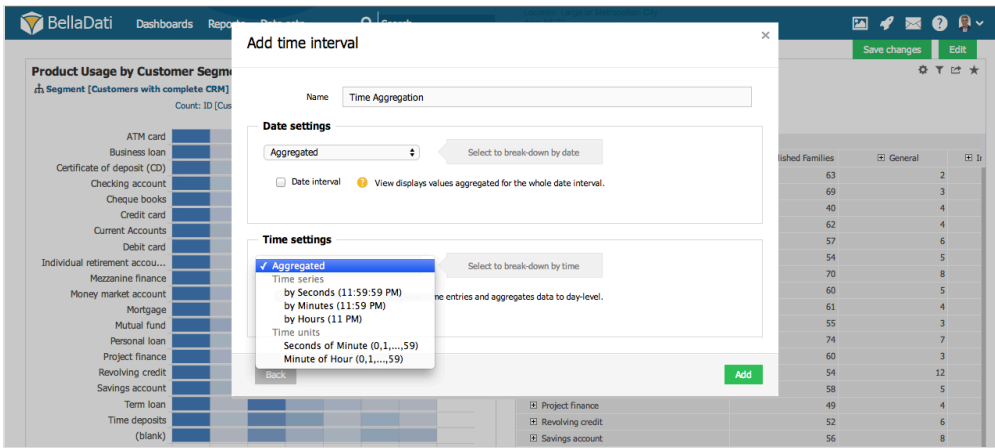
1. click on **Add time or formula** link
2. click on newly added **Times and formulas**
3. select **Add time interval**

You can place following date aggregations on the axis:

- **Time Series**
 - **by Days** - displays axis with all days from the data set or days from the date filter (**1/1/2013 - 1/3/2014**)
 - **by Weeks** - displays axis with all weeks from the data set or weeks from the date filter (**1/2013 - 48/2014**)
 - **by Months** - displays axis with all months from the data set or months from the date filter (**1/2013 - 8/2014**)
 - **by Quarters** - displays axis with all quarters from the data set or quarters from the date filter (**I/2013 - III/2014**)
 - **by Years** - displays axis with all years from the data set or years from the date filter (**2013 - 2014**)
- **Date Units**
 - **Day of Week** - displays axis with days of week (**Su - Sa**)
 - **Day of Month** - displays axis with days of month (**1 - 31**)
 - **Day of Year** - displays axis with days of year (**1 - 366**)
 - **Week of Year** - displays axis with weeks of year (**1 - 53**)
 - **Month of Year** - displays axis with months of year (**1 - 12**)
 - **Quarter of Year** - displays axis with quarters of year (**I - IV**)
 - **Year by Weeks** - displays axis with years taking in consideration to weeks (**2013 - 2014**)

Additionally, you can place following time aggregations (if available):

- **Time Series**
 - **by Seconds** - displays axis with all seconds from the data set or seconds from the time filter
 - **by Minutes** - displays axis with all minutes from the data set or minutes from the time filter
 - **by Hours** - displays axis with all hours from the data set or hours from the time filter
- **Time Units**
 - **Seconds of Minute** - displays axis with seconds of minute (**1 - 59**)
 - **Minutes of Hour** - displays axis with minutes of hour (**1 - 59**)

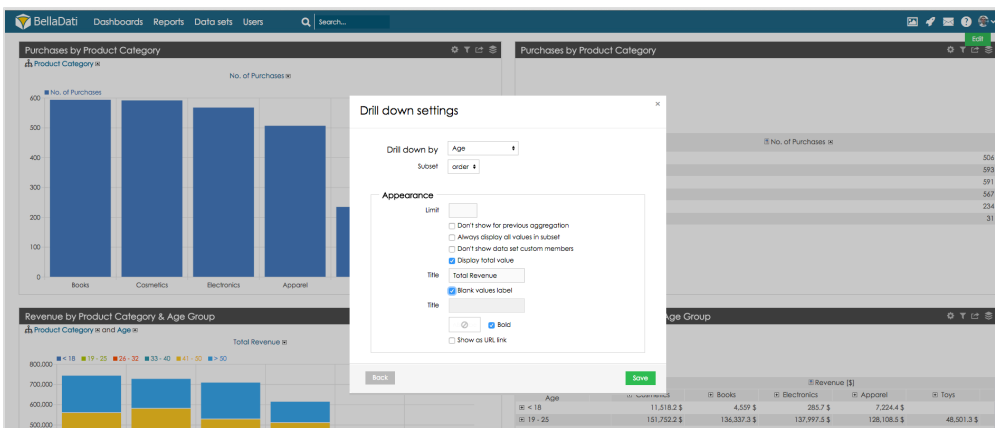


Drill Down Path

You can add Drill down path from *Table settings* dialog after selecting **Add drill down path**.

From *Drill down path* dialog you can perform following operations:

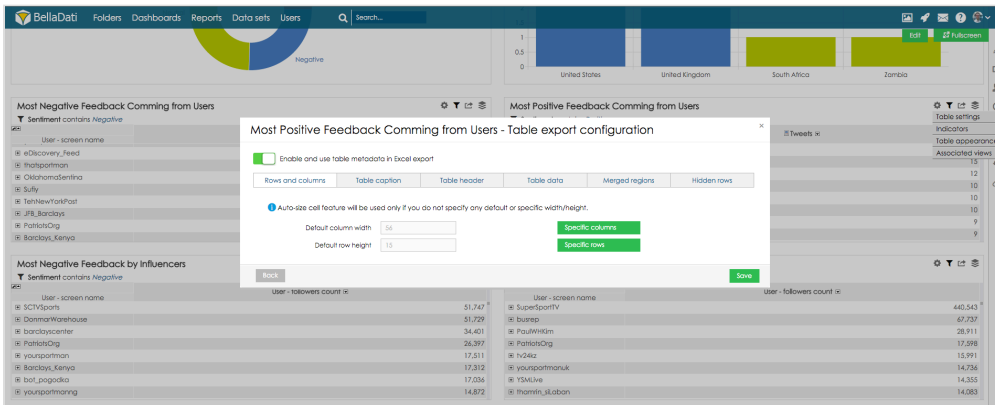
- Add new **Drill down path** to the view.
- Mask members with custom **URLs**
- Setup **Limit** for members in the drill down path. Displayed members depend on current sorting setup.
- Select if all values in subset will be displayed.
- Select if date set custom members shouldn't be displayed.
- Display **Total** value. New consolidated element will be added to the drill down path. Input title which will be displayed for total value.
- Input **label** which will be displayed instead of blank fields.
- **Format** total value's font color, style and background.



Setting table export configuration

Click on **toolbox** icon in table view and click button **Table export configuration**.

In newly displayed dialogue window select option **Enable and use table metadata in Excel export** - when this option is selected, custom table export settings will be used.



This dialogue window allows you to:

- set **column width** and **row height**
- set formatting of **table caption**
- set formatting of **table header**
- set formatting of **table data**
- set **merged regions**

Setting column width and row height

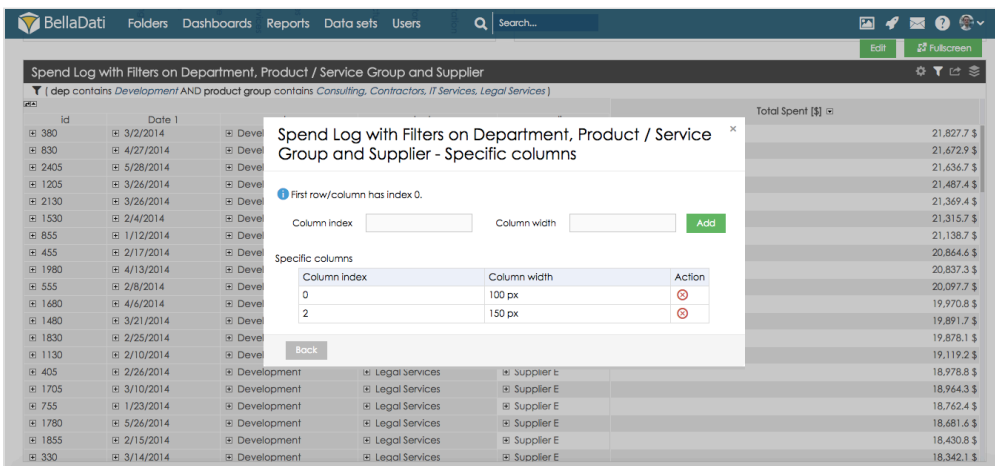
Use fields **Default column width** and **Default row height** to set custom column width and row height in pixels.

Click on button **Specific columns** to set different column width for different columns. These columns are specified in the field **Column index**, which defines position of the column. First column has index 0.

Example:

First column should have column width set to 100px and third column will have column width set to 150px.

1. Set column index to 0, column width to 100px and click button Add.
2. Set column index to 2, column width to 150px and click button Add.



Click on button **Specific rows** to set different row height for different rows. These rows are specified in the field **Row index**, which defines position of the row. First row has index 0.

Example:

First row should have row height set to 20px and fifth column will have column width set to 15px.

1. Set row index to 0, row height to 20px and click button Add.
2. Set row index to 4, row height to 15px and click button Add.

The screenshot shows the BellaDati interface with a table titled "Spend Log with Filters on Department, Product / Service Group and Supplier". A modal window is open, titled "Spend Log with Filters on Department, Product / Service Group and Supplier - Specific rows". The modal contains a message "First row/column has index 0." and two input fields: "Row index" set to 4 and "Row height" set to 15. There is an "Add" button. Below these fields is a table with columns "Row index", "Row height", and "Action". The table has two rows: one with "0" and "20 px" and another with "4" and "15 px". There are "Back" and "Add" buttons at the bottom of the modal.

Setting table caption formatting

Click button **Table header** to set custom formatting of table header. This section allows you to:

- set **Background color**
- set **Font color**
- set **Font name**
- set **Font size**

The screenshot shows the BellaDati interface with a table titled "Most Positive Feedback Coming from Users - Table export configuration". A modal window is open, titled "Most Positive Feedback Coming from Users - Table export configuration". The modal contains a checkbox "Enable and use table metadata in Excel export" which is checked. Below the checkbox are four tabs: "Rows and columns", "Table caption", "Table header", and "Table data". The "Table header" tab is selected. The modal contains a "Background color" input field, a "Font color" input field, a "Font name" input field, and a "Font size" input field. There are "Back" and "Save" buttons at the bottom of the modal.

Setting table header formatting

Click button **Table header** to set custom formatting of table header. This section allows you to:

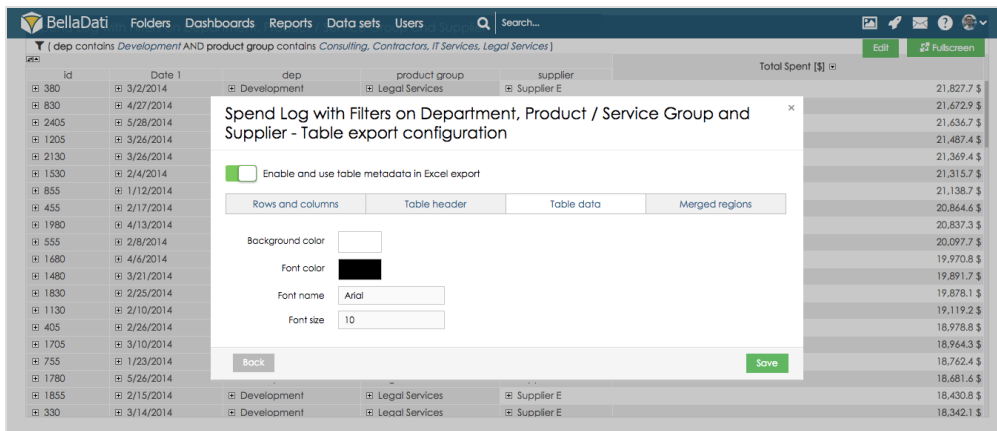
- set **Background color**
- set **Font color**
- set **Font name**
- set **Font size**

The screenshot shows the BellaDati interface with a table titled "Spend Log with Filters on Department, Product / Service Group and Supplier - Table export configuration". A modal window is open, titled "Spend Log with Filters on Department, Product / Service Group and Supplier - Table export configuration". The modal contains a checkbox "Enable and use table metadata in Excel export" which is checked. Below the checkbox are four tabs: "Rows and columns", "Table caption", "Table header", and "Table data". The "Table header" tab is selected. The modal contains a "Background color" input field, a "Font color" input field, a "Font name" input field, and a "Font size" input field. There are "Back" and "Save" buttons at the bottom of the modal.

Setting table data formatting

Click button **Table data** to set custom formatting of table data. This section allows you to:

- set **Background color**
- set **Font color**
- set **Font name**
- set **Font size**



Setting merged regions

This section allows you to merge column and rows into one field and set custom content for this region. Click button **Add merged region** to open settings of merged region.

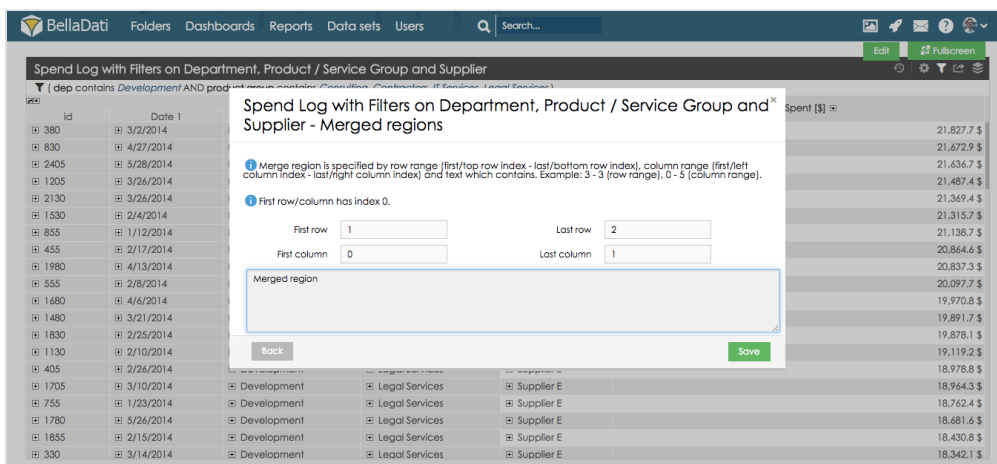
Merged region is defined by row and column ranges. First row and first column always have index 0.

- Use fields **First row** and **Last row** to define row range.
- Use fields **First column** and **Last column** to define column range.
- At the bottom field can be defined custom content of the merged region.

Example:

For row number 2 and 3 and columns number 1 and 2 should be displayed with custom text "Merged region":

1. Set value in the field first row to 1
2. Set value in the field last row to 2
3. Set value in the field first column 0
4. Set value in the field last row to 1
5. Set custom text to Merged region
6. Click Save to save the settings



Creating Chart



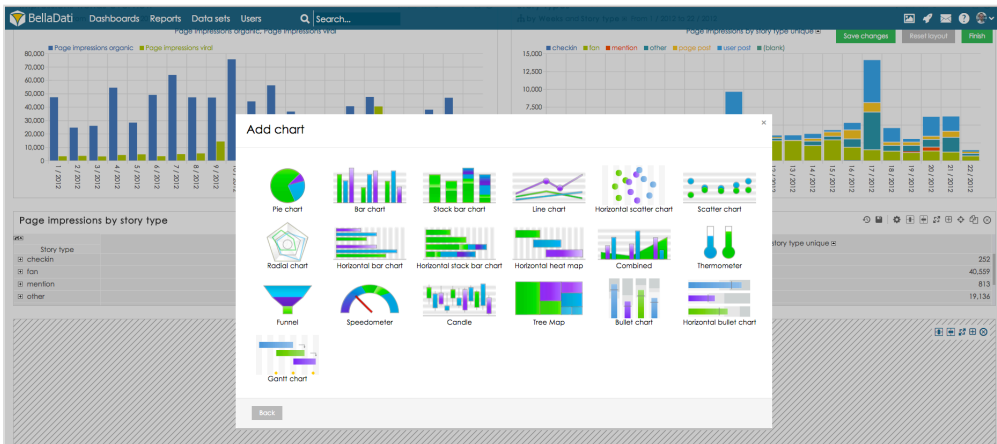
You need to be in [edit mode](#) in order to create new chart. Click on "Edit" in top report menu to activate edit mode.

Basic tutorial

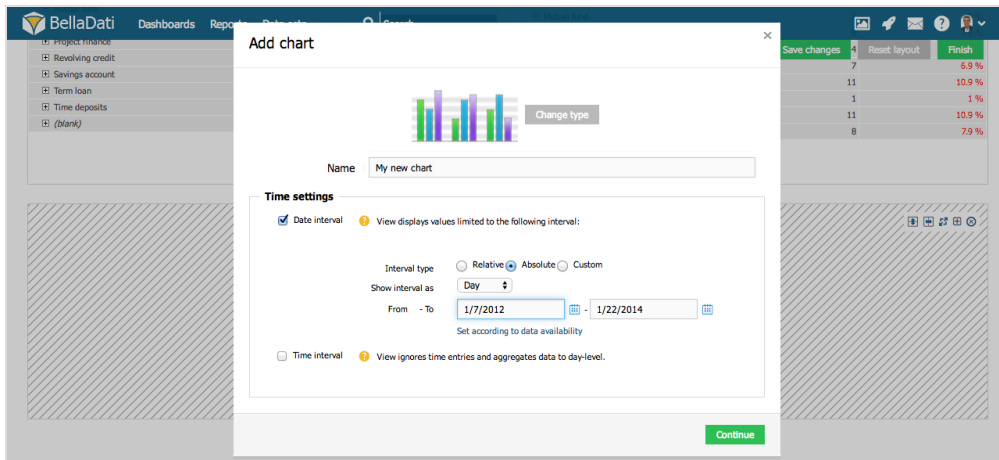
Chart is a graphic representation of various information and displays values of the indicators depending on user's criteria.

To add a new **Chart** hover over free place and click on **Chart** view type. The *Add chart* dialog box will appear.

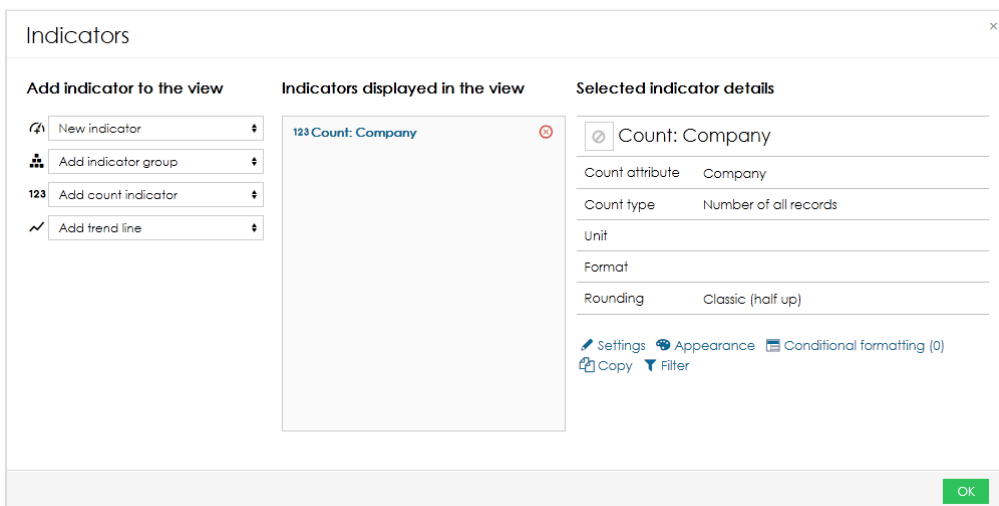
1. Select type of chart by clicking on one of thumbnails. BellaDati supports various chart types:
 - Pie chart
 - Bar chart
 - Stack bar chart
 - Line chart
 - Horizontal scatter chart
 - Scatter chart
 - Radial chart
 - Horizontal bar chart
 - Horizontal heat map
 - Candle chart
 - Thermometer
 - Funnel
 - Speedometer
 - Bullet chart
 - Horizontal bullet chart
 - Gantt chart
 - Combined (each indicator can be displayed differently): bar chart, stack bar chart, line chart, scatter chart



2. Enter name of the new chart.



3. Check **Date interval** if you need to restrict time period of displayed data - continue by [Setting Date Interval](#).
4. Add indicators into view from **Indicators** dialog box. *Indicators* dialog box is separated into three columns:
 - List of **Available indicators**
 - List of **Indicators displayed in the view**
 - Additional **Indicator details**



For more information about **Indicators** continue by [Displaying Indicators](#).

5. Add **drill-down** by selecting one or more **attributes**. Drill-down is used for changing granularity according to the chosen drill-down path. The settings for drill-down is different for various charts:
 - a. For most charts, you can set drill-down for both x-axis and y-axis. This applies for:
 - Bar chart
 - Stack bar chart
 - Line chart
 - Horizontal scatter chart
 - Scatter chart
 - Radial chart
 - Horizontal bar chart
 - Horizontal heat map
 - Combined
 - b. Some chart allows only one drill-down:
 - Pie chart - you can also set

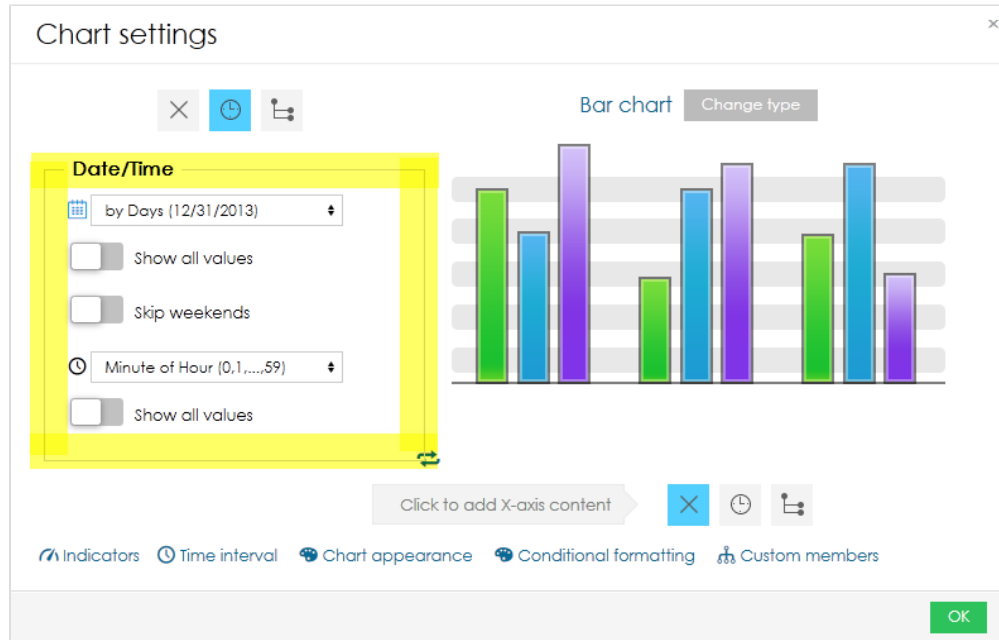
secondary drill-down - use option **Show details for**.

- Candle chart
- Thermometer
- Funnel
- Speedometer
- Bullet chart
- Horizontal bullet chart
- Gantt chart

You can use two basic types of drill-down:

- a. Drill-down by regular attribute - use attributes such as City, Country, Department or Customer for changing granularity.

Drill-down by date or time. You can place following date aggregations on the axis:



For more information continue by [Editing Chart Axes Content](#).

6. Click **OK** to finish the creation of chart. Alternatively you can edit advanced settings and appearance of chart.

Chart management

You can perform additional operations in the upper right corner of the inserted **Chart** view:

- Chart settings - see below
- Associate - continue by [Associated views](#)
- Drill down paths - continue by [Defining Drill-down Path](#)
- Indicators - continue by [Displaying Indicators](#)
- Chart appearance - continue by [Managing chart appearance](#)
- Filter setting - continue by [Using Filters](#)
- Export view - continue by [Exporting View](#)
- Add to dashboard - continue by [Creating Dashlet](#)
- Move chart - use drag&drop for changing the layout of report
- Duplicate chart - click on the Copy icon
- Delete chart - click on the "x" icon



Hover over **Indicators** in the toolbox list to quickly add or remove indicators. Hover over **Drill down path** in the toolbox list to quickly add or remove attributes.

Chart Settings

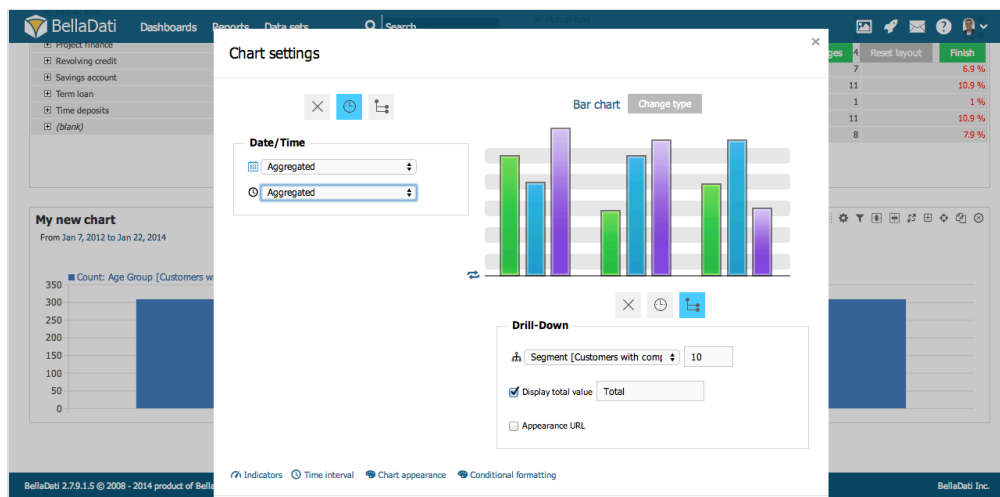
Click on **toolbox** icon or select **Chart settings** from the toolbox list to enter *Chart settings* dialog.

Each chart usually contains at least one indicator. *Chart settings* dialog allows you to:

- Edit **Indicators**
- **Change type** of the chart
- Add **Time axis** - continue by [Setting Date Interval](#)
- [Editing Chart Axes Content](#)
- Change **Chart appearance**
- Edit **Conditional formatting**
- Force indicators to **Display in the single chart**



When more **Indicators** are added to the chart, they are displayed in separate charts within the view by default. **Display in the single chart** option forces them to be grouped into single chart.



Displaying Source Data

See [how it works](#).

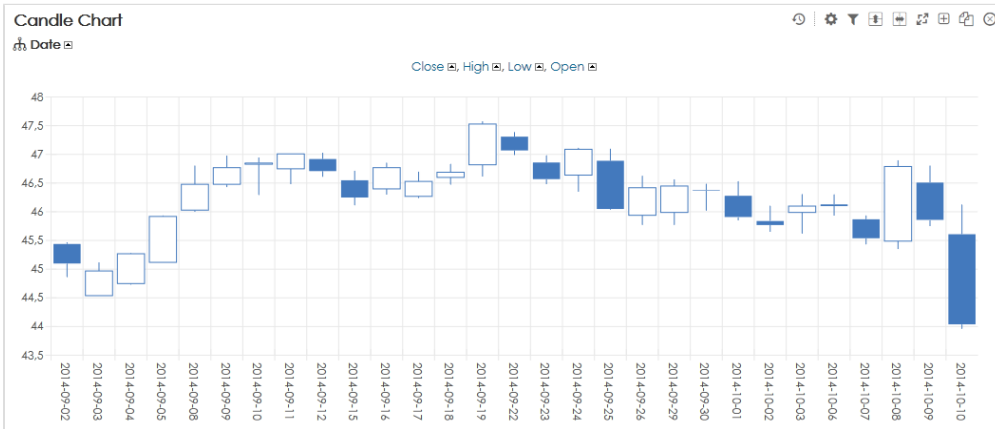
Creating Candlestick Chart



You need to be in [edit mode](#) in order to create new chart. Click on "Edit" in top report menu to activate edit mode.

A **Candlestick Chart** is a financial chart used to describe price movements during time. There are four basic values that describe each candle: **the open, the close, the high and the low**. Therefore in BellaDati these four indicators and a **Date attribute** have to be present in data set.

1. To add a new **Candlestick Chart** hover over free place and click on **Chart** view type and select Candlestick chart.
2. Add name of the chart and specify **Time interval** if needed - continue by [Setting Date Interval](#).
3. Add four indicators to chart and click OK.
4. Assign these indicators to correct values: **the open, the close, the high and the low**.
5. Add **drill-down** by **Date**
6. Finish by clicking on OK.



Creating Gantt Chart



You need to be in [edit mode](#) in order to create new chart. Click on "Edit" in top report menu to activate edit mode.

Gantt chart illustrates the start and finish dates of elements (tasks) of a project. In BellaDati there are two ways how display tasks in Gantt Chart:

- Task is specified by **Start Date** and **Length** of the task.
- Task is specified by **Start Date** and **End Date**.



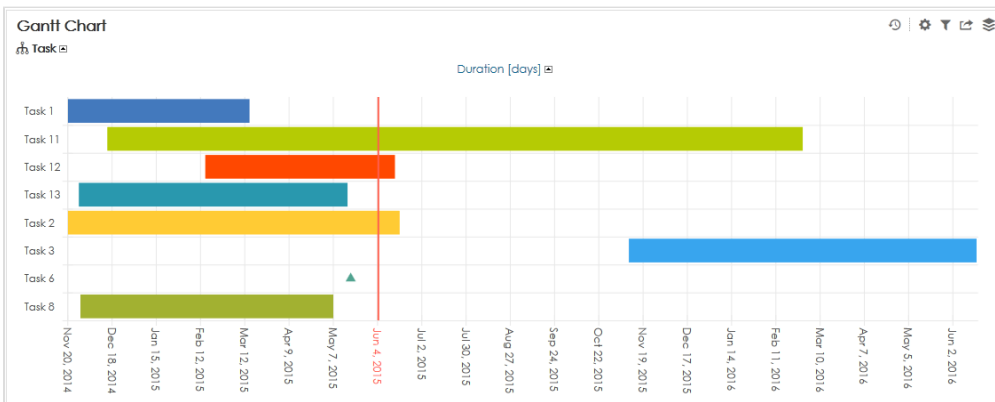
Task without duration (0 days) is displayed as milestone (triangle).

Task specified by Start Date and Length of the task

- To add a new **Gantt Chart** hover over free place and click on **Chart** view type and select Gantt chart.
- Add name of the chart and specify **Time interval** if needed - continue by [Setting Date Interval](#).
- Add an indicator which specifies length of the task.
- Select Start Date attribute.
- Finish by clicking on OK.

Task specified by Start Date and End Date

- To add a new **Gantt Chart** hover over free place and click on **Chart** view type and select Gantt chart.
- Add name of the chart and specify **Time interval** if needed - continue by [Setting Date Interval](#).
- Select Start Date attribute.
- Select End Date attribute.
- Enable "Use end date instead of a duration indicator definition".
- Finish by clicking on OK.



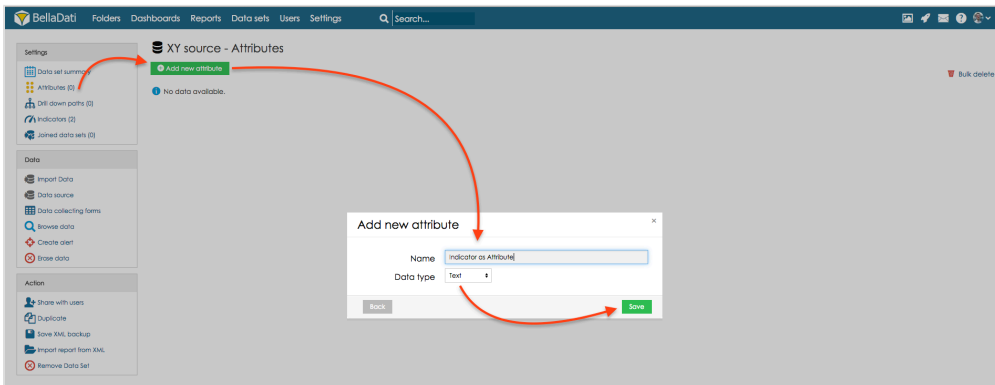
Creating XY chart

i XY charts displays two groups of indicators as one series of XY coordinates. XY charts show the relatedness of number values. As XY chart should be used scatter chart or line chart.

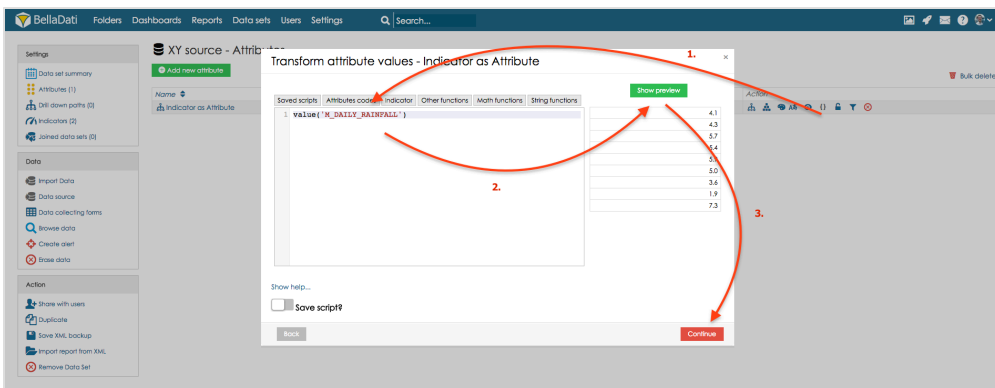
In XY scatter chart, there are displayed indicators on both axis of the chart. In BellaDati, there can be displayed indicator value on one axis only, so we have to transform indicator value into new attribute.

1. Transform indicator into attribute

Open source data set. Create new attribute of type Text.



Transform indicator values into this attribute. For this option, click on transformation script icon of newly created attribute and use transformation script, that will return value with indicator values:

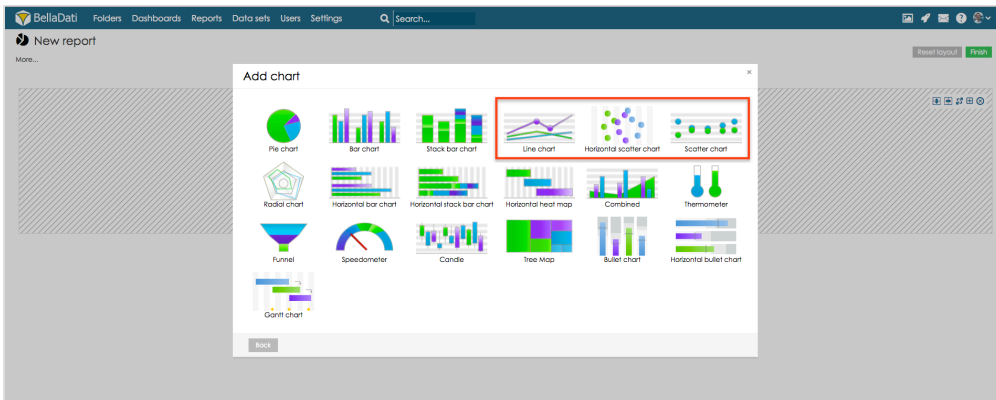


For more information about [transformation scripting](#), go to [Developers](#) section.

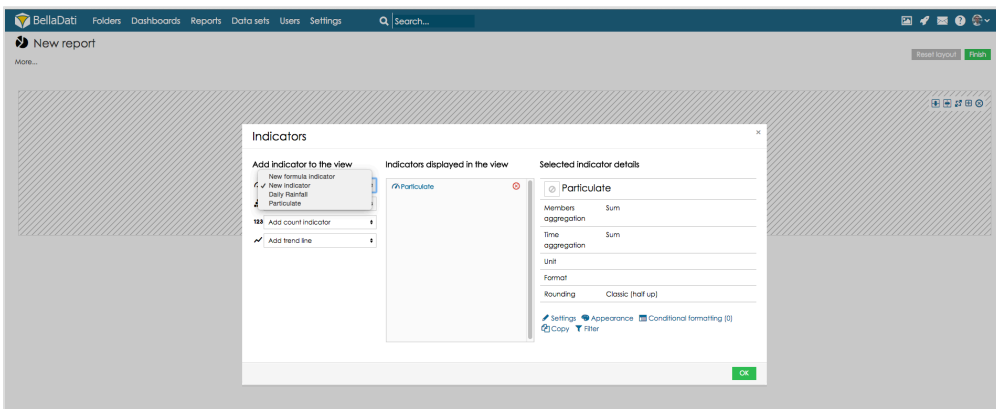
2. Create chart

Now, go back to the report, which is based on this data set or [create new report](#).

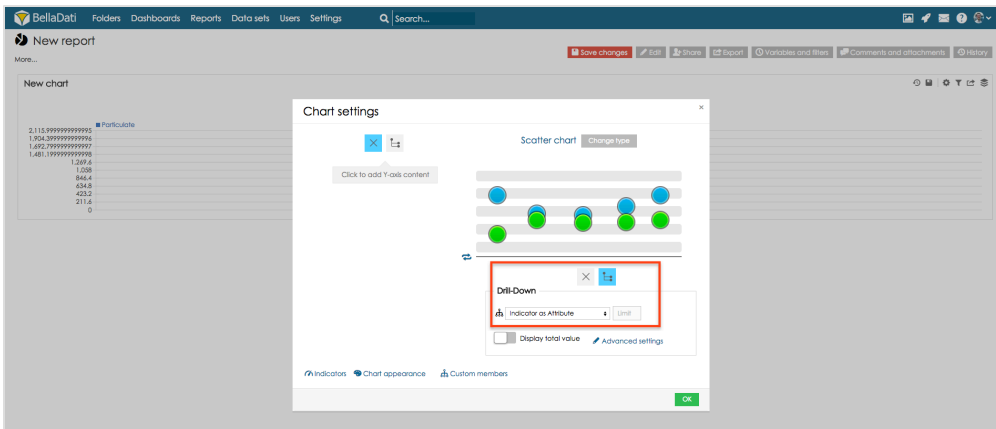
Now let's [create new XY chart](#). Create new view, select scatter chart, horizontal scatter chart or line chart.



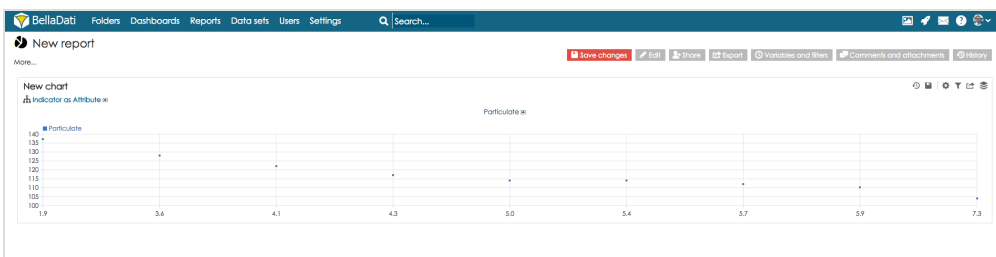
Select indicator which should be displayed.



Select drill-down that contains indicator values.

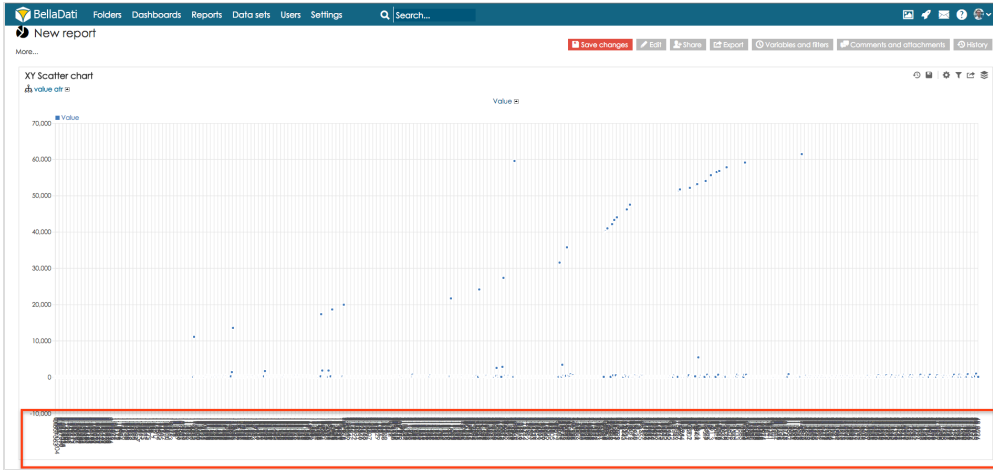


XY chart is created.

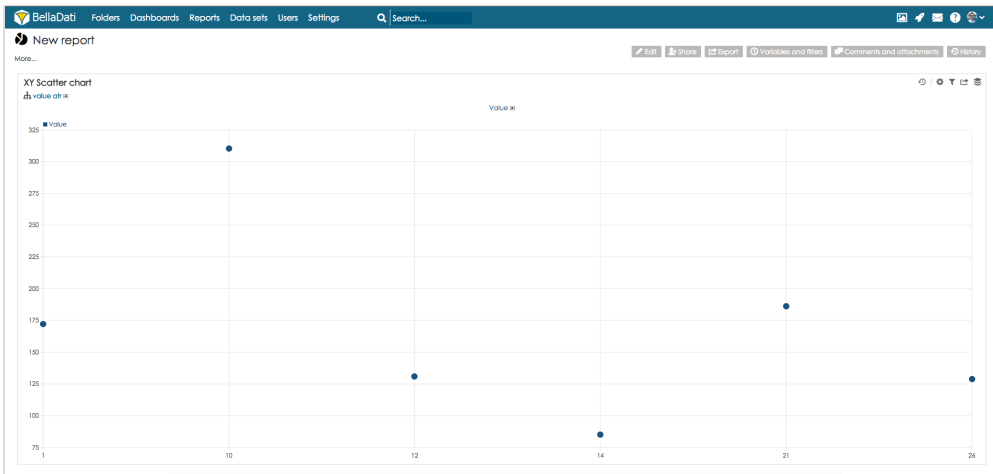


3. Additional settings


Problem: source data contain a lot of different values. This causes that values are not displayed correctly on axis with attribute.



Solution: in this case, we need to round attribute values. This should be done via transformation scripts. Proceed to use case with change of decimal values described.



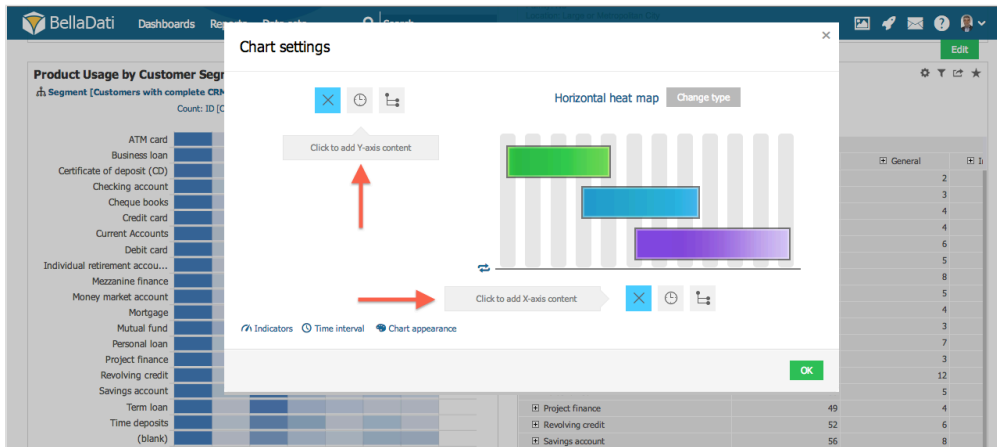
Editing Chart Axes Content

 Different chart types may have various number of axes.

BellaDati allows you to set each axis to:

- **Date Aggregation**
- **Drill-down path**
- **Empty content**

Click on **axis type** button to set the content.



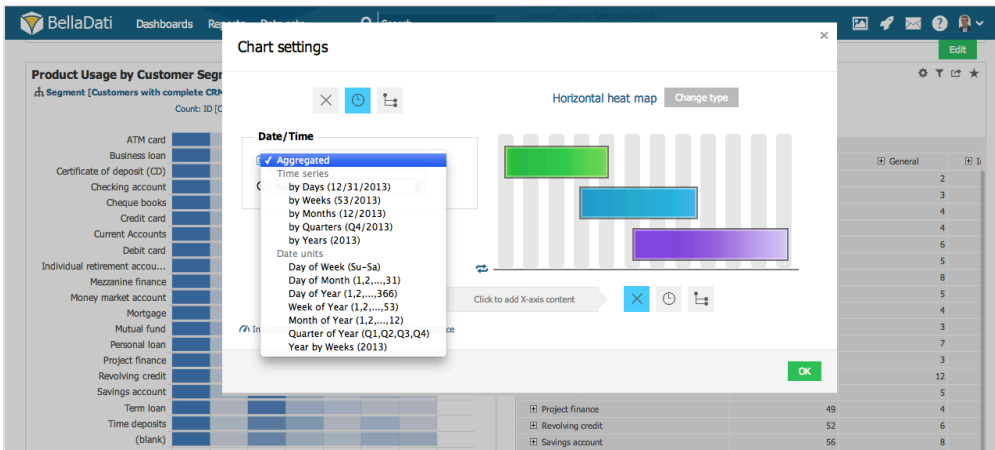
Setting Axis to Date Aggregation

You can place following date aggregations on the axis:

- **Time Series**
 - **by Days** - displays axis with all days from the data set or days from the date filter (**1/1/2013 - 1/3/2014**)
 - **by Weeks** - displays axis with all weeks from the data set or weeks from the date filter (**1/2013 - 48/2014**)
 - **by Months** - displays axis with all months from the data set or months from the date filter (**1/2013 - 8/2014**)
 - **by Quarters** - displays axis with all quarters from the data set or quarters from the date filter (**I/2013 - III/2014**)
 - **by Years** - displays axis with all years from the data set or years from the date filter (**2013 - 2014**)
- **Date Units**
 - **Day of Week** - displays axis with days of week (**Su - Sa**)
 - **Day of Month** - displays axis with days of month (**1 - 31**)
 - **Day of Year** - displays axis with days of year (**1 - 366**)
 - **Week of Year** - displays axis with weeks of year (**1 - 53**)
 - **Month of Year** - displays axis with months of year (**1 - 12**)
 - **Quarter of Year** - displays axis with quarters of year (**I - IV**)
 - **Year by Weeks** - displays axis with years taking in consideration to weeks (**2013 - 2014**)

Additionally, you can place following time aggregations (if available):

- **Time Series**
 - **by Seconds** - displays axis with all seconds from the data set or seconds from the time filter
 - **by Minutes** - displays axis with all minutes from the data set or minutes from the time filter
- **Time Units**
 - **Seconds of Minute** - displays axis with seconds of minute (**1 - 59**)
 - **Minutes of Hour** - displays axis with minutes of hour (**1 - 59**)



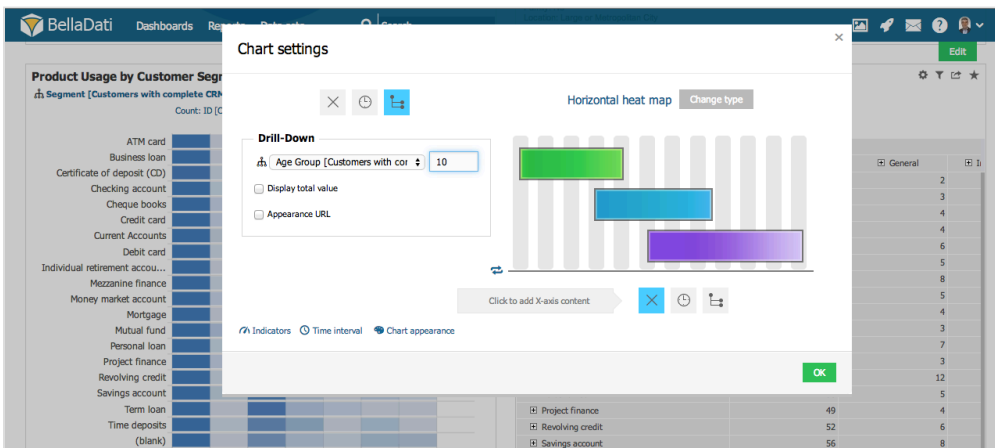
Setting Axis to Drill down path

You can place drill-down path on the axis. Click on the drill-down path axis button and select desired attribute. Additionally, you can edit:

- **Limit** for members in the drill down path. Displayed members depend on current sorting setup.
- **Total** value. Adds new member to the chart. Aggregation method depends on indicator settings - continue by [Displaying Indicators](#)




Display Total Value option does not apply to **Pie** and **Stack bar** charts.



Managing chart appearance

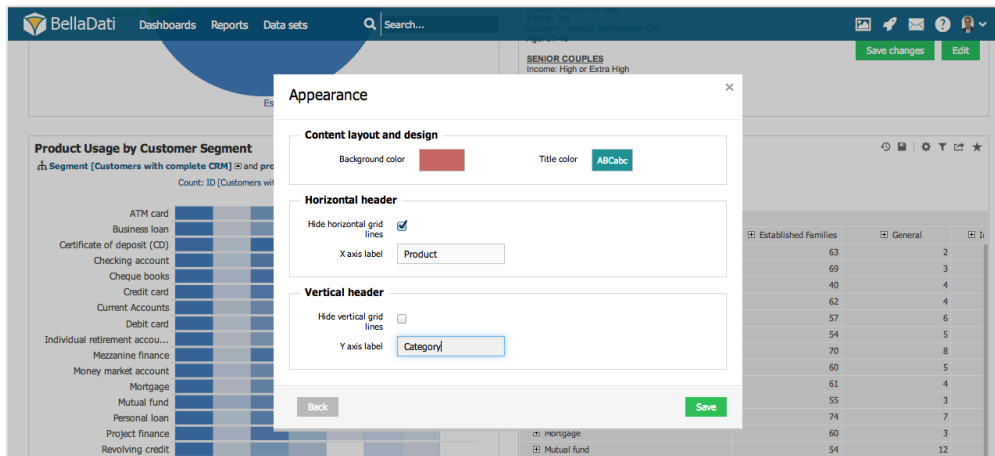
Chart appearance

You can edit Chart appearance from *Table settings* dialog after selecting **Chart appearance** or directly from the toolbox list.

 Each chart type includes relevant subset of appearance options.

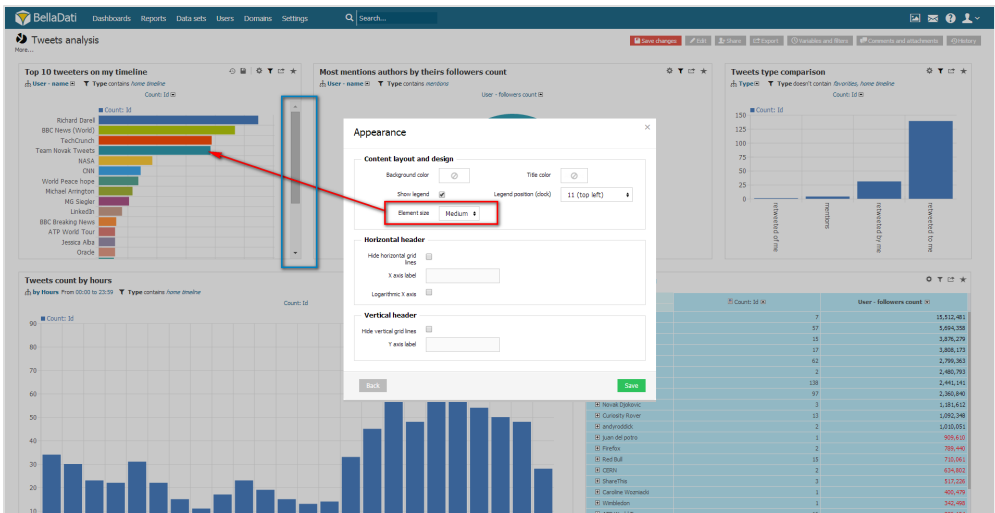
Appearance dialog allows you to:

- Specify **Color theme** and **Title color**
- **Sort attribute** allows you to order chart values by any attribute from the data set. You can choose ascending or descending order.
- **Background image**: Select image which will be displayed as background. You can upload new picture or select image from [Media gallery](#).
- **Separate to columns**. You can specify to how many columns will be the charts displayed within the view, if there are more columns (default is one chart per row). *Applicable for two or more indicators within one view.*
- **Value range**: You can override default lower and upper limits for displaying data on Y axis. *Note: Both limits must be set at the same time.*
- **Chart axis**: Hide grid on X axis, Hide grid on Y axis
- **Members on horizontal axis**: Applicable if drill-down and time dimension are set together. Date/time information are displayed on X axis by standard. Use this feature to display drill-down members on X axis instead (date/time information will be distinguished by legend).
- **Group** values
- **Show** values
- Define **Step size** (interval in which indicator plot values are displayed)



Displaying chart in "Scroll bar" mode

In case, your chart contains many elements (e.g. bars), and you won't let the system calculate the chart dimensions according to the view size, you can set the **Element size** parameter to specify the minimal element size. If the resulting size of chart is larger than current view size, scroll bars will appear. See the following example:



Attribute values appearance

Member appearance allows you to adjust and modify color, icon and translation of drill down members.

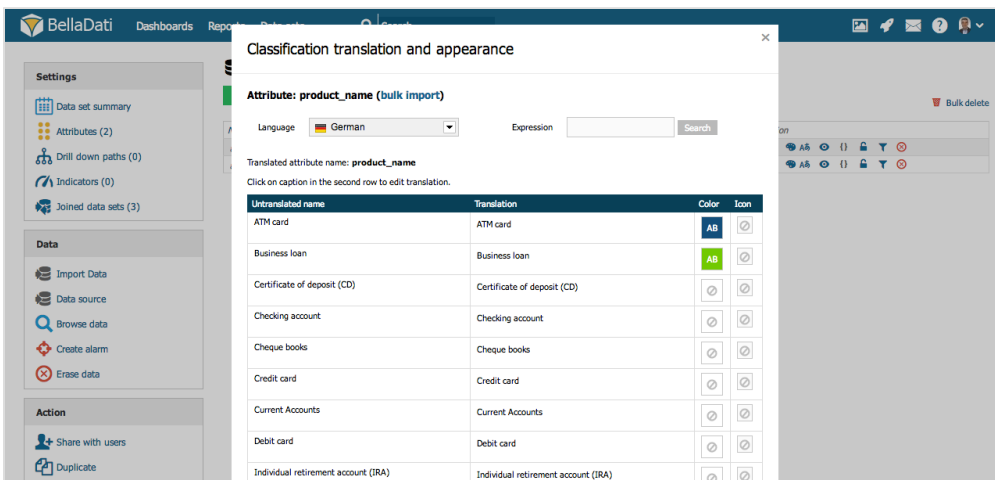
You can access **Member appearance** from **Attribute selection** dialog. Displayed members will refer to the attribute currently selected in the **Add drill down path** option.

Classification translation and appearance dialog enables:

- modifying node's font color and background
- assigning icon
- adjusting translations



These changes will affect all views and reports based on this data set. You must have editing permissions to perform such changes.



See [Members appearance](#) and [Attributes and members translation](#) for more details.

Creating Geo Maps

Geo maps allow you to visualize **Indicators** on the interactive maps.

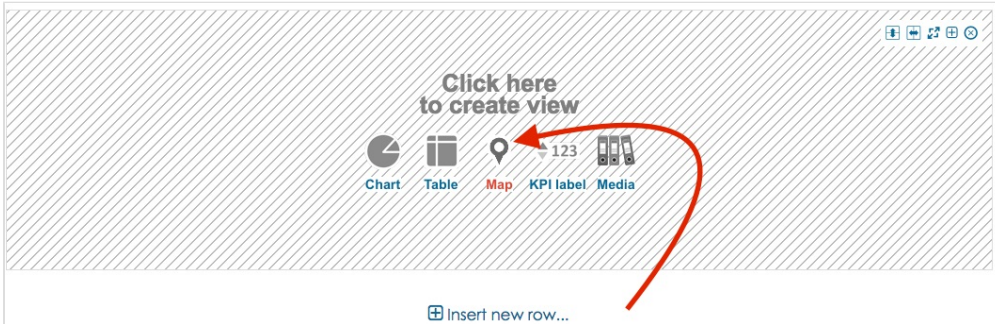


You need to be in [edit mode](#) in order to create new Geo map. Click on "Edit" in top report menu to activate edit mode.

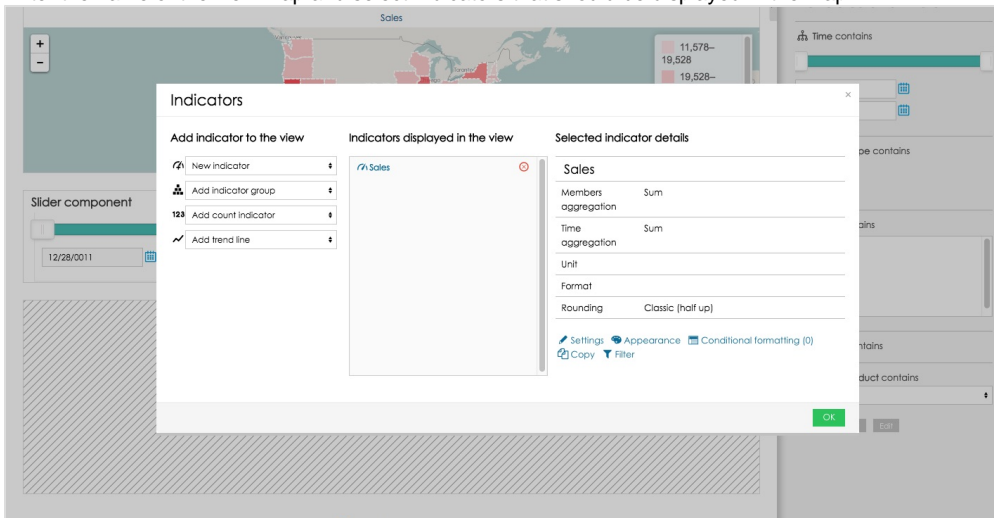


[Map Charts Geo Data](#) needs to be defined prior to creating new Maps. Or the Data Set has to include [GEO Point attribute](#)

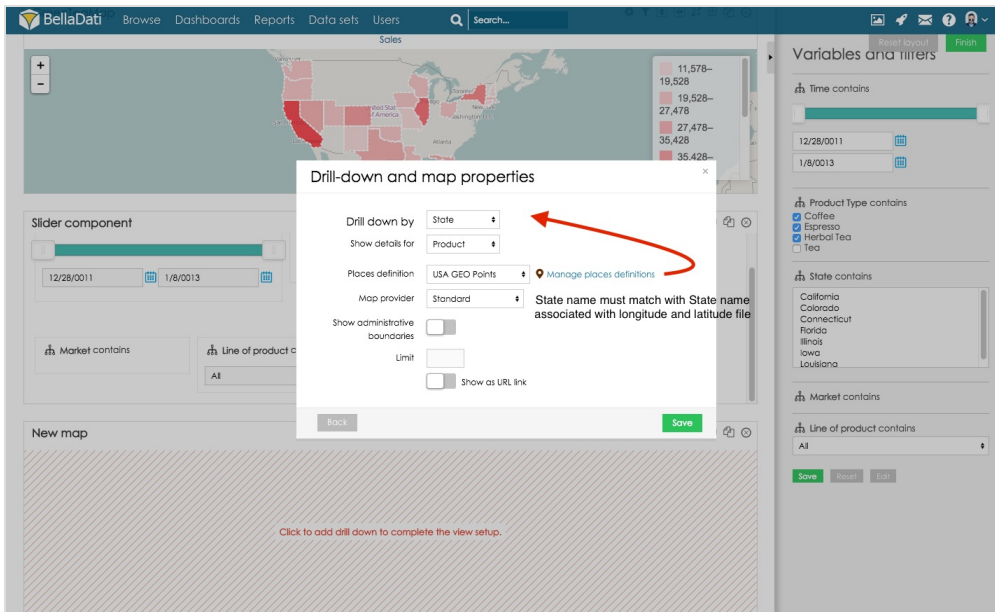
To add a new **Geo map**, hover over **Insert new row** and click on **Map** view type. New Map dialog will appear.



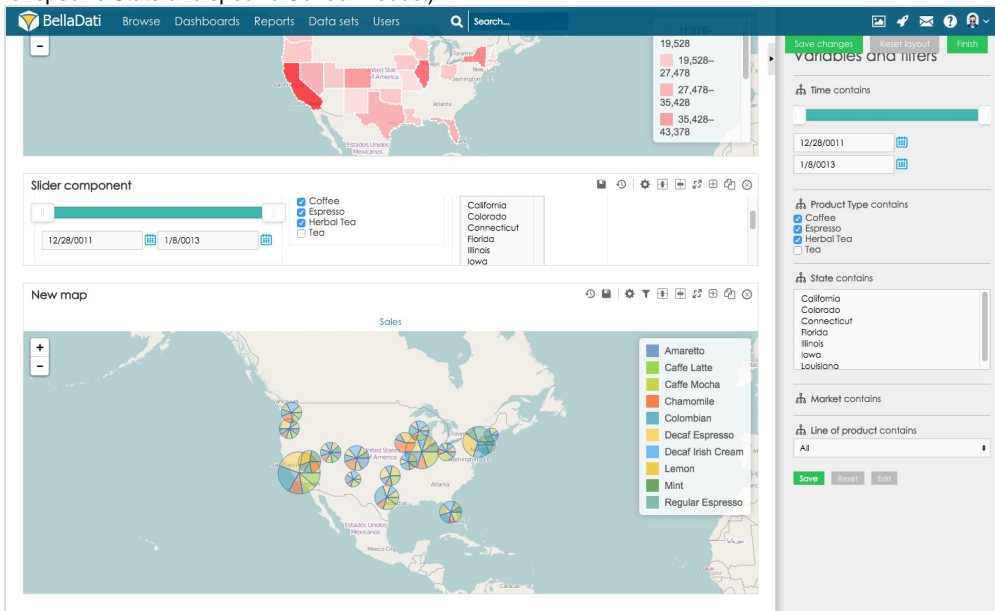
1. Enter the name of the New map and select Indicators that should be displayed in the map



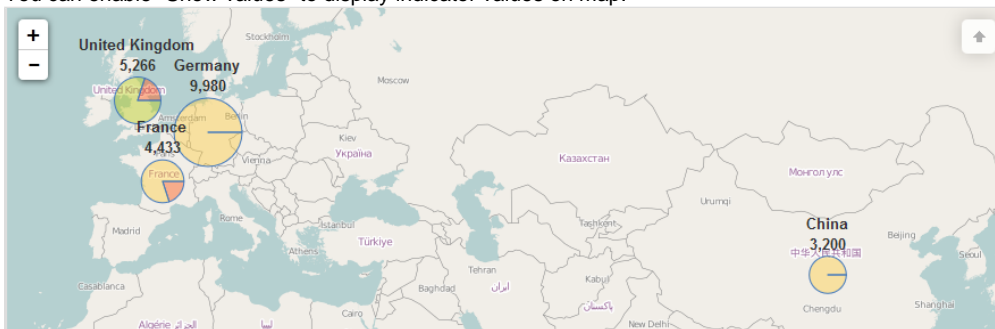
2. You have to add Drill down to complete the view setup. Drill down defines which attribute will be used to categorise and position data on the map (for instance Sales indicator for specific State)



3. You can select **Show details for**. This option will do drill-down for selected Indicator according the attributes (for instance Sales Indicator for specific State and specific Coffee Product)



4. You can enable "Show values" to display indicator values on map.



Example: The most common drill downs associated with Geo maps include Country, Region or City. When Country attribute is selected:

- Indicators are grouped according particular country.
- Values are placed to proper position corresponding with country location.

Drill Down

You can specify drill down from *Chart settings*. To access dialog:

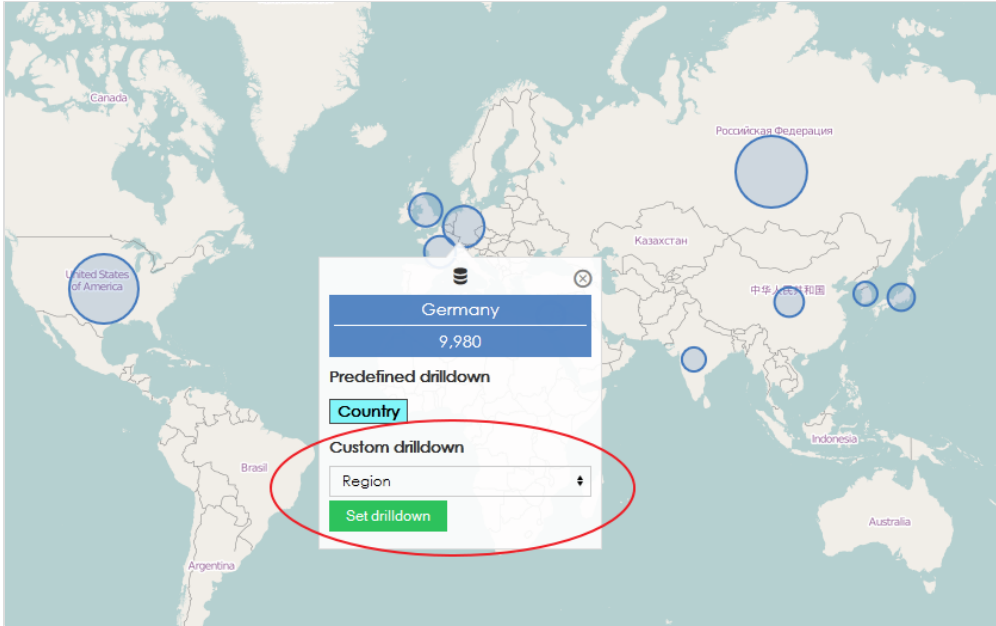
1. click on **add drill down to complete the view setup** link in the created view.
2. enter *Map chart settings* dialog from the toolbox and click **Add drill down path**.

Custom drill down

As in chart and tables, you can set **another drill down** to change the granularity of map. To do so you have to specify **default GEO location mapper** in **Attribute settings** in **Data Set**.

If it is all set, you can click on any drill down member, select Custom drill down Attribute and click on Set drill down.

If **Predefined drilldown** (for example Country > City) is also set, it is possible to drill-down with left mouse click. Default pop-up window is then available with right click.



You can download video of setting custom drill down [here](#)

Map settings

Map settings dialog allows you to:

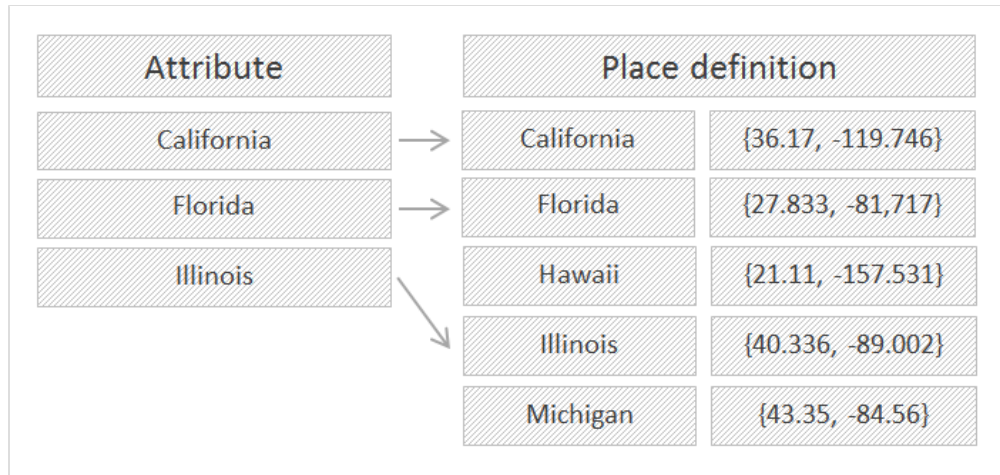
- Specify **Attribute or GEO Point** to be used for aggregating and categorizing indicator's values.
- **Show details for** monitored Indicator
- Select **Place definition** for proper positioning and visualization of data.
- **Manage places definitions** - continue by [Managing GEO Data](#).
- Select desired **map provider**.
- Setup **Limit** for members in the drill down path. Displayed members depend on current sorting setup.

Associating Attribute with Place definition

There is a very thin connection between **drill down** and **place definition**. While **drill down** maintains indicator's values and their categorization, **place definition** keeps pairs of location identification and its coordinates. BellaDati automatically connects attribute's members with places and therefore enables their proper positioning.

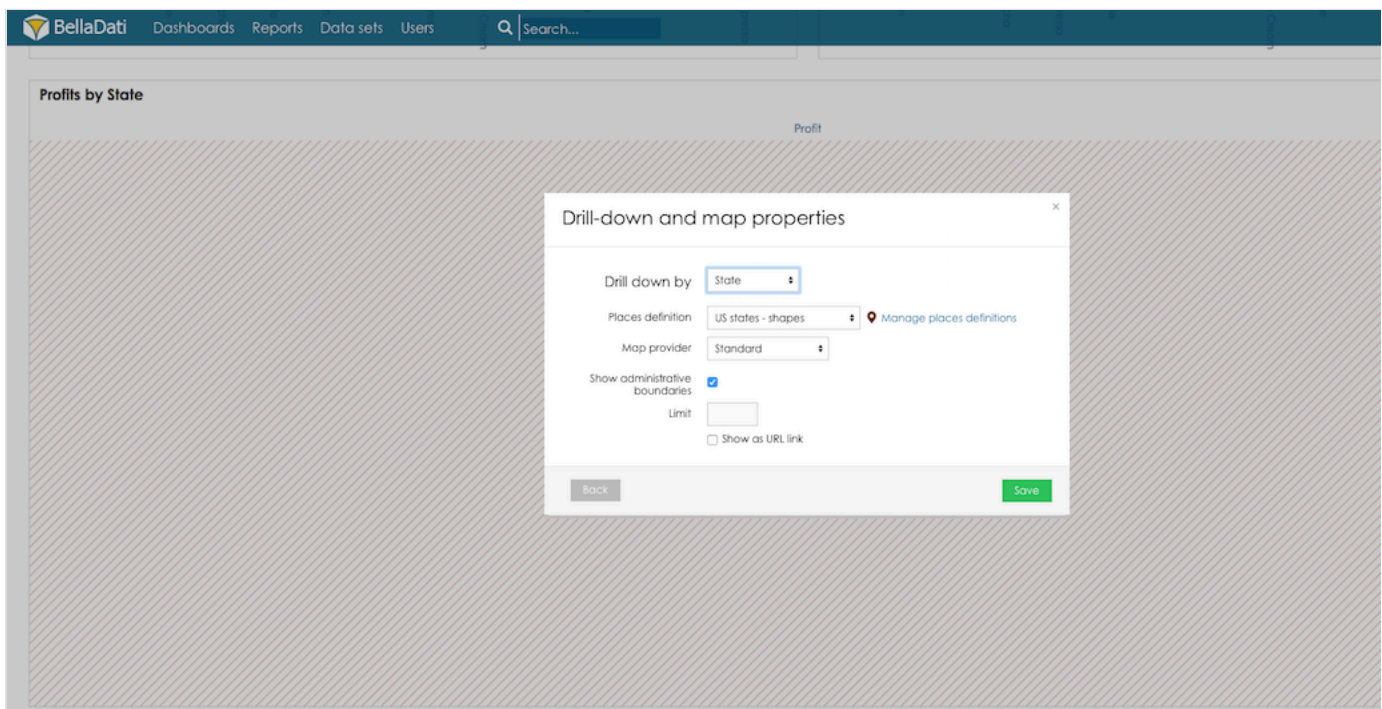


You should always select only attributes which can be associated with an existing **Place definition**. In case of attribute with no related **Geo data**, indicators will not be properly displayed.



Drill down option allows you to define:

1. how displayed indicators will be grouped and categorized.
2. what granularity will visualized data have.
3. which mapping background will be user (Standard, Black & White, Gray Scale, Outdoors, Cyclo, etc.)



Place definition option is used to associate selected attribute with **Geo data**.

Select **Place definition** according to:

1. **Drill down** path you selected.
2. **Style** of visualization you want to apply (point vs. region).

Each **Place definition** includes pair on place *identification* and its *coordinates*.

Coordinates can be represented as:

- **Points**. In point definition, location is identified by pair of values representig exact longitude and latitude.
- **Regions**. In region definition, area is restricted with polygon, composed of set of longitude and latitude values.

For more information about **Place definition** continue by [Managing GEO Data](#).

Using GEO Points

GEO point is a special attribute type holding latitude and longitude coordinates of a special location. GEO points are most convenient to use when location information are stored directly in imported Data Set. It automatically generated its own place definition and can be directly used in reports.

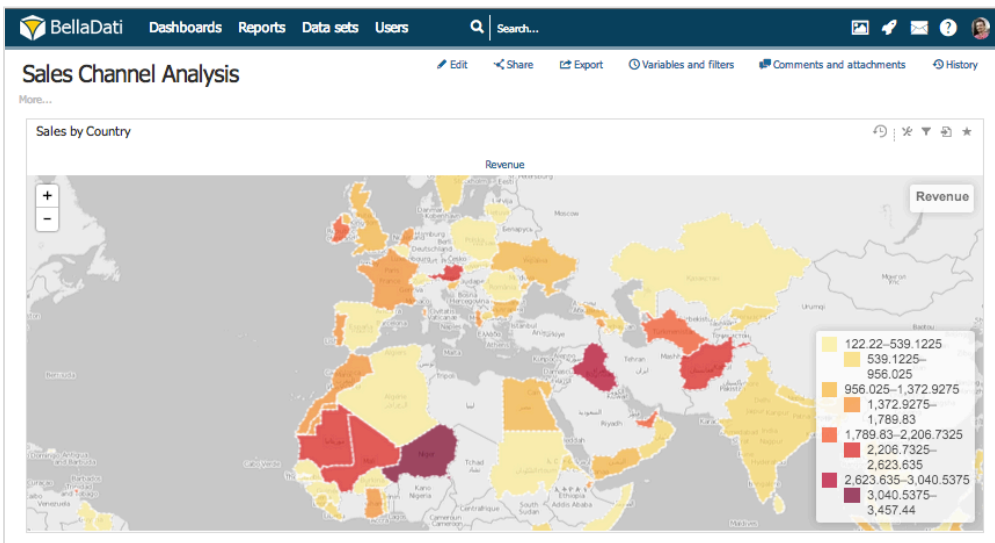
To use geo points:

1. Select **GEO point** attribute in Drill-down settings
2. Find relevant **Point definition**
3. Choose desired **map provider**

Map management

You can perform additional operations in upper right corner of the inserted **Map** view:

- Table settings
- Filter setting - continue by [Using Filters](#)
- Export view - continue by [Exporting View](#)
- Add to dashboard
- Move map
- Duplicate map
- Delete map

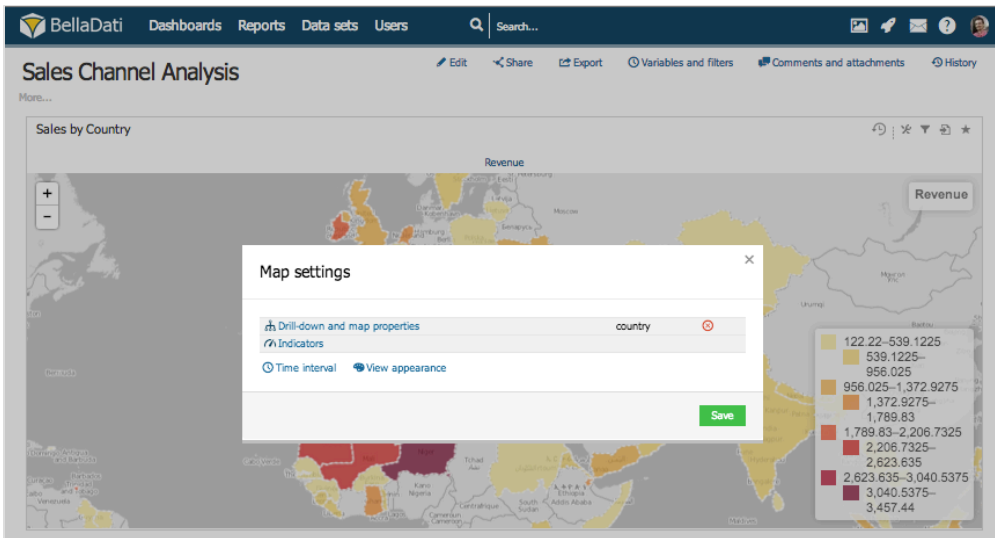


Map settings

Click on **toolbox** icon to enter *Map settings* dialog.

Map settings dialog shows **Indicators** and **Drill down path** currently applied to the map. BellaDati allows you to:

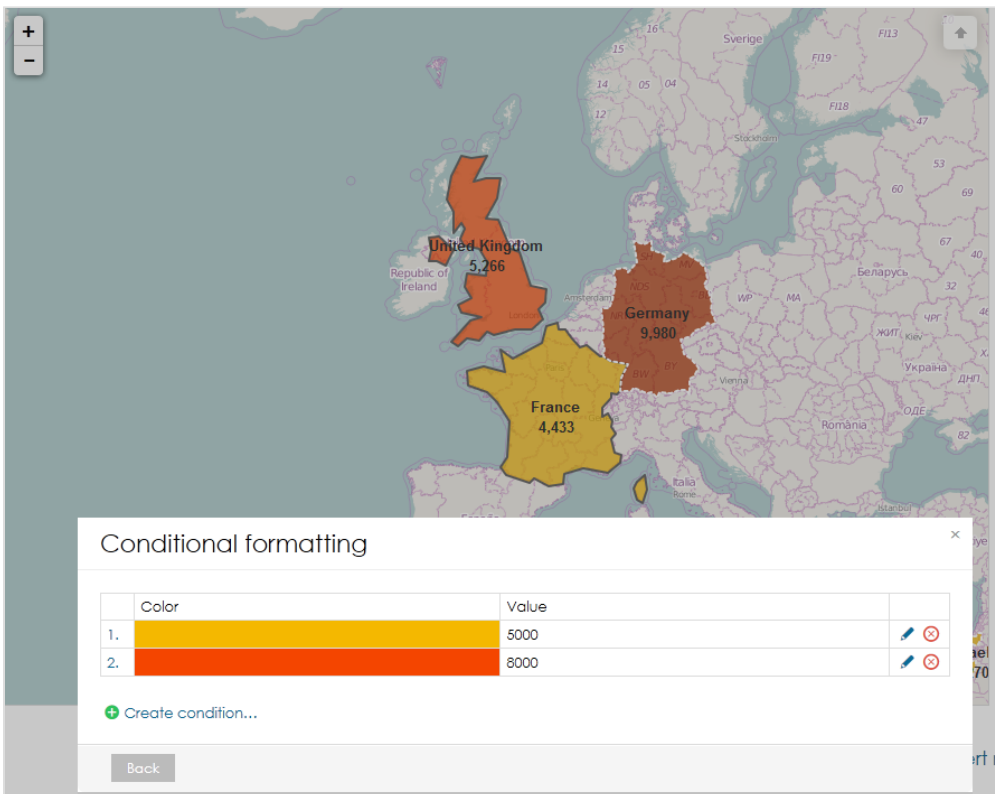
- **Enter** indicators or drill down path dialogs.
- **Remove** drill down path.
- **Add** time axis - continue by [Setting Date Interval](#).
- Add conditional formatting - see next paragraph.



Conditional formatting

There are two types of **conditional formatting** available for maps.

- Conditional formatting for indicators - [Displaying Indicators](#).
 - Background color is applied for shapes and foreground color for points.
- Map conditional formatting - chosen color is applied to drill-down member whose value is smaller or equal than specified value. You can create your own color scheme by using multiple rules.



Creating KPI labels

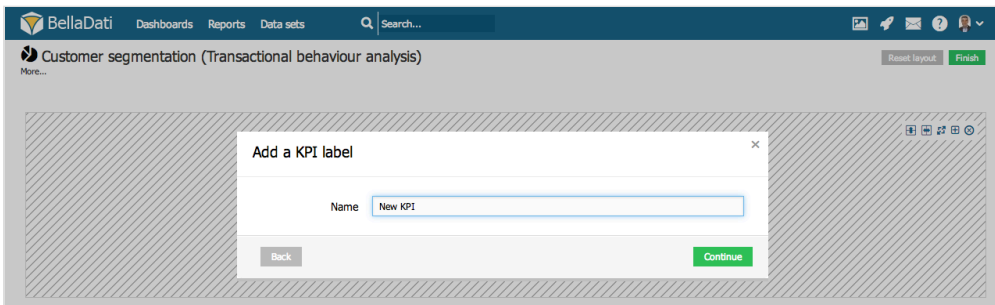
KPI labels allow you to clearly monitor important indicator values.



You need to be in [edit mode](#) in order to create new tables. Click on "Edit" in top report menu to activate edit mode.

To add a new **KPI label**, hover over free place and click on **KPI label** view type. The *Add a KPI label* dialog box will appear.

1. Enter name of the new KPI label.

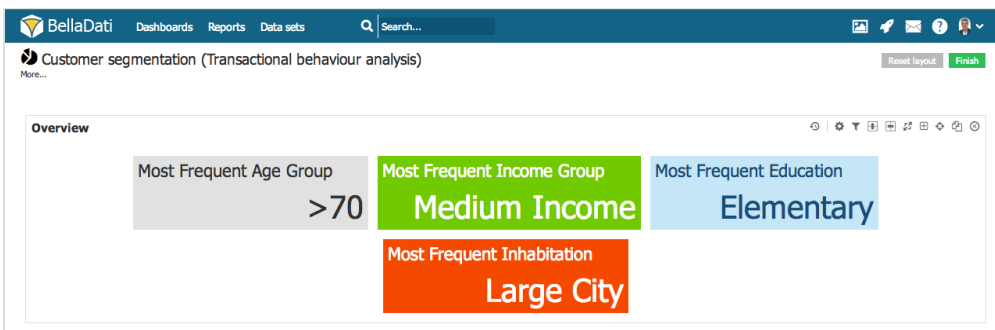


Confirm new **KPI label** view by clicking on green **Add** button. BellaDati will guide you through additional setup.

- **Indicators:** select and edit displayed indicators in the KPI label - continue by [Displaying Indicators](#).

Text-based KPI labels

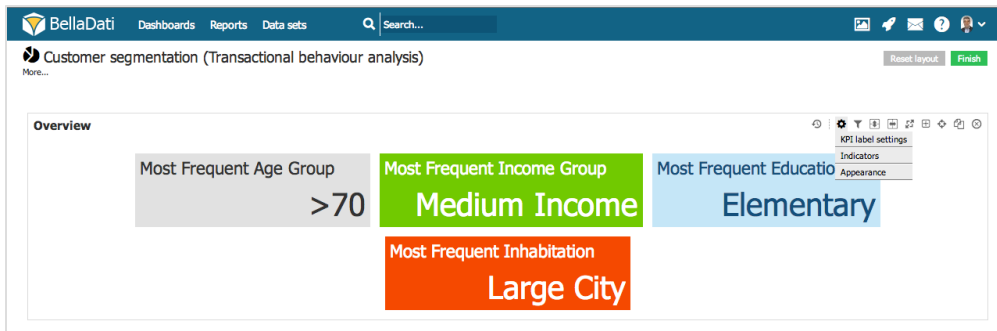
KPI label also allows displaying text values. Refer to formulas [use case](#) to learn more about text-based KPIs.



KPI label management

You can perform additional operations in upper right corner of the inserted **KPI label** view:

- KPI label chart settings
- KPI label appearance settings - [Setting appearance of KPI Label](#)
- Filter setting - continue by [Using Filters](#)
- Export view - continue by [Exporting View](#)
- Add to dashboard
- Move KPI label
- Duplicate KPI label
- Delete KPI label



KPI label settings

Click on the toolbox icon to enter *KPI label chart settings* dialog.

KPI label chart settings dialog allows you to:

- Edit **Time interval** if you need to restrict time period of displayed data - continue by [Setting Date Interval](#).
- Edit **Indicators** by entering *Indicators* dialog - continue by [Displaying Indicators](#)

Setting appearance of KPI Label

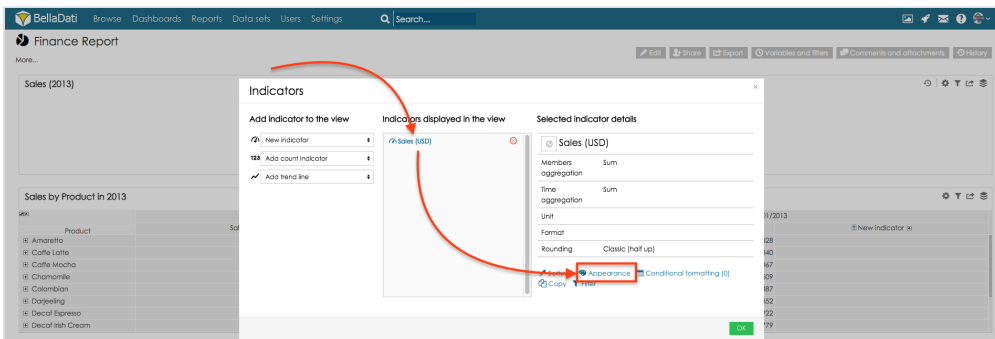
In KPI labels you can modify

- basic color
- size
- if the label should be bold
- background shape
- symbol (infographic) which will be displayed in the shape

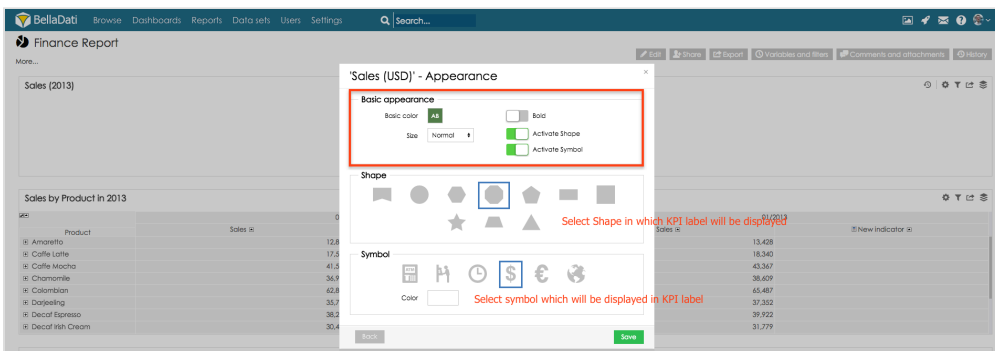
Setting is done through **Indicators** settings



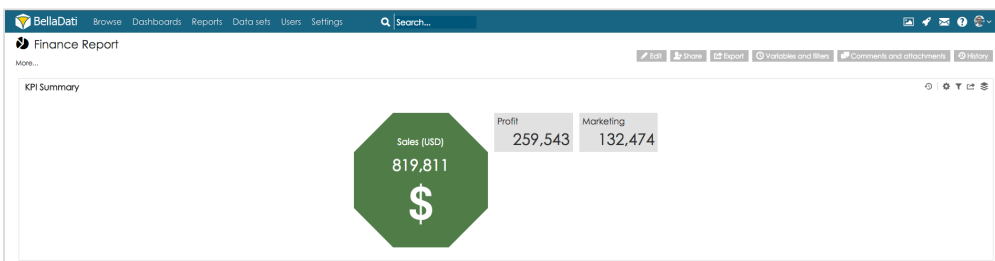
Select **Indicator**, that you are viewing in the KPI Label and click on **Appearance** button.



Modify color, size, shape, symbol and click **Save**.




Fields Background shape and Symbol are not Obligatory, if they are not selected report will be displayed in default settings. Example of default KPI label settings is displayed on the picture below:




Adding Custom Content

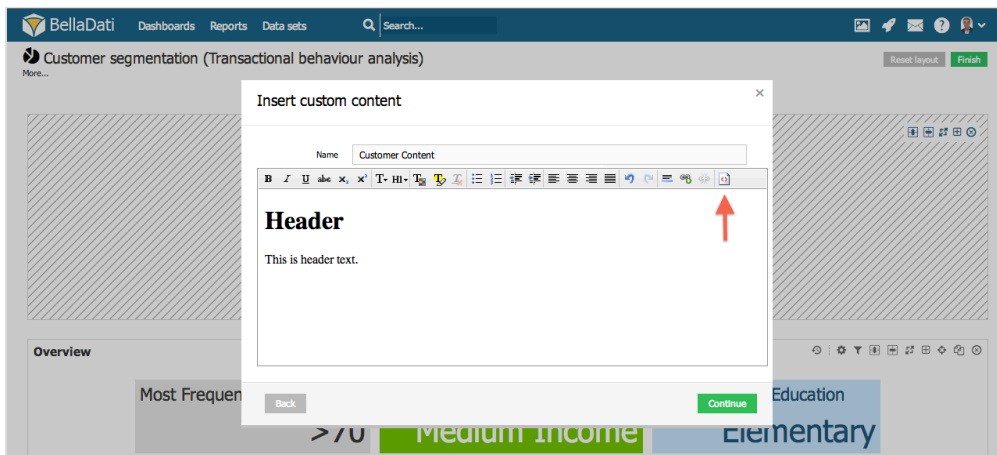
Custom content view allows you to enrich your report with arbitrary content.

 You need to be in [edit mode](#) in order to create new custom content. Click on "Edit" in top report menu to activate edit mode.

To add a new **Custom content**, hover over free place and click on **Custom content** view type. The *Insert custom content* dialog box will appear.

1. Enter name of the new content.
2. Leverage **Rich text editor** to properly format your content.

 You can switch to native HTML by selecting **HTML** icon in the right corner of the toolbox.




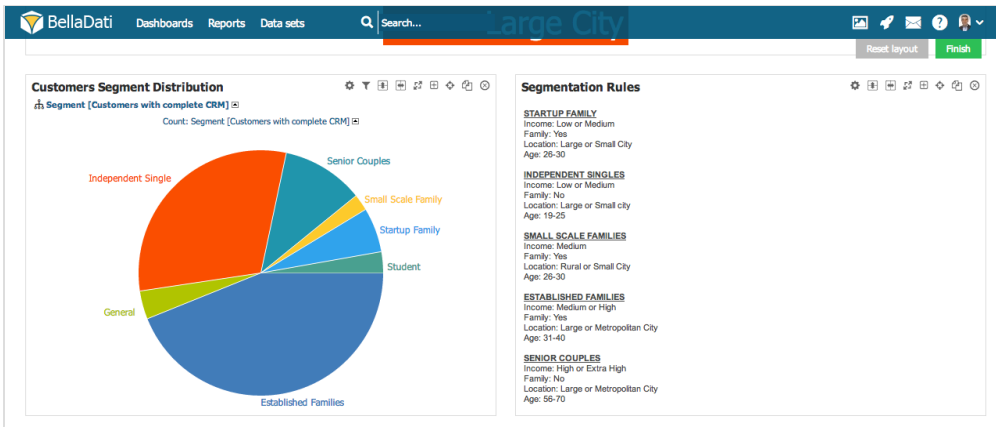
BellaDati also allows you to set up:

- Content color theme
- Title color

Through **Custom content** you can add objects such as:

- Hyperlinks
- Images
- Videos
- Email feed
- RSS feed
- Social media feeds

 Refer to particular service for detail guide how to embed content into web page.



Displaying Indicators

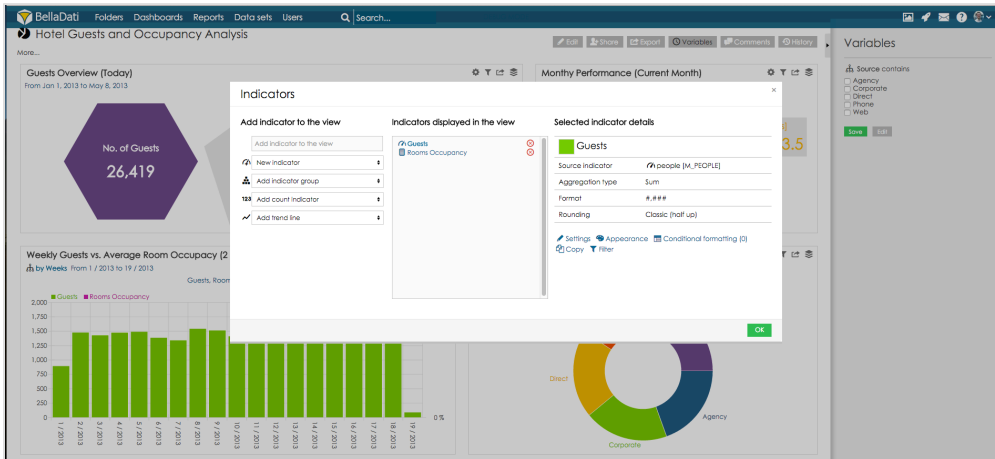


This option is related to views. Always refer to [views](#) or particular view types ([table](#), [chart](#), [Geo map](#) or [KPI label](#)) before proceeding with this section.

Indicators represent values displayed in the created view.

You can add indicators into view from **Indicators** dialog box. *Indicators* dialog box is separated into three columns:

- List of **Available indicators**
- List of **Indicators displayed in the view**
- Additional **Indicator details**



Adding indicators

There are two options for adding indicators to the view:

1. **Add from existing:** Click on the desired indicator from the list of **Available indicators**. Indicator will be moved to the **Displayed indicators** section.
2. **Create new:** Type the name of new indicator into *New indicator* input form of **Displayed indicators** section and click green **plus** sign. To learn more about creating new Indicators, please follow with [Using Formulas](#).



You can remove displayed indicators by clicking on the red cross sign.

Indicator details

Indicator details are accessible in the right column after selecting displayed indicator from **Displayed indicators** section.

Indicator details include:

- Name
- Source indicator name
- Color
- Unit
- Format
- Rounding
- Members aggregation

You can perform following tasks to edit displayed indicator:

- Edit **Indicator setting**
- Edit **Appearance settings**
- Create **Conditional formatting**
- **Duplicate** indicator

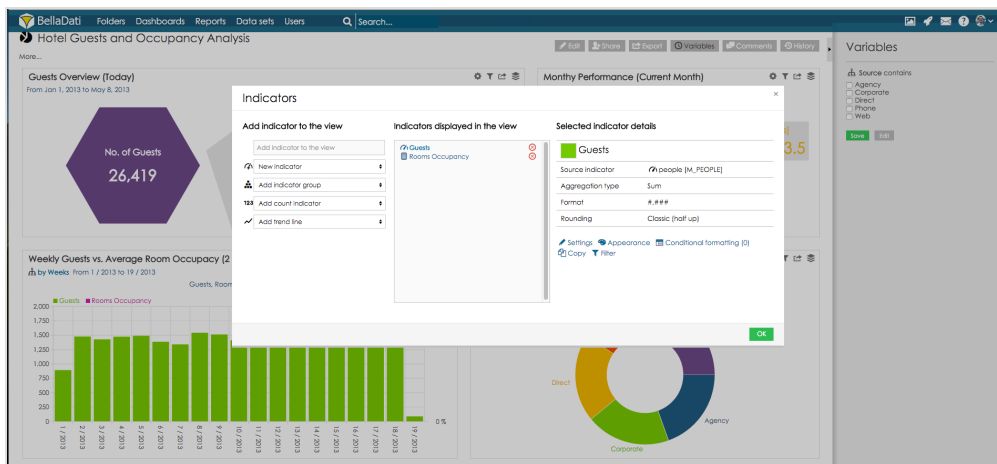
Edit indicator setting

Click **Indicator setting** to enter edit dialog. From the **Indicator setting** dialog you can edit:

- **Name** of the indicator.
- **Unit** to be displayed with the indicator.
- **Format** of the indicator data. Click **show help** to expand format syntax hints.
- **Rounding** of decimal values.
 - Select **Classic (half-up)** option to apply traditional rounding function.
 - Select **Always up** to automatically round data to the higher values.
 - Select **Always down** to automatically round data to the lower values.
- **Members aggregation** to define how aggregated values should be processed. Click **show help** to expand members aggregation hints.
 - Select **Sum** to display total value of indicator records.
 - Select **Average** to display average value of indicator records.
 - Select **Number of records** to display count of indicator records.
 - Select **Minimum** to display minimal value from indicator records.
 - Select **Maximum** to display maximal value from indicator records.
- **Empty values** processing. Select checkbox to edit custom value for empty values replacement.
- **Show as URL link**. Define URL link to create drill-through.
- **Numeric transformation**. Choose value from combobox to select how the value should be transformed (show percent, thousandth, ten thousandth, millionth, ten millionth, hundred millionth or billionth).



You can also use **statistical functions** except basic members and time aggregations. Proceed with [Using Core Statistical Functions](#) to learn more.



Edit appearance settings

Click **Appearance settings** to enter edit dialog. From the **Appearance settings** dialog you can edit:

- **Font color**. Click **basic color** icon to expand the pallet of predefined font and background colors.
- **Font style**. Select **Bold** checkbox to make the indicator values appear in bold.



Select **Default** from **basic color** pallet to reset the font color.

You can also edit indicator color directly from indicator detail. Click on the box next to the indicator name and choose color.

Edit conditional formatting

Click **Conditional formatting** to enter edit dialog. There are two options for creating conditional formatting.

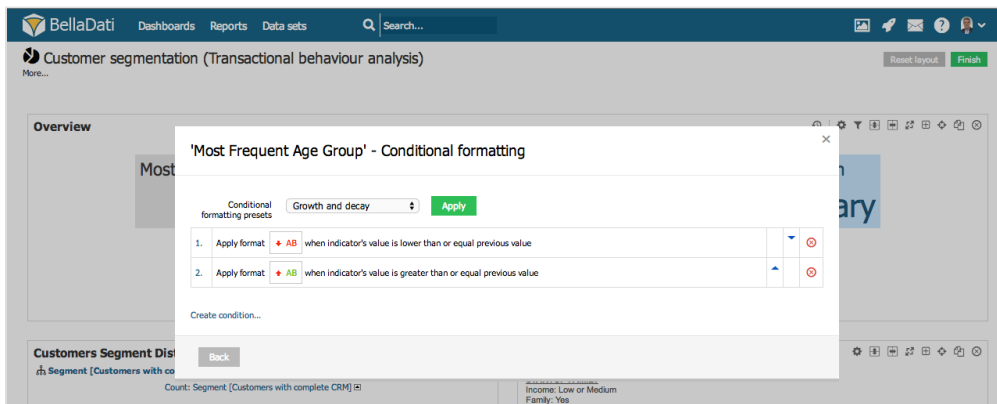
Apply **preset conditional formatting** styles.


- Select **Black and red numbers** to apply discrete formatting based on provided **Threshold** values.
- Select **Growth and decay** to apply continuous formatting based on indicator values.
- Select **No conditional formatting** to reset any defined styles.

Create own conditions and define styles. Click **Create condition** link to expand condition options.

- Select **font color and background** to be applied to conforming indicator values from predefined palette.
- Select **symbol** to be append to conforming indicator values.
- Select **condition** to evaluate indicator values. BellaDati offers following conditions:
 - greater than
 - lower than
 - greater by
 - lower by
 - greater by (%)
 - lower by (%)
- Insert **value** related to condition or select **previous value** as source for evaluating the condition.
- Select **Show growth/decrease in %** checkbox to enrich indicator values with relative changes.
- Condition can be also defined by formula. Click on dropdown **indicator** and choose value **formula**. Conditional formatting will be applied when this formula returns true.

Click **Add** button to confirm create conditional formatting. You can add multiple conditions by repeating the procedure.



 You can remove conditional formatting by clicking on the red cross sign.

Using Core Statistical Functions



Make sure that you are familiar with [Displaying Indicators](#) before proceeding with this section.

BellaDati allows you to apply statistical functions on existing indicators.

Available Functions

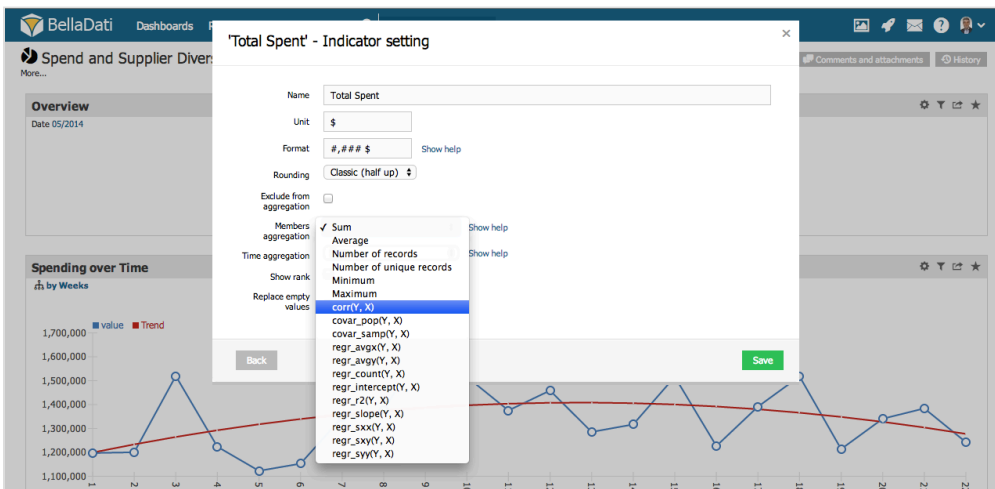
Following functions are currently available in BellaDati:

- **corr(Y, X)** Correlation Coefficient
- **covar_pop(Y, X)** Population Covariance
- **covar_sample(Y, X)** Sample Covariance
- **regr_avgx(Y, X)** Average of the independent variable ($\text{sum}(X)/N$)
- **regr_avgy(Y, X)** Average of the dependent variable ($\text{sum}(Y)/N$)
- **regr_count(Y, X)** Number of input rows in which both expressions are nonnull
- **regr_intercept(Y, X)** y-intercept of the least-squares-fit linear equation determined by the (X, Y) pairs
- **regr_r2(Y, X)** Square of the correlation coefficient
- **regr_slope(Y, X)** slope of the least-squares-fit linear equation determined by the (X, Y) pairs
- **regr_sxx(Y, X)** $\text{sum}(X^2) - \text{sum}(X)^2/N$ ("sum of squares" of the independent variable)
- **regr_sxy(Y, X)** $\text{sum}(X*Y) - \text{sum}(X) * \text{sum}(Y)/N$ ("sum of products" of independent times dependent variable)
- **regr_syy(Y, X)** $\text{sum}(Y^2) - \text{sum}(Y)^2/N$ ("sum of squares" of the dependent variable)

Applying Functions


To use statistical functions:

1. go to **Indicators**
2. select **first Indicator (Y)** you want to use
3. go to **Indicators settings**
4. in **Members aggregation** select desired statistical functions
5. from the available drop-down select **second Indicator (X)**



Exporting View


Export allows you to store reports or views permanently outside of BellaDati for your own presentation or specific analysis.

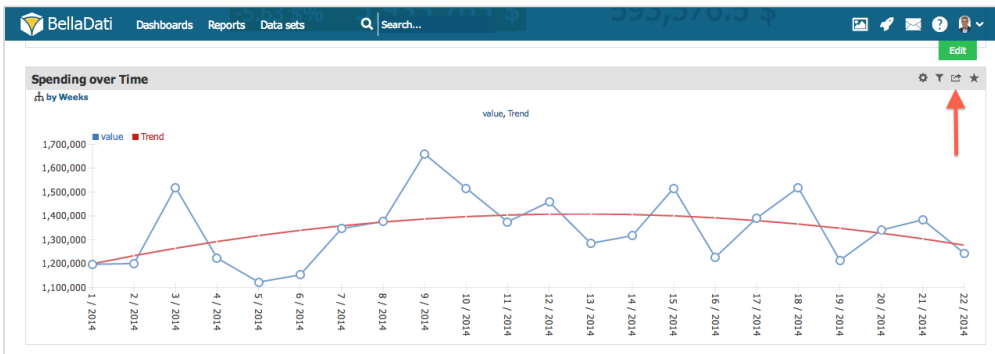
 **Export** option is available for Chart, Table and KPI label view.

You can access **Export** option from the toolbox in the upper right corner of the view.

Export dialog offers exporting view to following file formats:

- **PDF**
- **PNG**
- **Microsoft Excel**
- **Microsoft PowerPoint**
- **Embed to page** - continue by [Sharing Report](#) for more details.

 Export to Microsoft Power Point is only available for the whole report.



Exporting to PDF

Set Export type to **PDF**.

Export view dialog allows you to set:

- **Size:** Available options include: **A1**, **A2**, **A3** and **A4**.
- **Orientation:** Available options include: **Portrait** and **Landscape**.

Exporting to PNG


Set Export type to **PNG**.


Export view dialog allows you to set:

- Image **Width**
- Image **Height**

Exporting to Microsoft Excel

Set Export type to **Microsoft Excel**.

 Export to **Microsoft Excel** option is available only for **Table** view type.

 Maximum table rows count in the export is currently limited to 1000 due to performance reasons. This limit can be raised for BellaDati Enterprise tariffs or licenses.

Publishing View

Publishing allows you to embedd existing views to your web based application, company extranet or publicly on the Internet. You can choose if the reports should be displayed with or without the need of being logged in.


Set Export type to **Embed to page**. You can select from the following window sizes:

- **Small:** (250x180)
- **Medium:** (500x300)
- **Large:** (fits window width)

BellaDati will generate **iFrame** object you can insert into your page or portal.

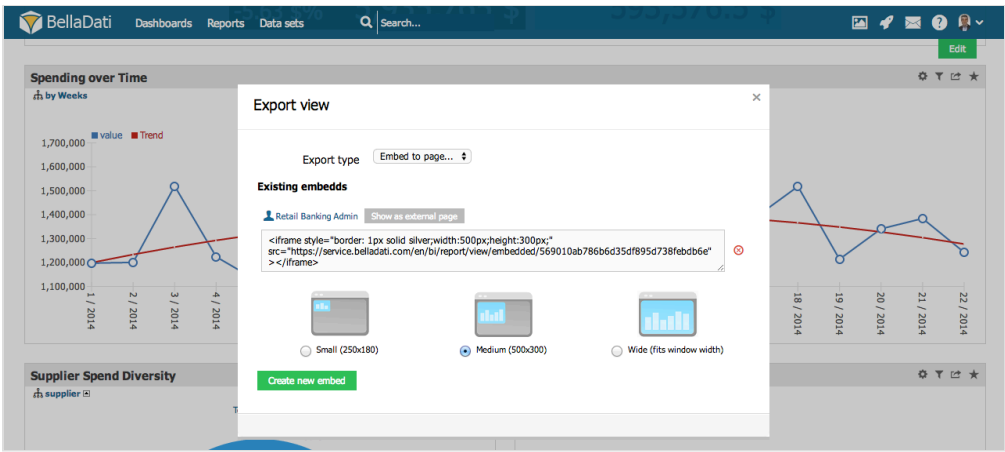
From *Export view* dialog you can also:


- Show generated **iFrame** object in the new tab or window.
- Change the size of generated object.

 Domain option "Public sharing" must be enabled to allow public sharing. It is recommended to test iFrames on another computer, browser or after logout.

Customization parameters

hideLink=true	Disables displaying of the source report link in the iFrame
hideHeader=true	Disables displaying of report header in the iFrame



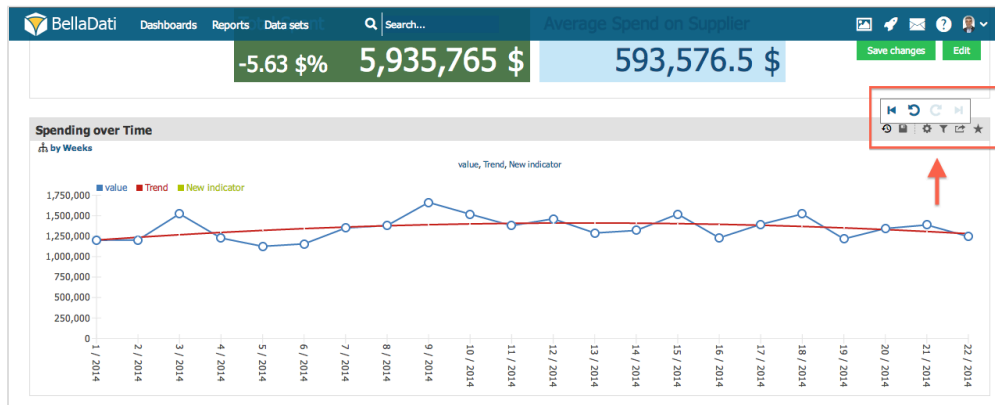
 See also [complete REST API documentation](#) for detailed information about BellaDati platform embedding options.

Reversing Changes (Undo & Redo)

BellaDati allows you to move among changes you made to the view.

Once any changes occur, BellaDati displays **Clock** icon. Hover over icon to:

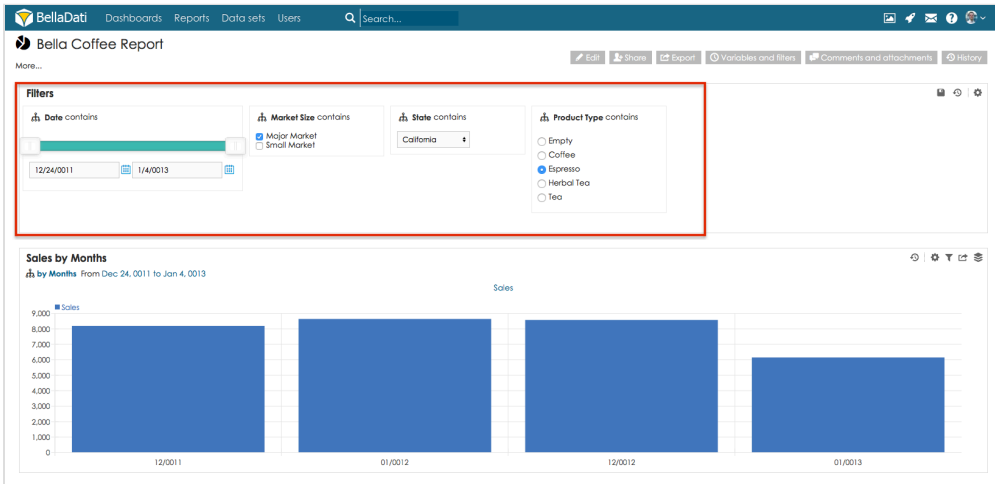
- Go to the **initial settings**
- Move **one change back**
- Move **one change forward**
- Move to the **last settings**





Changes are available only during the session.

Creating Filters

Filter view allows you to dynamically **modify content** of the report. It is feature, that makes filtering and drilling into the report very straightforward for report viewers. Since all filter options, data and time are accessible via custom components. Report creator can apply these components into the report. See screenshot below.



 You need to be in [view mode](#) in order to add variables and filters.

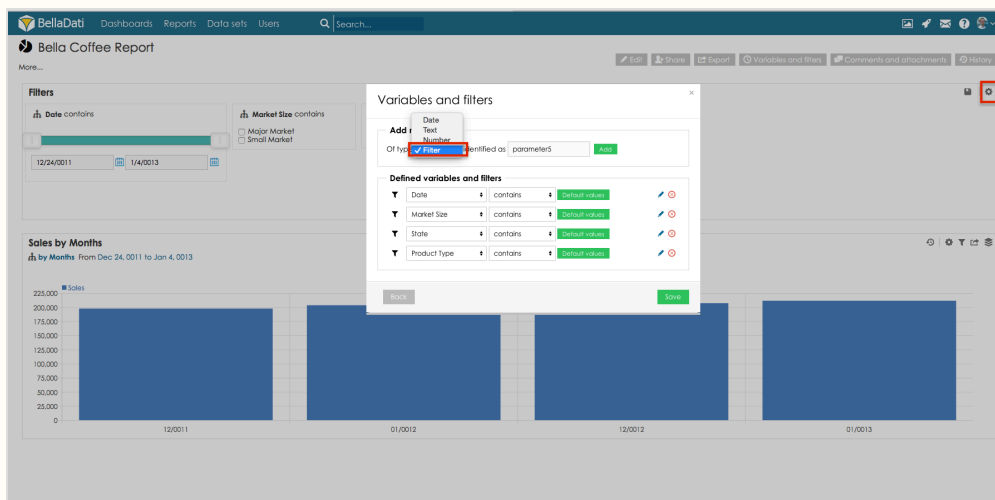
 Only [report author](#) or [report editor](#) can create report variables and set their default values. Every other user that has access to the report, can only change report variable values.

List of custom Filter components includes.

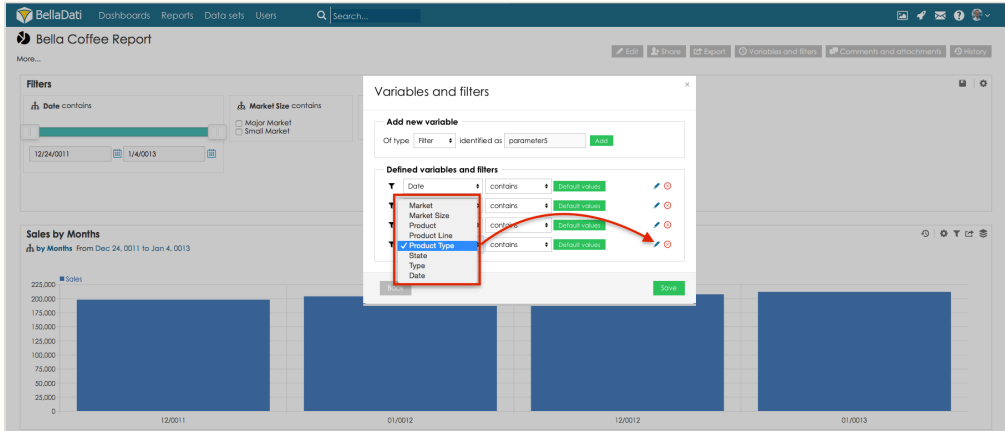
- Slider
- Multiple select
- List
- Check box
- Radio buttons
- Text

Creating filter and GUI component

Click on option icon in the top right corner of the filter view to open *Variables and Filters dialogue*. In this dialogue choose type *Filter*.

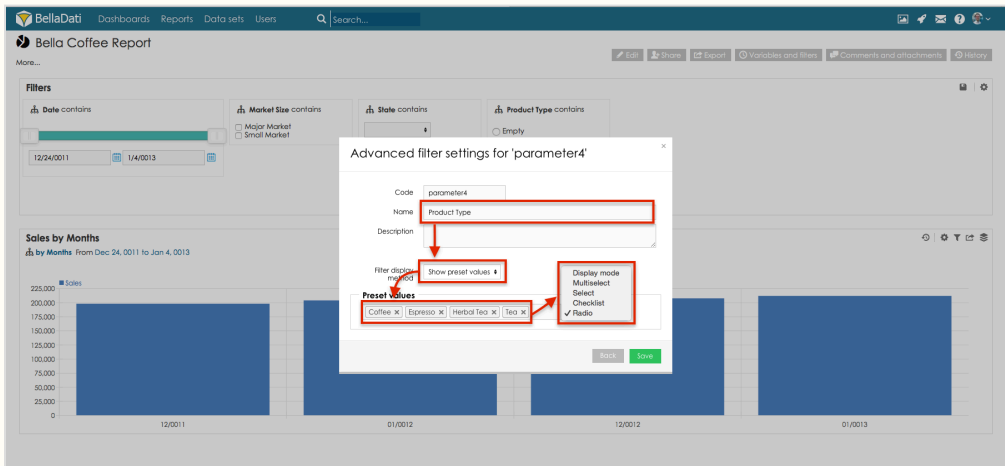


Select from which attribute should be the component created and go to *Filter settings*.



Selecting the component

Select name for your component. Choose display option *Display preset values* or *Display available values*. Select attribute members, that should be visible inside of your component and select type of the component.



Conditional formatting

There are two types of conditional formatting in BellaDati:

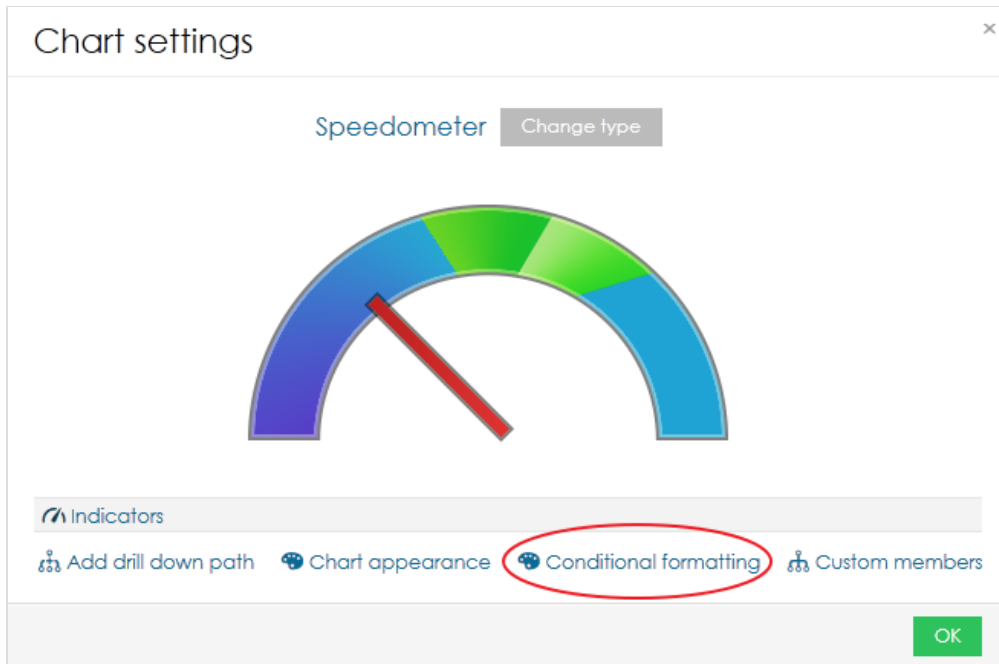
1. Indicator conditional formatting - continue by [Displaying Indicators](#)
2. View conditional formatting

View conditional formatting

This form of conditional formatting is available for charts and maps only and it has a little different function in each case.

Chart conditional formatting

This type of conditional formatting is used for highlighting values on axis or scale, for example if you want to highlight some critical value. It is accessible from chart settings.



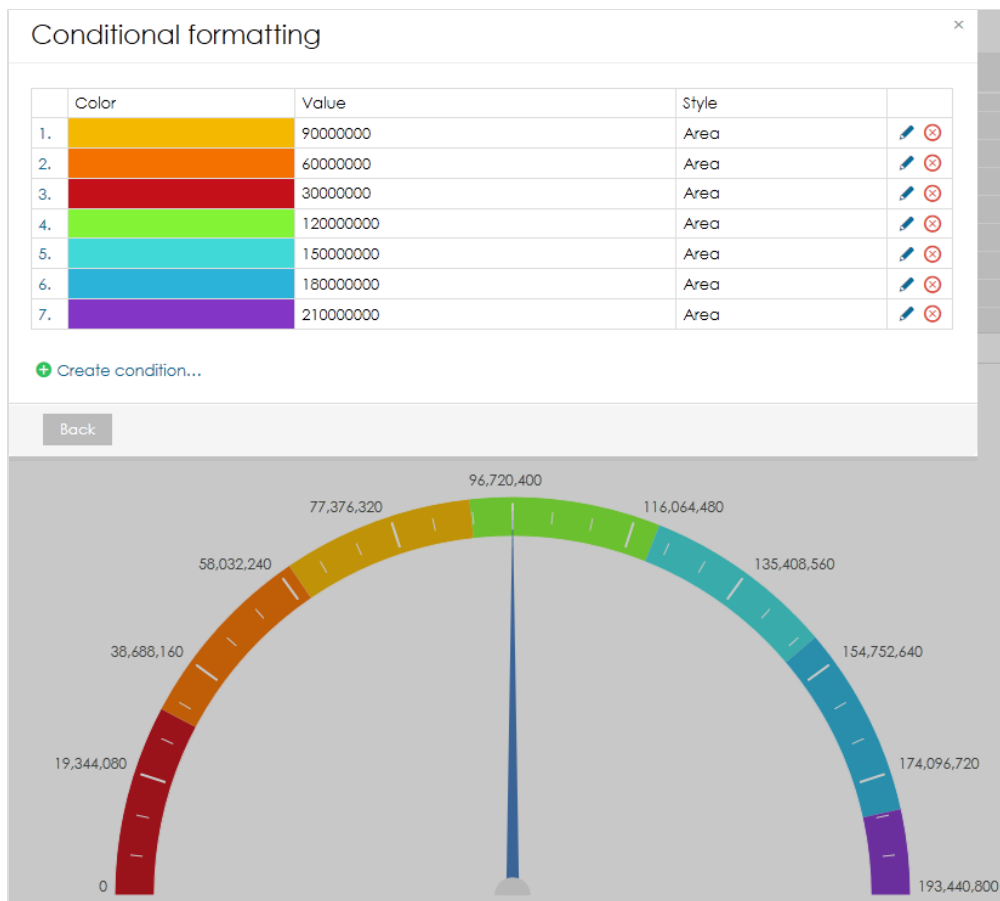
It is available for these charts:

- Bar chart
- Stack bar chart
- Line chart
- Horizontal bar chart
- Horizontal stack bar chart
- Combined chart
- Thermometer
- Speedometer
- Bullet chart
- Horizontal bullet chart

Step-by-step tutorial

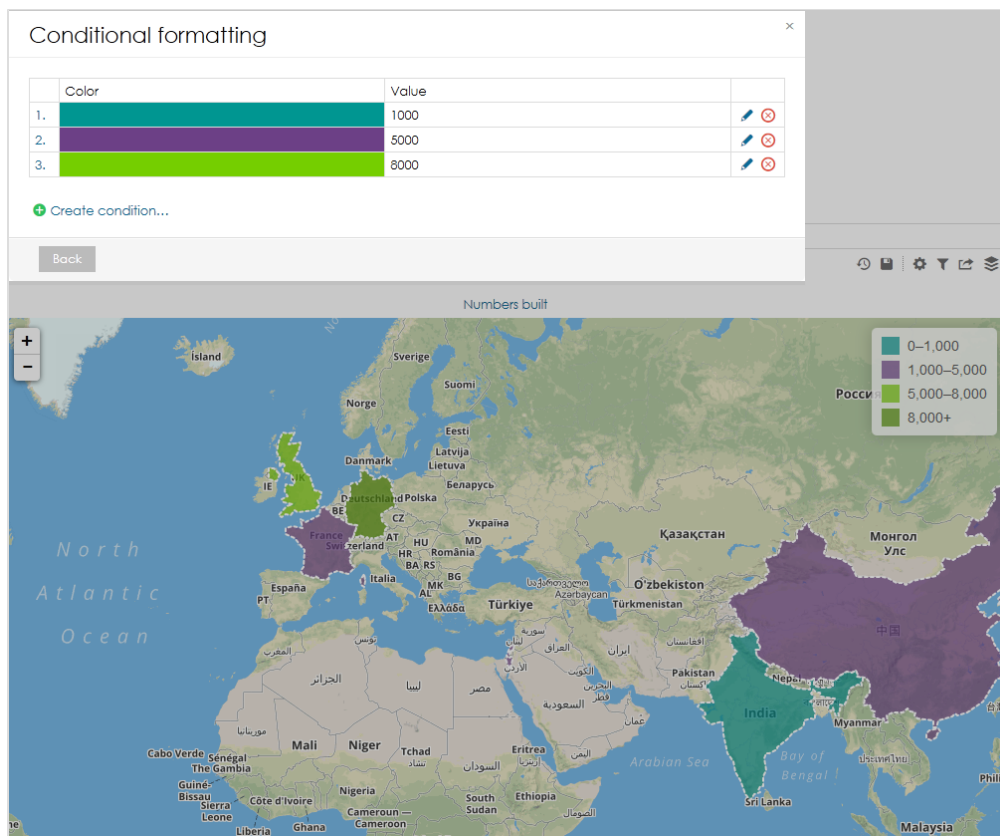
1. Formatting can be added by clicking on "*Create condition*"
2. Select desired colour.
3. Type value that will serve as top limit. Formulas can be used. That way you can easily specify quarters etc.
4. Select style - line or area.
5. Confirm by clicking on "Add".

It is possible to add multiple conditions. This way you can create beautiful scales.



Map conditional formatting

Conditional formatting in map is primarily used for creating custom color scale. The procedure is the same as in charts.



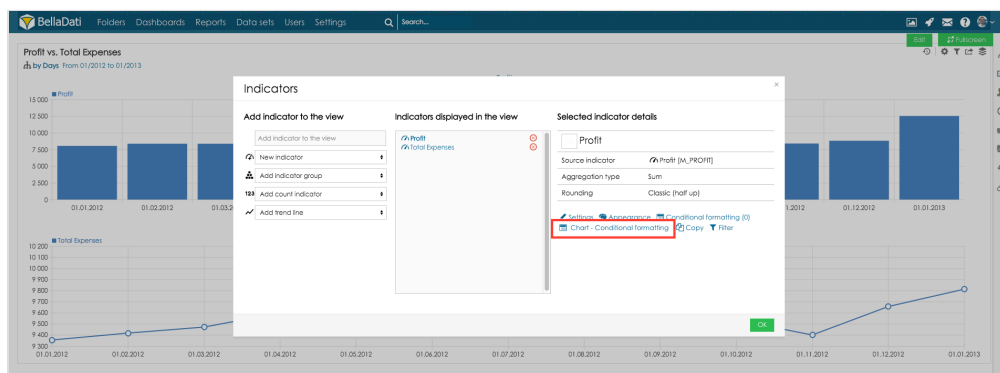
View with multiple indicators conditional formatting

In some cases, you might like to define different conditional formatting for the views which contain multiple indicators.

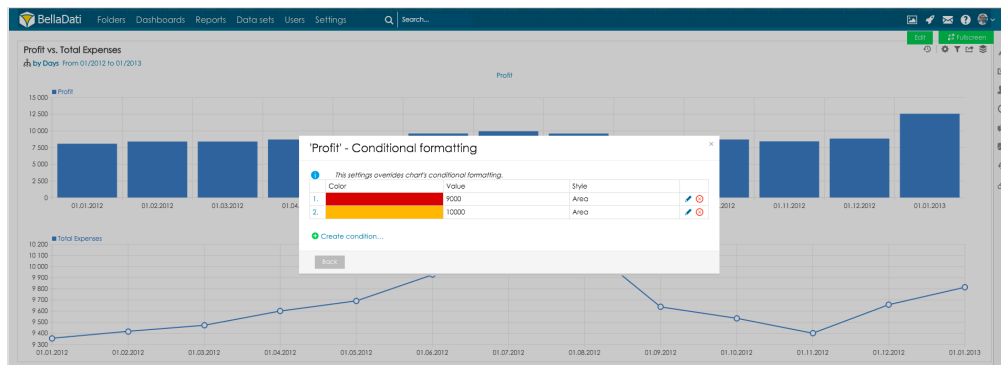
This option is available only if option Display in single chart is not selected and more than 1 indicator is added.

This feature is not available in Pie chart, Horizontal Stack Bar chart, Radial chart, Heatmap and Funnel chart.

This feature is accessible from **indicator settings**, select indicator and go to **Chart - conditional formatting**:



Now set conditional formatting in the same way as in chart conditional formatting:



This setting overrides chart's conditional formatting.

Sharing Report



Data set sharing functions are only available for the owners of the particular report.

Report sharing functions allows you to perform following actions:

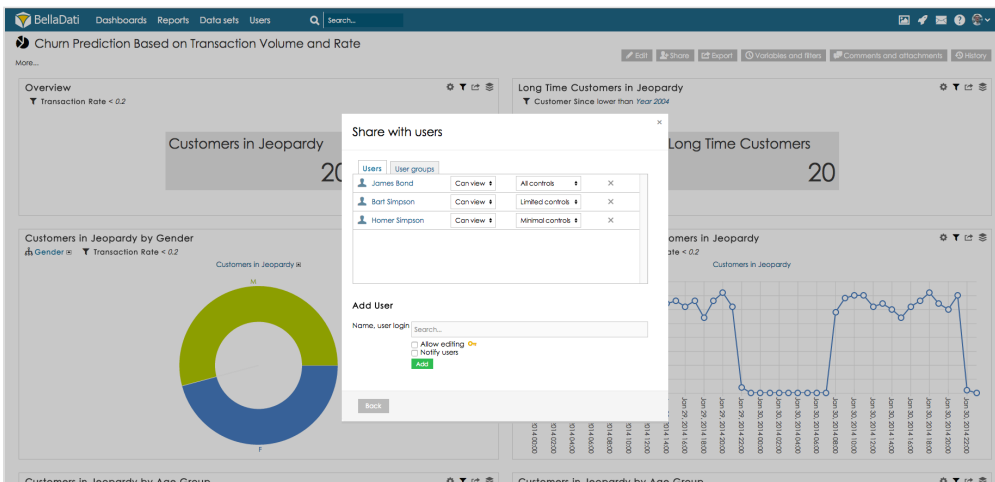
- Grant access to the report for selected [users](#) or [user groups](#).
- Optionally notify users about granted access to reports.

When you are setting up report sharing for individual users or user groups, please distinguish following two [access levels](#):

- **Read-only access:** Reports can be only viewed. Basic operations such as drill-down, exporting and report variable modifications are allowed within the user's session without affecting the original report. There are 3 different control types with different permissions allowed for the read-only access. See [the list](#) of allowed operations for selected control type.
- **Full access:** All functions except sharing and report deleting are available.



Users with full access can manage the report in the same way as it's owner except sharing settings or complete removal.

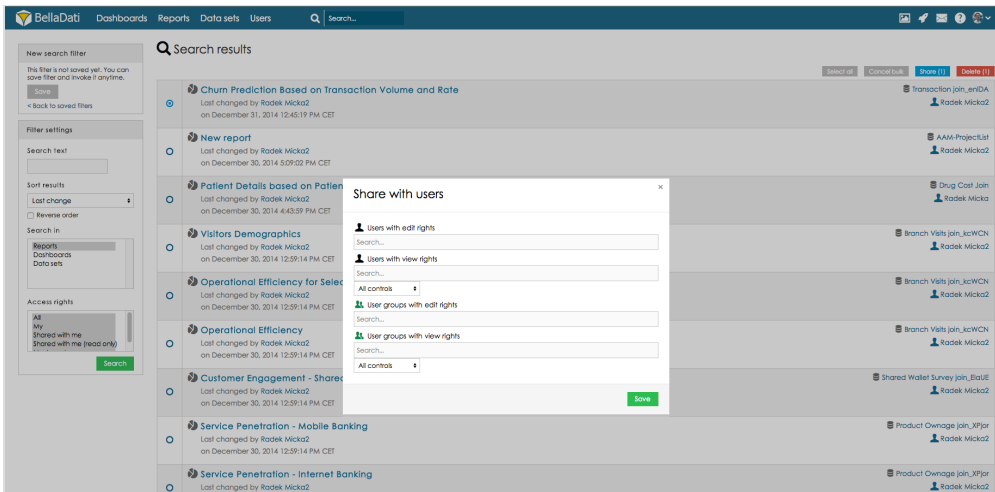


Bulk sharing of the reports

BellaDati also allows you to share multiple reports with users and user groups.

You can find **bulk sharing** in the **Reports list**. To share multiple reports:

1. Click **Bulk change** in upper right corner
2. **Select** desired reports to share
3. Click button **Share**
4. Select **users** and **user groups** who will have access to chosen reports



Sharing report groups

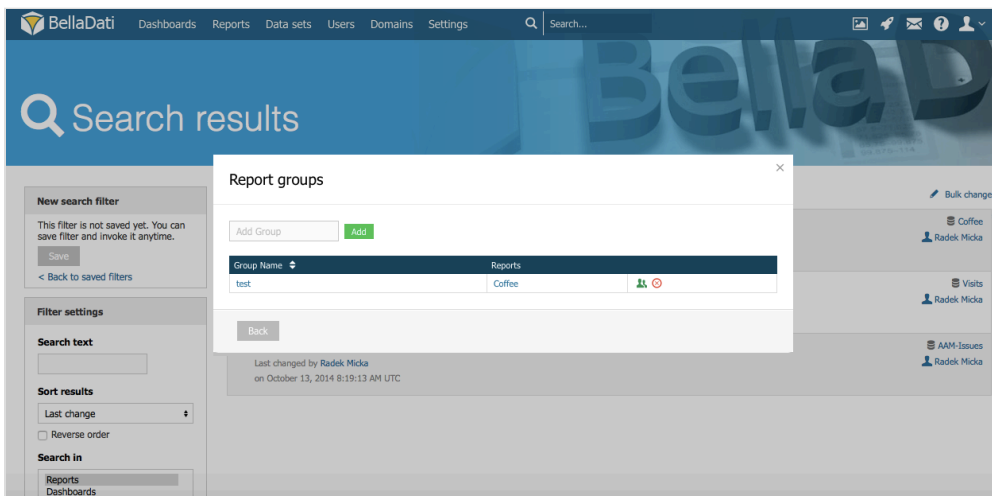
BellaDati also allows you to create report groups which can simply used for sharing of the reports with users or user groups.



Only report owner is allowed to add report to the report groups.

Point to the **Reports** in main menu on the top of the screen and click "Report groups" item which will appear.

1. Enter a name of the user group and click on button Add.
2. Click on Group Name and choose which report will be added to the report group.
3. Click on Users sign to set the users or user groups who will the report group be shared with.



Sharing console

BellaDati also allows you to see the list of reports with users and user groups the report is shared with. The list contains separate columns for report owner, report editors, read-only users, group editors, read only groups.

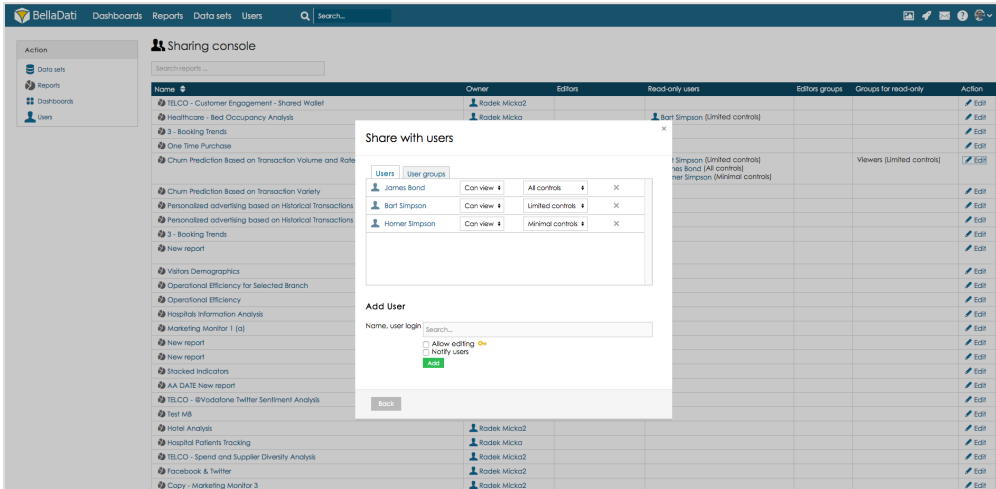


Only **Domain administrator** is allowed to access the Sharing console.

You can find Sharing console in the Users section. To access Sharing console:

1. Point to the Users in the main menu and click the Sharing console item.
2. Choose Reports in the Actions box

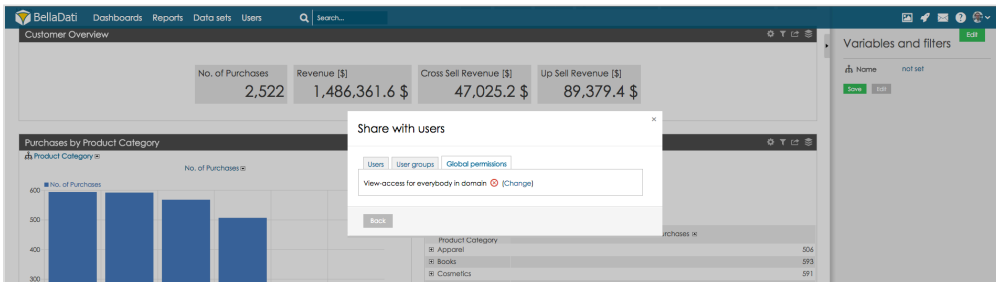
To change the sharing settings click button Edit for selected report.



Global sharing


Report can be also shared with all the users in the domain. Go to Sharing console, choose tab Global permissions and change the selection so the green icon is displayed.

This report is now available for all the users in the domain (view access).




Report alias

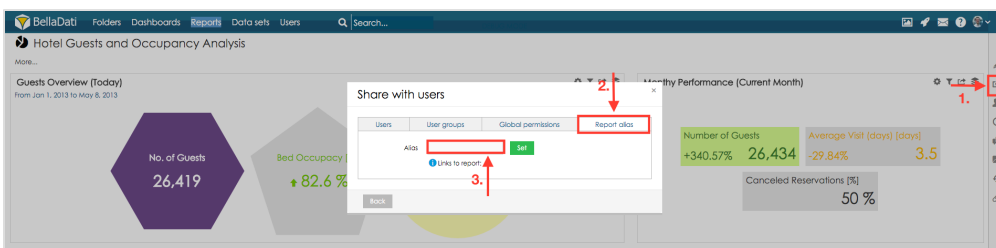
As advanced way to share the report, you can use the option to share your dashboard using an alias. By using this alias, user will be able to access the report.

 Report needs to be shared with user (or user group) who is accessing it.

In order to create report alias:

1. Click menu button Share
2. Select tab Report alias
3. Set report Alias

 Alias needs to be unique in the domain



To access the report use newly displayed link (relative path is `/bi/reports/alias`)

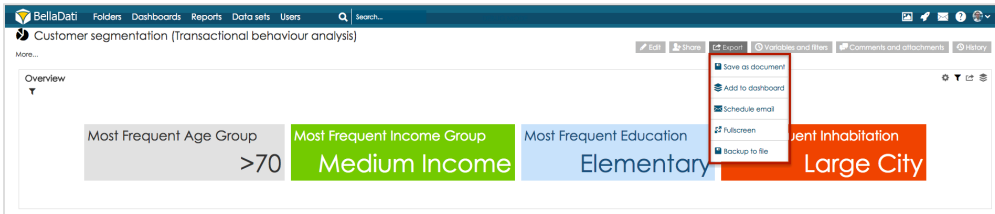
Exporting Report



You need to be in [view mode](#) in order to export report.

Export allows you to **Save report**, **Publish** it to dashboard or schedule **Emailing**.

Click **Export** or select **Save as document** from report toolbox list in the upper right corner to open *Export report* dialog.



Export report dialog allows you to save report in following formats:

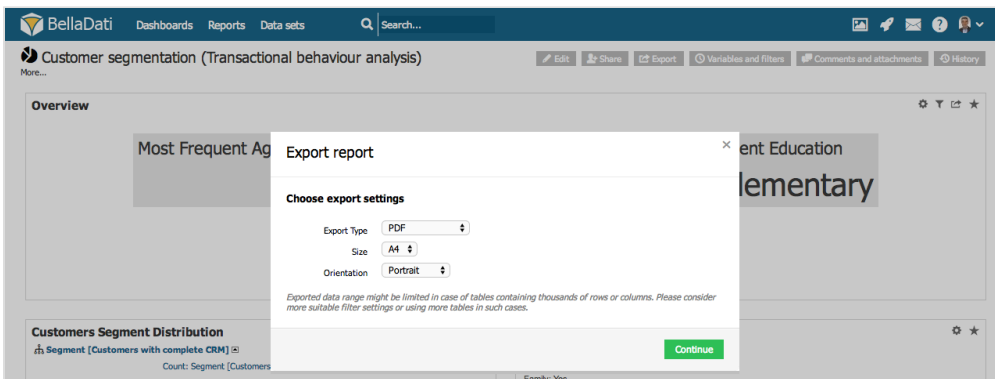
- **PDF:** You can specify **Orientation** and **Size** option.
- **Microsoft PowerPoint**
- **Microsoft Excel**



Geo map views will not be currently exported. Export to Microsoft Excel will include only **Table** views.



Maximum table rows count in the exports is currently limited to 1000 due to performance reasons. This limit can be raised for BellaDati Enterprise tariffs or licenses.

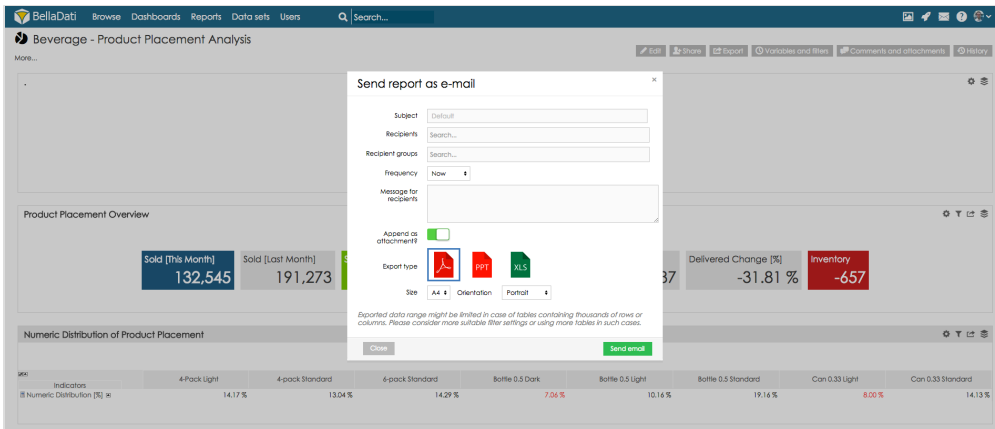


Publishing to Dashboard

For pinning report/view to dashboard - continue by [Publishing to Dashboard](#).

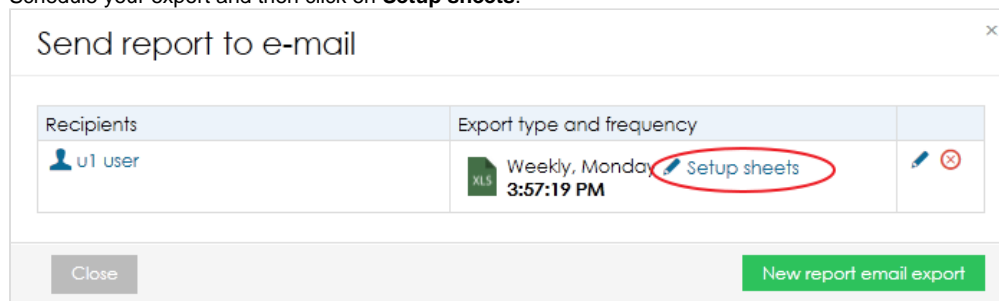
Emailing report

Select **Schedule email** from report toolbox list to open *Send report as e-mail* dialog.

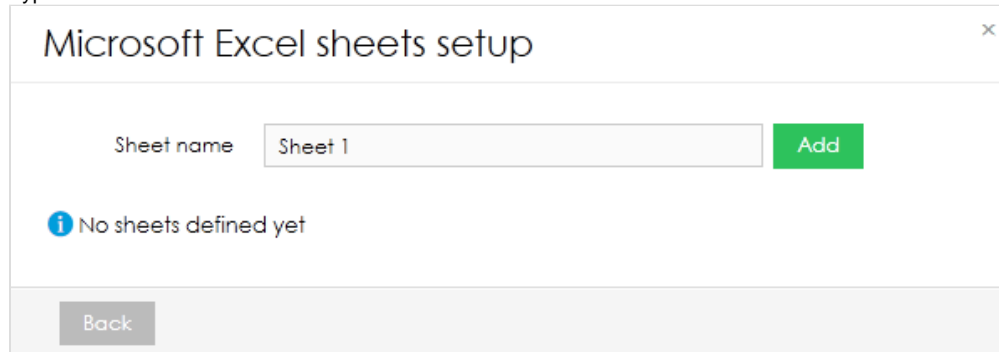


Dialog allows you to:

- Create **Subject**.
- Select Views that will be displayed in email body. Attachment will still contain all views. In the email body will be used ordering of selected views.
- Add **Recipients**.
- Add **Recipients groups**.
- Set up **Frequency** of delivery.
- Append **Message**.
- Select if report should be attached as attachment (views will be included in email body and attachment).
- Select if report **will be sent to all recipients at once** (other recipients will be put on cc).
- Select **Export type, Size and Orientation**.
 - if **XLS** is selected and email is **scheduled** (not sent immediately), it is possible to specify content of Excel sheets.
 1. Schedule your export and then click on **Setup sheets**.



2. Type **Sheet name** and click on **Add**.



3. Select views that you want to include on this sheet.

Report views on sheet

Sheet name

Sheet 1

Save

View


Table 1

Add

View	Action
Table 1	

Back

4. Repeat for all sheets.


 You can also send multiple reports via one email. For more info proceed to [Export schedule](#).

Schedule report export to FTP

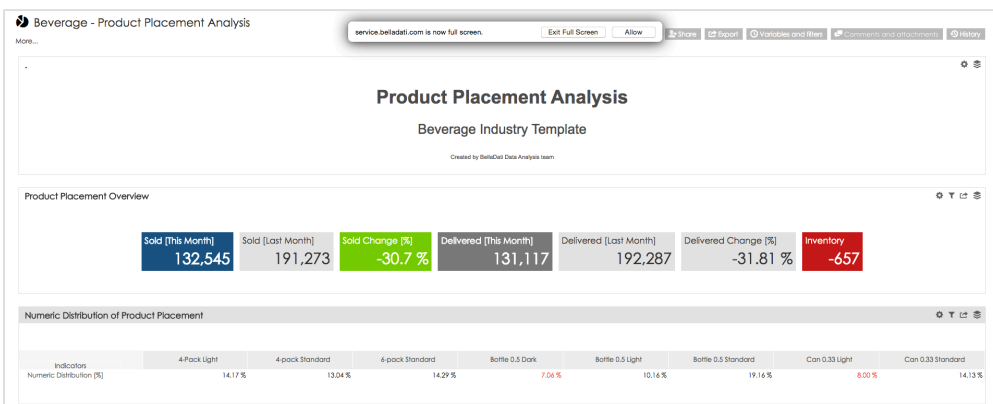
For scheduling report export to FTP - continue by [Schedule report export](#).

Fullscreen mode

Choose option *Fullscreen* from report toolbox list to enter fullscreen mode, which will display only the report itself. This mode should be used mainly for presentation of reports.

 Fullscreen mode is available only in following browsers:

- Firefox 10 and newer versions
- Chrome 15 and newer versions
- Opera 12.1 and newer versions
- Internet Explorer 11




Export to XML

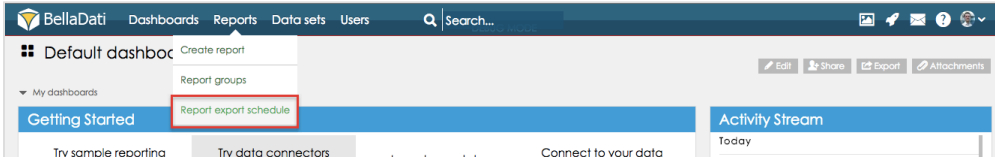
Click *Backup to file* to create report XML backup. XML file will be promptly downloaded to your hard drive. If you would like to reimport other version of report go to [data set summary](#).

Export schedule

The report export page serves to schedule exports to FTP, dashboard email export or report email export. You can also manage all of these reports.

 This page is available only for [domain administrator](#).

The report export page is accessible in the top menu of BellaDati. Point your mouse to the option **Report** and click on displayed link Report export schedule.

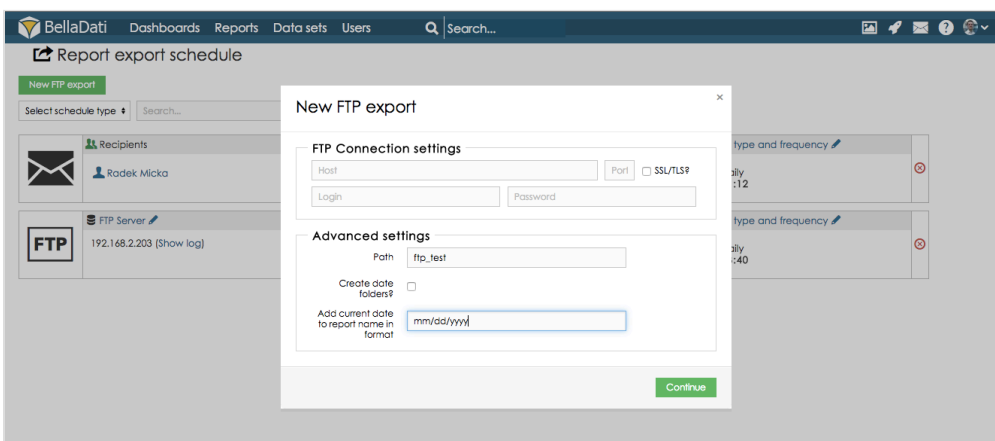


Schedule FTP export

Click button **New FTP export** to create new FTP connection.

FTP Connection settings

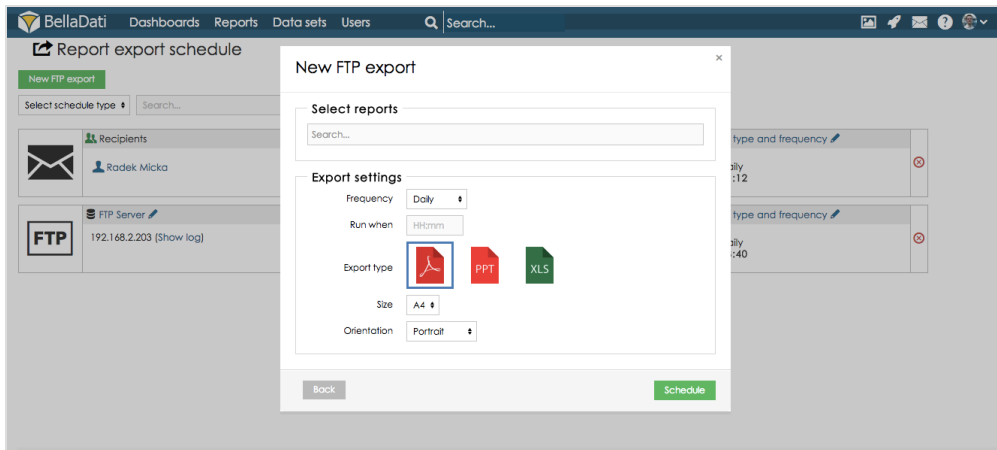
Host	FTP server address
Port	Used port of FTP server <i>(optional)</i>
SSL/TLS?	Enables FTPS/FTPES mode <i>(optional)</i>
Login	FTP login
Password	FTP password
Path	Path where the report will be saved
Create date folders?	Creates new folder for export date (if selected)
Add current date to report name in format	Create date pattern by which the report will be saved



Proceed to next settings by clicking button **Continue**.

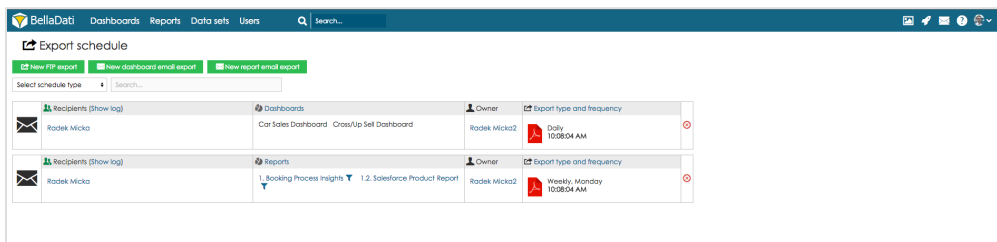
This dialog allows you to:

- Select reports which will be exported
- Select frequency how often the report will be saved
- Choose time when export will run
- Select export type
- Select size and orientation (available only for PDF export)



Click button **Schedule** to save the settings of this export.

Created export is displayed in the table where can be changed export settings and selected reports for export (both FTP and email exports are displayed).



Choose if only FTP or email exports should be displayed by *Select schedule type* drop down or browse scheduled exports by *Search* field.

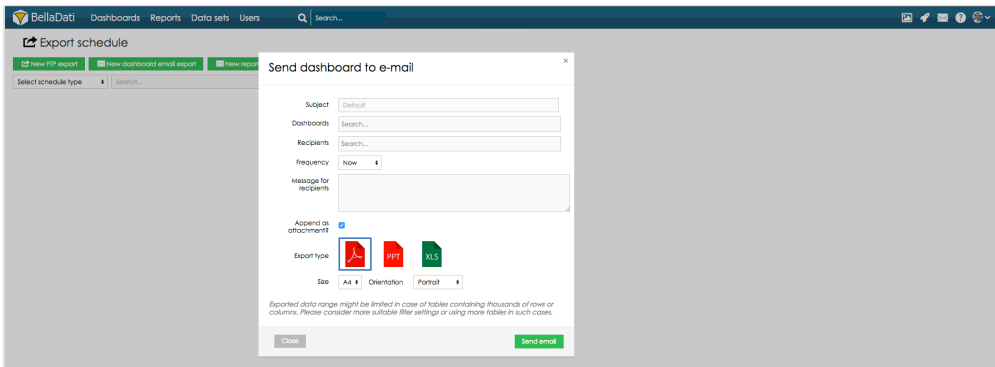
Send dashboard to email




This feature allows you to send multiple dashboards in one email.

Click button *New dashboard email export* to open dialog window with settings of sent email. In this window, you are allowed to set:

- **Subject** of the email
- Select **Dashboards** which will be sent (if no dashboards are selected, email is sent without details of the dashboard)
- Select **Recipients** to whom the dashboard will be sent
- Set **Frequency** when email will be sent
- Add **Message for recipients**
- Choose **Export type** - type of file to which the dashboard will be exported

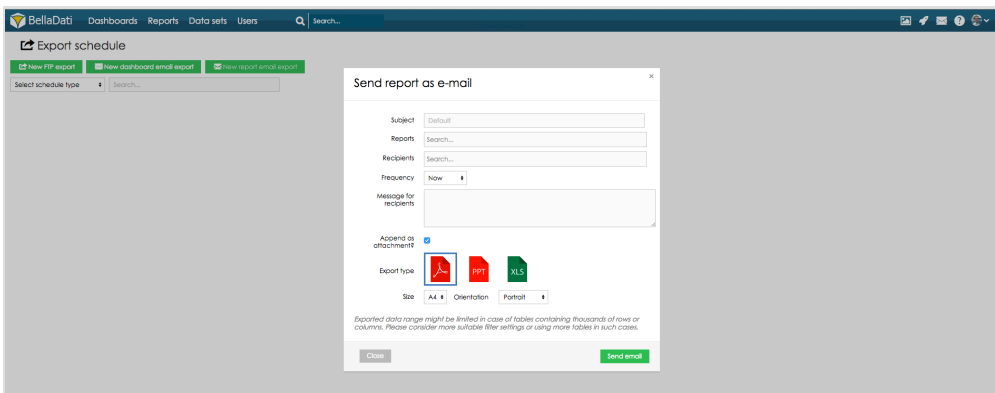


Send report to email

 This feature allows you to send multiple reports in one email.

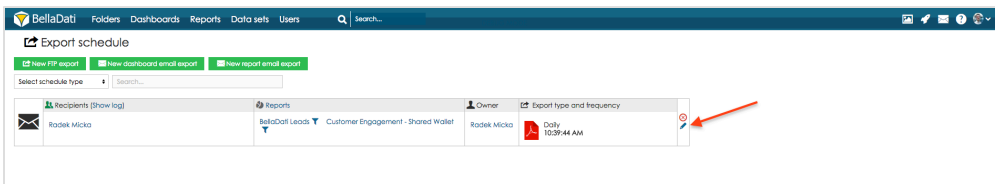
Click button *New report email export* to open dialog window with settings of sent email. In this window, you are allowed to set:

- **Subject** of the email
- Select **Reports** which will be sent (if no reports are selected, email is sent without details of the report)
- Select **Recipients** and **Recipient groups** to whom the report will be sent
- Set **Frequency** when email will be sent
- Add **Message for recipients**
- Choose **Export type** - type of file to which the report will be exported



Edit export schedule

If you would like to change settings of scheduled export hit a click on "pencil" icon displayed on the list of scheduled exports.

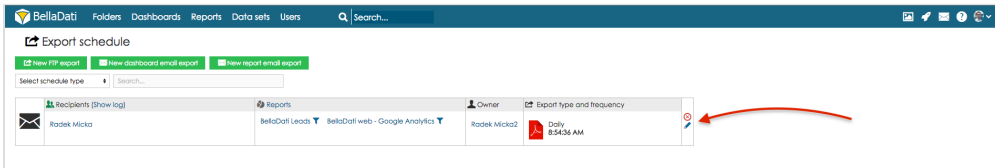



Displayed dialogue window allows you to:

- Add/remove **Reports** which will be exported
- Change ordering of **Reports** which will be exported
- Select **Views** and **Ordering of the views** displayed in email body (click on the name of the report to open dialogue window with list of the reports, which allows you to select views which should be added/removed, or change position of views)
- Add/edit **Recipients** and **Recipient groups** who will receive email
- Set **Frequency** by which selected email will be sent
- Add/edit **Message for recipients**
- Set **Export type**
- Adjust size of the views in the email body

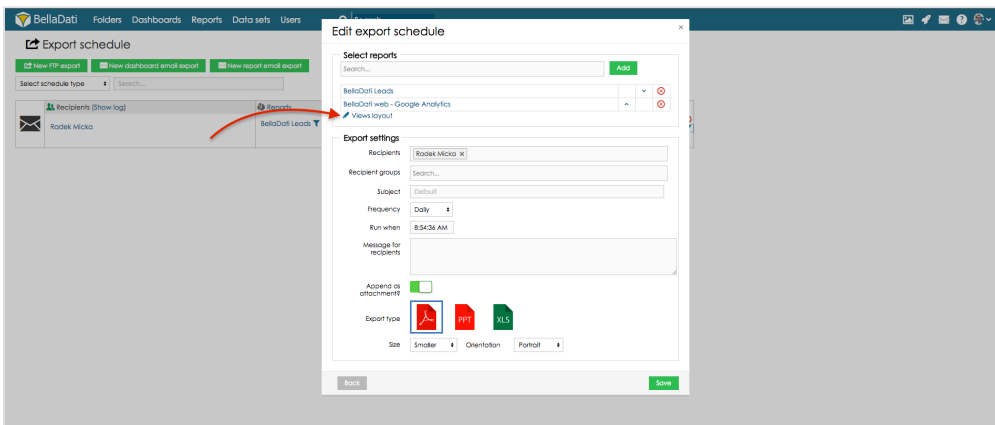
Adjusting size of the views in the email body

To adjust size of the views in email body, go to [Export schedule](#) and select option Edit.

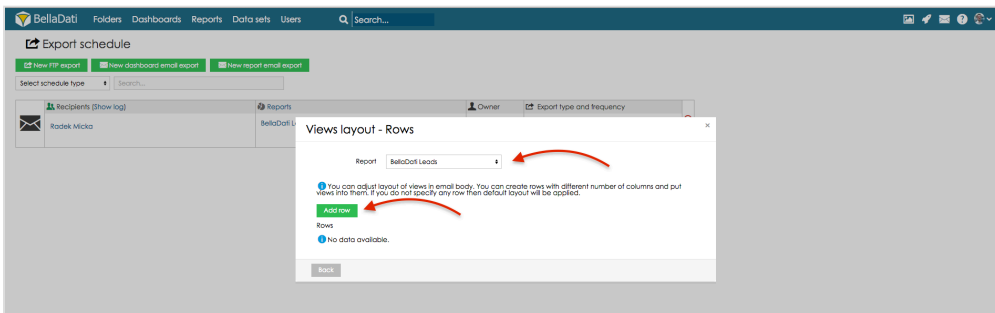


 Only size of the charts can be adjusted.

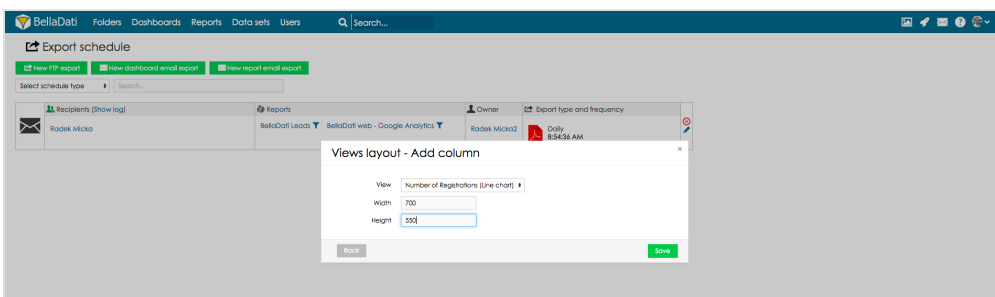
Go to views layout:



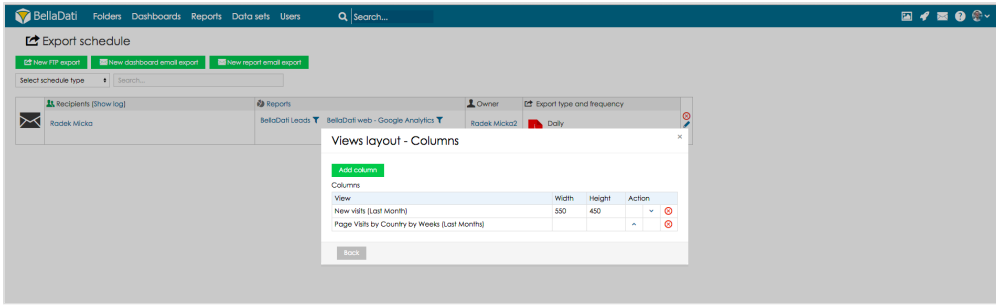
Select report you want to set up and click button **Add row**.




Click the button **Add column** to open **Views layout**. Select view which should be adjusted and set the **width** and **height** and click **Save**:

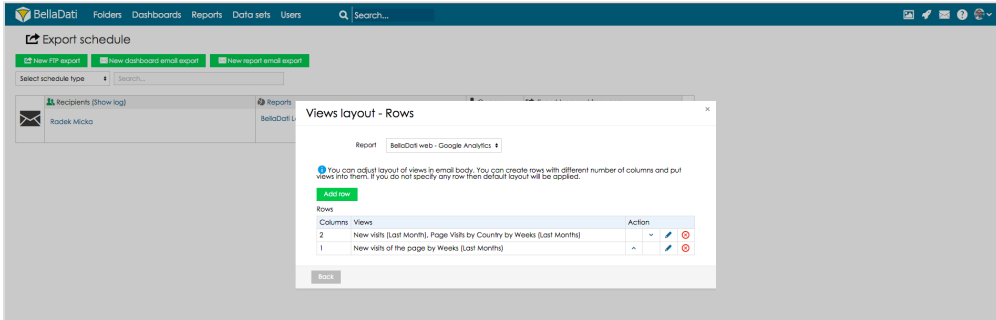



If there is added more columns into one row, visualisations will be added into one row, next to each other.



 Position of columns can be changed by using arrows in section Action.

If you need to manage position of the rows, go *Back* to the rows settings.



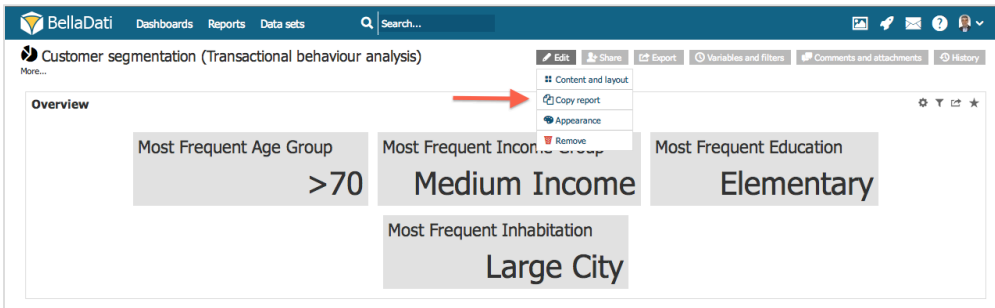
 Position of rows can be changed by using arrows in section Action.

Copying Report



You need to be in [view mode](#) in order to copy report.

Select **Copy report** from report toolbox list in the upper right corner to open *Copy report* dialog.

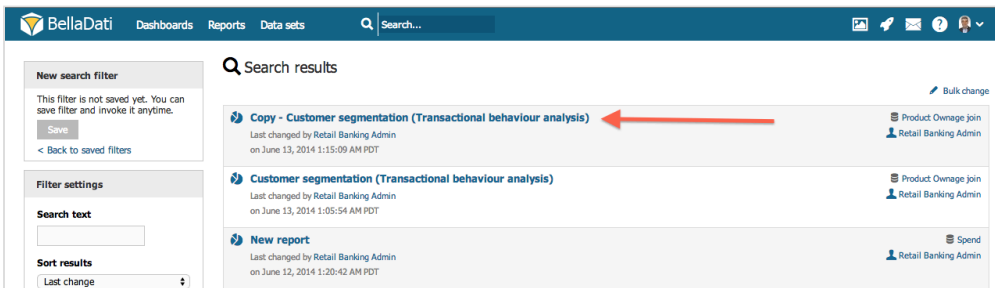


Copy report dialog allows you to include in the copy of the report:

- **Comments**
- **Attachments**



Copied report can be found in **Reports** window and will be prefixed with *Copy -*.



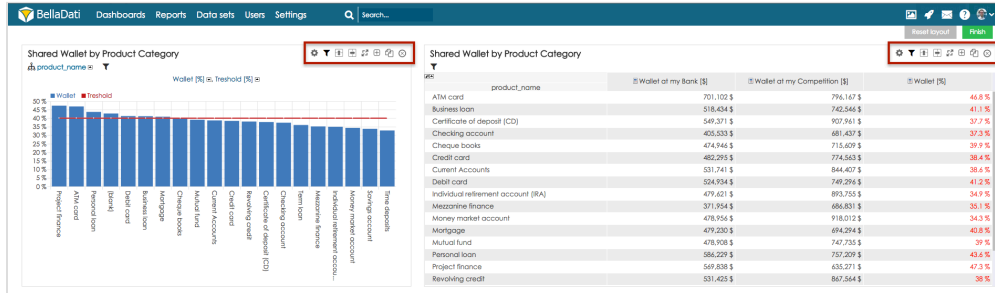
Managing Layout



You need to be in **edit mode** in order to manage layout.

- Click on "Edit" in top report menu to activate edit mode.
- Click on the green "Finish" button on the top of right column with templates list to save changes and exit edit mode.

Use **buttons** in the upper right corner of the view to manipulate with it.



Edit mode

You can perform the following actions when being in edit mode:

- **Split vertically:** Divides current view into two rows. Original view will be kept in the upper part.
- **Split horizontally:** Divides current view into two columns. Original view will be kept in the left part.
- **Enlarge to the whole view:** View will be extended across the whole report.
- **Insert new row:** New row will be added above the current view.
- **Copy:** Click on the desired area to make the view copy.
- **Delete:** Report will be deleted.



The user needs report editor role or editing permission to manage report layout.

Moving views

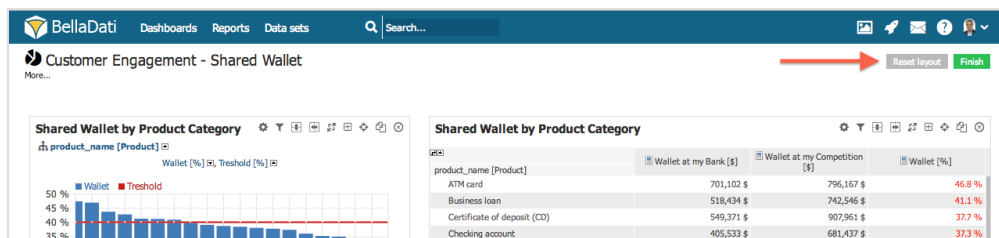
Position of the views is managed by drag and drop. Just drag the view and drop it above another view. Positions of these two views are switched.

Reset layout

Click on **Reset layout** button in upper right corner to place views into their default positions.



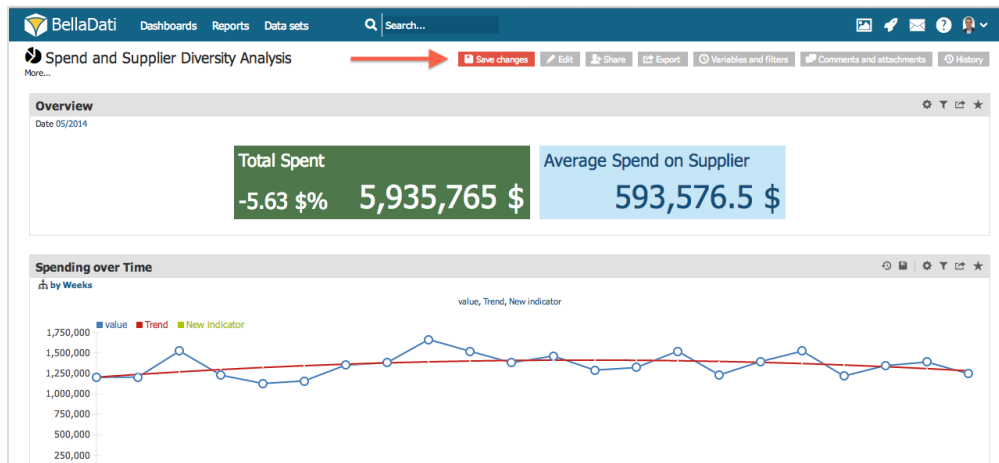
Layout reset will remove all empty view.



Saving Reports

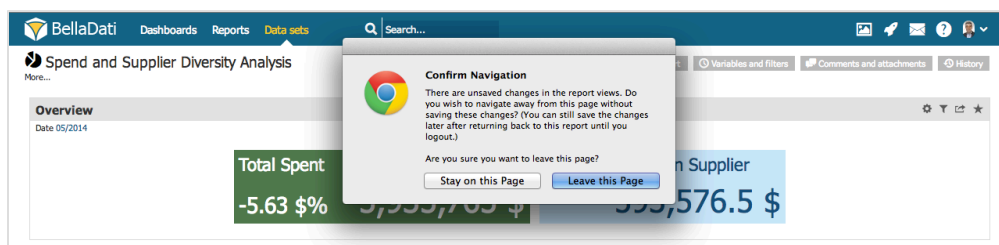
Saving Reports

You can save changes of all views in the report by going to **Edit** in upper report menu and selecting **Save Changes**.



Leaving Unsaved Report

BellaDati will notify you with popup window when leaving report with unsaved views.



Using Formulas

Formulas are used to create **Calculated** (derived) **Indicators** from basic indicators in BellaDati.

There are four types of indicators defined by formula:

- Formula indicators defined on data set level. These are available in all reports based on this data set.
- Formula indicators defined ad hoc on view level in each report. These are available only for the particular view and belong to the two subcategories:
 - Additional formula defined indicators
 - Formulas on date/time axis

Creating Formulas



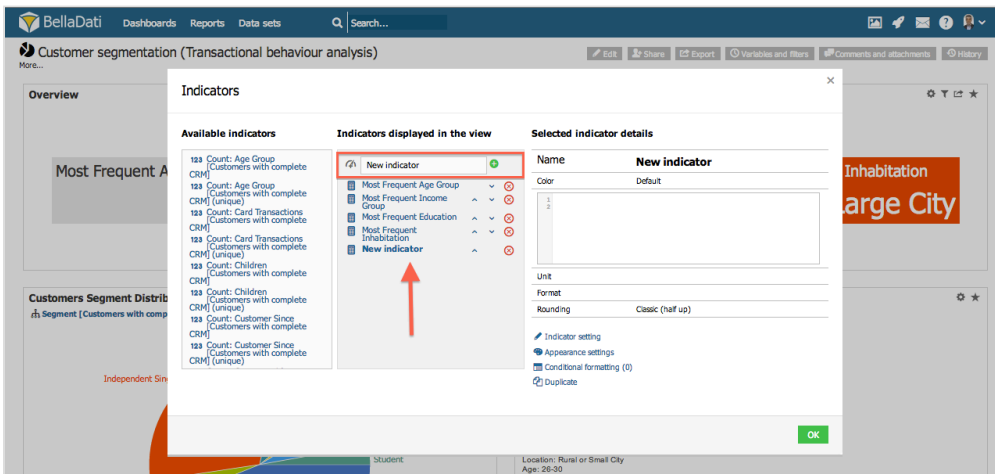
Make sure that you are familiar with [Displaying Indicators](#) section prior proceeding with **Formulas**.

You can edit **Formulas** only of **Calculated Indicators**. To create **Calculated Indicator**:

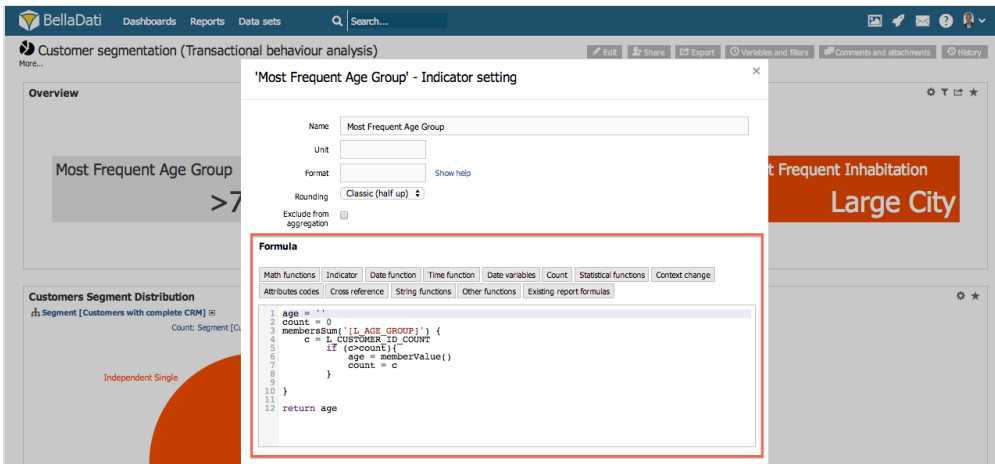
1. Go to **Indicators Settings** dialog.
2. Provide name and click on green plus button next to **New Indicator** input.
3. Click on **Indicator Settings**.
4. Create/edit formula in **Formula** window.



Calculated indicators can be determined by **calculator** symbol next to indicator name.



You can leverage **autocomplete** or **lists** of available functions for rapid formula development.



See [Formula Reference Guide](#) for complete specification of available formulas.

Editing Formulas

Click on Indicator name in the report to open *Indicator settings* window.

Aggregation in Calculated Indicators

Notice that *Indicators setting* dialog of **Calculated Indicator** lacks specification of **Members** and **Time aggregation**.

This is because you can define it programmatically in combination with additional functions.

Member aggregation can be defined by suffixing **Indicator** with:

- @SUM for aggregation
- @AVG for average
- @MIN for minimum
- @MAX for maximum
- @DC for distinct count



Example: use M_SALES@SUM to obtain Total Sales or M_PRICE@MIN to find the lowest price.

Visit [Formula Reference Guide](#) to learn more about aggregations in formulas.

Formula Reference Guide



This summary provides overview of all formulas that can be used in reports or data sets (predefined indicators). If you are searching for **transformation scripting** during data import, see [Developer documentation](#).

Indicators and codes

Each dataset indicator is specified by its unique code starting with `M_` (M as Measure). Accessing the calculated indicator's value is possible by typing this code directly into the formula. For example:

Another way how to get the value of the indicator is to use the `value()` function:



Strings must be always enclosed by apostrophes: `'L_NAME'`.

Both examples have the same result.

Drill-down (members) aggregation

Members aggregation determines the way how to count values in the case that exists more records for one selected member in a single time unit. Aggregation type is specified by adding the appropriate suffix to the indicator's code. When not specified, the `SUM` aggregation is applied.

Suffix	Description	Example
@SUM	Calculates the sum of all values for the selected drill-down attribute	<code>M_NAME@SUM</code>
@MIN	Calculates the minimum of all values for the selected drill-down attribute	<code>M_NAME@MIN</code>
@MAX	Calculates the maximum of all values for the selected drill-down attribute	<code>M_NAME@MAX</code>
@AVG	Calculates the average of all values for the selected drill-down attribute	<code>M_NAME@AVG</code>
@DC	Calculates the distinct count of all values for the selected drill-down attribute	<code>M_NAME@DC</code>

Counting of level members

Each drill-down level is represented by particular members, for example the level `City` contains members like `Berlin`, `Paris`, `New York` etc. To get the count of these members, use the following syntax:

Datetime Functions

Datetime Functions



For the purpose of this reference guide: **Date** refers only to year, months, days (and quarters, weeks) and their combinations. **Time** refers to hours, minutes and seconds and their combinations. For combination of date and time we strictly use **datetime** term.



See the [Date](#) and [time](#) functions inherited from transformation scripting.

Date strings

The date string parameters are entered absolutely (dd.MM.yyyy, yyyy-MM-dd or according to domain settings) or relatively (time variables) by operators:

`date +/- n[d|w|m|q|y]`

where

- `date` is date in dd.MM.yyyy or yyyy-MM-dd format, or one of `actualyear`, `actulamonth`, `actualquarter`, `actualweek`, `actualday`, `now`
- `n` represents the count of:
- `d` days, `w` weeks, `m` months, `q` quarters or `y` year.

Examples:

Another way how to create the date strings is following:

Changing datetime context

What is the datetime context? Consider following example:

	01/2011	02/2011	03/2011
formula indicator	1000	1200	1300

Formula is evaluated for each column - in this example, in columns are values evaluated for particular months. During the evaluation of value 1000, the datetime context was 01/2011, then during the processing of value 1200, the context was 02/2011 etc.

Function	Description
<code>Object dateAt(String dateString, { expression })</code>	Changes the context of the evaluated expression to <code>dateString</code> date. Example:
<code>Object dateInterval(String from, String to, { expression })</code>	Changes the date context of the expression and evaluates it aggregated in the specified interval <code>from</code> - <code>to</code> . Example Values for the indicators <code>M_NAME_1</code> and <code>M_NAME_2</code> are aggregated for the whole period.
<code>Object timeAt(String timeString, { expression })</code>	Changes the context of the evaluated expression to <code>timeString</code> date. Example:

Object timeInterval(String from, String to, { expression })	Changes the time context of the expression and evaluates it aggregated in the specified interval <i>from - to</i> . Example Values for the indicators M_NAME_1 and M_NAME_2 are aggregated for the whole period.
Object dateAt (String date, String period) { expression })	Changes the context of the evaluated expression to <i>dateString date</i> - aggregated by the defined period {DAY, D, }
Object dateInterval (String from, String to, String period) { expression })	Changes the date context of the expression and evaluates it aggregated in the specified interval <i>from - to</i> . Aggrega WEEK, W, MONTH, M, YEAR, Y} is also performed.
Object minus(String date, period, int)	Subtracts the amount of specified <i>period</i> (years, months, weeks, days) to the defined date value.
Object minus(String time, period, int)	Adds the amount of specified <i>period</i> (years, months, weeks, days) to the defined date value.
Object plus(String date, period, int)	Adds the amount of specified <i>period</i> (years, months, weeks, days) to the defined date value.
Object plus(String time, period, int)	Adds the amount of specified <i>period</i> (hours, minutes, seconds) to the defined date value.
Object timeAt (String time, String period) { expression })	Changes the context of the evaluated expression to <i>timeString time</i> - aggregated by the defined period {HOUR, H
Object timeInterval (String from, String to, period) { expression })	Changes the time context of the expression and evaluates it aggregated in the specified interval <i>from - to</i> . Aggrega MINUTE, SECOND} is also performed.
Object withoutDateTime() { expression })	Evaluates the expression without date time interval.

Obsolete functions



These functions may be removed in further releases

Function	Description
Object value(String dateString, String indicator)	Loads the indicator's value at specified date.
Object value(String dateString, int drill_down_level)	Loads the indicator's value at specified date aggregated for N previous levels.

Object value(String dateFrom, String dateTo, String indicator)	Loads cumulative indicator's value for specified date interval.
Object value(String dateFrom, String dateTo, String indicator, int drill_down_level)	Loads cumulative indicator's value for specified date interval aggregated for N previous levels.

Advanced functions


Function	Description						
Number cumulateFromDate(String startDate, String indicator)	<p>This function gradually adds the current value to the cumulated value. Example:</p> <table><tr><td></td></tr><tr><td>M_NAME_1</td></tr><tr><td>cumulateFromDate('2011-01-01', 'M_NAME_1')</td></tr></table>		M_NAME_1	cumulateFromDate('2011-01-01', 'M_NAME_1')			
M_NAME_1							
cumulateFromDate('2011-01-01', 'M_NAME_1')							
Number cumulateFromTime(String startTime, String indicator)	<p>This function gradually adds the current value to the cumulated value. Example:</p> <table><tr><td></td></tr><tr><td>M_NAME_1</td></tr><tr><td>cumulateFromTime('00:01', 'M_NAME_1')</td></tr></table>		M_NAME_1	cumulateFromTime('00:01', 'M_NAME_1')			
M_NAME_1							
cumulateFromTime('00:01', 'M_NAME_1')							
prev(String indicatorCode)	<p>Value of the passed indicator calculated for previous date or time value (e.g. previous month).</p> <table><tr><td></td><td>01/2011</td></tr><tr><td>M_NAME_1</td><td>1000</td></tr><tr><td>prev('M_NAME_1')</td><td></td></tr></table>		01/2011	M_NAME_1	1000	prev('M_NAME_1')	
	01/2011						
M_NAME_1	1000						
prev('M_NAME_1')							
Number prev(String indicatorCode, int prevLevelAgg)	<p>Value of the passed indicator calculated for previous date or time value (e.g. previous month) and its number of previous levels which should be aggregated.</p>						
Number next(String indicatorCode)	<p>Value of the passed indicator calculated for next date or time value (e.g. previous month).</p> <table><tr><td></td><td>01/2011</td></tr><tr><td>M_NAME_1</td><td>1000</td></tr><tr><td>next('M_NAME_1')</td><td>1200</td></tr></table>		01/2011	M_NAME_1	1000	next('M_NAME_1')	1200
	01/2011						
M_NAME_1	1000						
next('M_NAME_1')	1200						
Number next(String indicatorCode, int prevLevelAgg)	<p>Value of the passed indicator calculated for next date or time value (e.g. previous month) and number of previous levels which should be aggregated.</p>						
int daysBetween(DateTime dateFrom, DateTime dateTo)	<p>Function calculates number of days between provided dates.</p>						
int daysBetween(LocalDate dateFrom, LocalDate dateTo)	<p>Function calculates number of days between provided dates.</p>						
int monthsBetween(DateTime dateFrom, DateTime dateTo)	<p>Function calculates number of months between provided dates.</p>						
int monthsBetween(LocalDate dateFrom, LocalDate dateTo)	<p>Function calculates number of months between provided dates.</p>						

<code>int yearsBetween(DateTime dateFrom, DateTime dateTo)</code>	Function calculates number of years between provided dates.
<code>int yearsBetween(LocalDate dateFrom, LocalDate dateTo)</code>	Function calculates number of years between provided dates.

For changing the whole context of the evaluated expression, you can use following functions:

Number <code>prev(String period, { expression })</code>	Changes the context of the expression to desired previous date period. The value of the period can be YEAR, MONTH, DAY, HOUR, MINUTE, SECOND. Example: <table border="1"> <tr> <td></td><td>01/2011</td></tr> <tr> <td>M_NAME_1</td><td>1000</td></tr> <tr> <td>prev(MONTH) { M_NAME_1 }</td><td></td></tr> </table>		01/2011	M_NAME_1	1000	prev(MONTH) { M_NAME_1 }	
	01/2011						
M_NAME_1	1000						
prev(MONTH) { M_NAME_1 }							
Number <code>next(String period, { expression })</code>	Changes the context of the expression to desired next date period. The value of the period can be YEAR, MONTH, DAY, HOUR, MINUTE, SECOND.						

The following example works with values, which are loaded for one year before the actual table/chart datetime entry:


 You may simply enter only the first letter of time unit (in case of time context changing formulas) instead of their full names (Y,Q,M,W,D), for example prev(Y){}

Nested expressions

Date and time functions can be combined (if applicable), eg.: Value of M_INDICATOR at 8th December 2010, 9:04:02AM:

Datetime Constants

Datetime Constants

Accessing context datetime

Function	Description
<code>String contextDay()</code>	Returns the formula's context day dd.MM.yyyy format (or domain format).
<code>String contextWeek()</code>	Returns the beginning of the context's week in dd.MM.yyyy format (or domain format).
<code>String contextMonth()</code>	Returns the beginning of the context's month in dd.MM.yyyy format (or domain format).
<code>String contextQuarter()</code>	Returns the beginning of the context's year in dd.MM.yyyy format (or domain format).
<code>String contextYear()</code>	Returns the formula's context week start in dd.MM.yyyy format (or domain format).
<code>int dateDayOfYear()</code>	Returns the day of year from the context date
<code>int dateDayOfMonth()</code>	Returns the day of month from the context date
<code>int dateDayOfWeek()</code>	Returns the day of week from the context date
<code>int dateMonth()</code>	Returns the number of month from the context date
<code>int dateYear()</code>	Returns the number of month from the context date
<code>int timeHour()</code>	Returns the number of hour from the context time
<code>int timeMinute()</code>	Returns the number of minute from the context time
<code>int timeSecond()</code>	Returns the number of second from the context time
<code>int daysInMonth()</code>	Returns the number of days in the context date

Examples:

	31.1.2011 00:01:00	1.2.2011 00:01:00	2.2.2011 00:01:00
<code>String contextDay()</code>	31.1.2011	1.2.2011	2.2.2011
<code>String contextWeek()</code>	31.1.2011	31.1.2011	31.1.2011
<code>String contextMonth()</code>	1.1.2011	1.2.2011	1.2.2011
<code>String contextQuarter()</code>	1.1.2011	1.1.2011	1.1.2011
<code>String contextYear()</code>	1.1.2011	1.1.2011	1.1.2011
<code>int dateDayOfYear()</code>	31	32	33
<code>int dateDayOfMonth()</code>	31	1	2
<code>int dateDayOfWeek()</code>	1	2	3
<code>int dateMonth()</code>	1	2	2
<code>int dateYear()</code>	2011	2011	2011
<code>int timeHour()</code>	0	0	0
<code>int timeMinute()</code>	1	1	1
<code>int timeSecond()</code>	0	0	0

Accessing actual date

Function	Description
<code>String actualDay()</code>	Returns the actual day in dd.MM.yyyy format
<code>String actualDate()</code>	Returns the actual day in dd.MM.yyyy format
<code>String actualWeek()</code>	Returns the actual week in dd.MM.yyyy format
<code>String actualMonth()</code>	Returns the actual month in dd.MM.yyyy format
<code>String actualQuarter()</code>	Returns the actual quarter in dd.MM.yyyy format
<code>String actualYear()</code>	Returns the actual year in dd.MM.yyyy format

Math Functions

Math Functions

Function	Description
<code>double abs(double a)</code>	Returns the absolute value of a double value.
<code>float abs(float a)</code>	Returns the absolute value of a float value.
<code>int abs(int a)</code>	Returns the absolute value of an int value.
<code>long abs(long a)</code>	Returns the absolute value of a long value.
<code>double acos(double a)</code>	Returns the arc cosine of a value; the returned angle is in the range 0.0 through pi.
<code>double asin(double a)</code>	Returns the arc sine of a value; the returned angle is in the range -pi/2 through pi/2.
<code>double atan(double a)</code>	Returns the arc tangent of a value; the returned angle is in the range -pi/2 through pi/2.
<code>double atan2(double y, double x)</code>	Returns the angle theta from the conversion of rectangular coordinates (x, y) to polar coordinates (r, theta).
<code>double cbrt(double a)</code>	Returns the cube root of a double value.
<code>double ceil(double a)</code>	Returns the smallest (closest to negative infinity) double value that is greater than or equal to the argument and is equal to a mathematical integer.
<code>double copySign(double magnitude, double sign)</code>	Returns the first floating-point argument with the sign of the second floating-point argument.
<code>float copySign(float magnitude, float sign)</code>	Returns the first floating-point argument with the sign of the second floating-point argument.
<code>double cos(double a)</code>	Returns the trigonometric cosine of an angle.
<code>double cosh(double x)</code>	Returns the hyperbolic cosine of a double value.
<code>double exp(double a)</code>	Returns Euler's number e raised to the power of a double value.
<code>double expm1(double x)</code>	Returns $e^x - 1$.
<code>double floor(double a)</code>	Returns the largest (closest to positive infinity) double value that is less than or equal to the argument and is equal to a mathematical integer.
<code>int getExponent(double d)</code>	Returns the unbiased exponent used in the representation of a double.
<code>int getExponent(float f)</code>	Returns the unbiased exponent used in the representation of a float.
<code>double hypot(double x, double y)</code>	Returns $\sqrt{x^2 + y^2}$ without intermediate overflow or underflow.
<code>double IEEERemainder(double f1, double f2)</code>	Computes the remainder operation on two arguments as prescribed by the IEEE 754 standard.
<code>double log(double a)</code>	Returns the natural logarithm (base e) of a double value.
<code>double log10(double a)</code>	Returns the base 10 logarithm of a double value.
<code>double log1p(double x)</code>	Returns the natural logarithm of the sum of the argument and 1.
<code>double max(double a, double b)</code>	Returns the greater of two double values.
<code>float max(float a, float b)</code>	Returns the greater of two float values.
<code>int max(int a, int b)</code>	Returns the greater of two int values.
<code>long max(long a, long b)</code>	Returns the greater of two long values.
<code>double min(double a, double b)</code>	Returns the smaller of two double values.
<code>float min(float a, float b)</code>	Returns the smaller of two float values.

<code>int min(int a, int b)</code>	Returns the smaller of two int values.
<code>long min(long a, long b)</code>	Returns the smaller of two long values.
<code>double nextAfter(double start, double dir)</code>	Returns the floating-point number adjacent to the first argument in the direction of the second argument.
<code>float nextAfter(float start, double dir)</code>	Returns the floating-point number adjacent to the first argument in the direction of the second argument.
<code>double nextUp(double d)</code>	Returns the floating-point value adjacent to d in the direction of positive infinity.
<code>float nextUp(float f)</code>	Returns the floating-point value adjacent to f in the direction of positive infinity.
<code>double pow(double a, double b)</code>	Returns the value of the first argument raised to the power of the second argument.
<code>double random()</code>	Returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0.
<code>double rint(double a)</code>	Returns the double value that is closest in value to the argument and is equal to a mathematical integer.
<code>long round(double a)</code>	Returns the closest long to the argument.
<code>int round(float a)</code>	Returns the closest int to the argument.
<code>double scalb(double d, int scaleFactor)</code>	Return $d \times 2^{\text{scaleFactor}}$ rounded as if performed by a single correctly rounded floating-point multiply to a member of the double value set.
<code>float scalb(float f, int scaleFactor)</code>	Return $f \times 2^{\text{scaleFactor}}$ rounded as if performed by a single correctly rounded floating-point multiply to a member of the float value set
<code>double signum(double d)</code>	Returns the signum function of the argument; zero if the argument is zero, 1.0 if the argument is greater than zero, -1.0 if the argument is less than zero.
<code>float signum(float f)</code>	Returns the signum function of the argument; zero if the argument is zero, 1.0f if the argument is greater than zero, -1.0f if the argument is less than zero.
<code>double sin(double a)</code>	Returns the trigonometric sine of an angle.
<code>double sinh(double x)</code>	Returns the hyperbolic sine of a double value.
<code>double sqrt(double a)</code>	Returns the correctly rounded positive square root of a double value.
<code>double tan(double a)</code>	Returns the trigonometric tangent of an angle.
<code>double tanh(double x)</code>	Returns the hyperbolic tangent of a double value.
<code>double toDegrees(double angrad)</code>	Converts an angle measured in radians to an approximately equivalent angle measured in degrees.
<code>double toRadians(double angdeg)</code>	Converts an angle measured in degrees to an approximately equivalent angle measured in radians.
<code>double ulp(double d)</code>	Returns the size of an ulp of the argument.
<code>float ulp(float f)</code>	Returns the size of an ulp of the argument.
<code>long factorial(int value)</code>	Returns the factorial of passed value.

Regression functions

Function	Description
<code>linereg(String indicatorCode)</code>	Linear regression
<code>polyreg(2, String indicatorCode)</code>	Quadratic regression
<code>polyreg(3, String indicatorCode)</code>	General polynomial regression

Special Functions

Special functions

Function	Description																				
<code>void filter(String filterExpression, { expression })</code>	Evaluates the passed expression with the specified filter. For example:																				
<code>Double aggregatePrevLevel(int countOfPrevlevels, { expression })</code>	Returns the aggregated value of the embedded expression. The aggreg:																				
<code>Double firstValue()</code>	Returns the lowest value of selected of selected level (measure). For ex: <table><tr><th></th><th>Date</th><th>fir</th></tr><tr><td>Project1</td><td>1/1/2015</td><td>1/1</td></tr><tr><td>Project1</td><td>5/1/2015</td><td>1/1</td></tr><tr><td>Project2</td><td>3/1/2015</td><td>3/1</td></tr><tr><td>Project2</td><td>8/1/2015</td><td>3/1</td></tr></table>		Date	fir	Project1	1/1/2015	1/1	Project1	5/1/2015	1/1	Project2	3/1/2015	3/1	Project2	8/1/2015	3/1					
	Date	fir																			
Project1	1/1/2015	1/1																			
Project1	5/1/2015	1/1																			
Project2	3/1/2015	3/1																			
Project2	8/1/2015	3/1																			
<code>Double forEachRow('expression')</code>	Computes the expression on data set row level and calculates the sum f multiplication within the <code>forEachRow()</code> method and outside: <table><tr><th></th><th></th><th>Ind1</th><th>Ind2</th><th>forEachR</th></tr><tr><td>Member</td><td></td><td>5</td><td>30</td><td>80</td></tr><tr><td></td><td>DrillDownMember1</td><td>3</td><td>20</td><td>60</td></tr><tr><td></td><td>DrillDownMember2</td><td>2</td><td>10</td><td>20</td></tr></table>			Ind1	Ind2	forEachR	Member		5	30	80		DrillDownMember1	3	20	60		DrillDownMember2	2	10	20
		Ind1	Ind2	forEachR																	
Member		5	30	80																	
	DrillDownMember1	3	20	60																	
	DrillDownMember2	2	10	20																	
<code>Double forEachRow('expression','aggregation')</code>	Computes the expression on data set row level and calculates the aggreg include SUM, MIN, MAX, AVG, COUNT, DCOUNT.																				
<code>Double formatNumber(number value)</code>	Formats the value according the indicator's format from settings.																				
<code>Double formatNumber(number value, 'String pattern')</code>	Formats the value according the indicator's format from settings. String p																				
<code>Double lastValue()</code>	Returns the lowest value of selected of selected level (measure). For ex: <table><tr><th>Project</th><th>Date</th><th>la</th></tr><tr><td>Project1</td><td>1/1/2015</td><td>5/</td></tr><tr><td>Project1</td><td>5/1/2015</td><td>5/</td></tr><tr><td>Project2</td><td>3/1/2015</td><td>7/</td></tr><tr><td>Project2</td><td>7/1/2015</td><td>7/</td></tr></table>	Project	Date	la	Project1	1/1/2015	5/	Project1	5/1/2015	5/	Project2	3/1/2015	7/	Project2	7/1/2015	7/					
Project	Date	la																			
Project1	1/1/2015	5/																			
Project1	5/1/2015	5/																			
Project2	3/1/2015	7/																			
Project2	7/1/2015	7/																			

Double members(String path, { expression })	Computes the expression for desired drill-down member values. Example: <
--	---

String memberIdentifier()	Returns the attribute code and value of current member in following form <table border="1" data-bbox="820 201 1494 396"> <tr> <td></td><td>Population</td></tr> <tr> <td>Prague</td><td>1200000</td></tr> <tr> <td>Berlin</td><td>3000000</td></tr> <tr> <td>London</td><td>7825200</td></tr> </table>		Population	Prague	1200000	Berlin	3000000	London	7825200						
	Population														
Prague	1200000														
Berlin	3000000														
London	7825200														
String memberValue()	Returns the value of current member. Example: <table border="1" data-bbox="820 525 1494 720"> <tr> <td></td><td>Population</td></tr> <tr> <td>Prague</td><td>1200000</td></tr> <tr> <td>Berlin</td><td>3000000</td></tr> <tr> <td>London</td><td>7825200</td></tr> </table>		Population	Prague	1200000	Berlin	3000000	London	7825200						
	Population														
Prague	1200000														
Berlin	3000000														
London	7825200														
rank() { expression }	Returns numerical order (rank) of the indicator applied in the expression <table border="1" data-bbox="820 846 1494 1041"> <tr> <td>User</td><td>Score</td></tr> <tr> <td>Peter</td><td>90.3</td></tr> <tr> <td>John</td><td>92.7</td></tr> <tr> <td>Anna</td><td>89.7</td></tr> </table>	User	Score	Peter	90.3	John	92.7	Anna	89.7						
User	Score														
Peter	90.3														
John	92.7														
Anna	89.7														
rankInner() { expression }	Returns numerical order (rank - ascending) of the indicator applied in the <table border="1" data-bbox="820 1167 1494 1505"> <tr> <th>Drill-down</th><th>Sales</th></tr> <tr> <td>Coffee</td><td>75000</td></tr> <tr> <td>Caffe Latte</td><td>25000</td></tr> <tr> <td>Caffe Mocca</td><td>35000</td></tr> <tr> <td>Decaf Espresso</td><td>15000</td></tr> <tr> <td>Tea</td><td>100000</td></tr> <tr> <td>Lemon</td><td>100000</td></tr> </table>	Drill-down	Sales	Coffee	75000	Caffe Latte	25000	Caffe Mocca	35000	Decaf Espresso	15000	Tea	100000	Lemon	100000
Drill-down	Sales														
Coffee	75000														
Caffe Latte	25000														
Caffe Mocca	35000														
Decaf Espresso	15000														
Tea	100000														
Lemon	100000														

<code>rankInner(true) { expression }</code>	<div>Returns numerical order (rank - descending) of the indicator applied in the</div> <table><tr><th>Drill-down</th><th>Sales</th></tr><tr><td>Coffee</td><td>75000</td></tr><tr><td>Caffe Latte</td><td>25000</td></tr><tr><td>Caffe Mocca</td><td>35000</td></tr><tr><td>Decaf Espresso</td><td>15000</td></tr><tr><td>Tea</td><td>100000</td></tr><tr><td>Lemon</td><td>100000</td></tr></table>	Drill-down	Sales	Coffee	75000	Caffe Latte	25000	Caffe Mocca	35000	Decaf Espresso	15000	Tea	100000	Lemon	100000	
Drill-down	Sales															
Coffee	75000															
Caffe Latte	25000															
Caffe Mocca	35000															
Decaf Espresso	15000															
Tea	100000															
Lemon	100000															
<code>withoutDateTime() { expression }</code>	<div>Opt-out from applied Date and Time intervals. Example:</div> <table><tr><td>Date</td><td>M_INDICATOR</td></tr><tr><td>1/12 - 12/12</td><td>1000</td></tr><tr><td>1/1/2013 - 1/31/2013</td><td>100</td></tr></table>	Date	M_INDICATOR	1/12 - 12/12	1000	1/1/2013 - 1/31/2013	100									
Date	M_INDICATOR															
1/12 - 12/12	1000															
1/1/2013 - 1/31/2013	100															
<code>withoutDrillDown() { expression }</code>	<div>Opt-out from applied Drill-downs. Example:</div> <table><tr><td>Country</td><td>M_INDICATOR</td><td>with</td></tr><tr><td>+UK</td><td>3000</td><td>3000</td></tr><tr><td>--London</td><td>1000</td><td>3000</td></tr><tr><td>--Manchester</td><td>1000</td><td>3000</td></tr><tr><td>--Oxford</td><td>1000</td><td>3000</td></tr></table>	Country	M_INDICATOR	with	+UK	3000	3000	--London	1000	3000	--Manchester	1000	3000	--Oxford	1000	3000
Country	M_INDICATOR	with														
+UK	3000	3000														
--London	1000	3000														
--Manchester	1000	3000														
--Oxford	1000	3000														
<code>withoutFilter() { expression }</code>	<div>Opt-out from applied filters. Example: (Filter is set to SEGMENT=SMB)</div> <table><tr><td>City</td><td>M_INDICATOR</td><td>without</td></tr><tr><td>Paris</td><td>1000</td><td>3000</td></tr><tr><td>London</td><td>900</td><td>3000</td></tr><tr><td>Berlin</td><td>1100</td><td>3000</td></tr></table>	City	M_INDICATOR	without	Paris	1000	3000	London	900	3000	Berlin	1100	3000			
City	M_INDICATOR	without														
Paris	1000	3000														
London	900	3000														
Berlin	1100	3000														

Passing parameters to time formula

It is possible to define the time entry by formula. Following functions are applicable for these kind of formulas only.

Function	Description
<code>void set(String name, Object value)</code>	Sets the parameter value.

Object get(String name)	Loads the parameter value.
Object indicator()	Returns indicator from the context of the row of current table. This function returns also values of formula defined indicators (def

Consider following example - we have several indicators with codes M_NAME_1, M_NAME_2 and M_NAME_3. These indicators are used in formulas 1 - 3. The time area is defined by time formulas 1 - 5.

	Time formula 1: M_NAME_1	Time formula 2: M_NAME_2	Time formula 3: M_NAME_3	Time formula 4: s = get('suffix'); return value('M_NAME' + s);	Time formula 5: value(actualYear(), 'now-1m',indicator())
Formula 1: set('suffix', '_1'); return M_NAME_1;	1000	2000	3000	1000	1000
Formula 2: set('suffix', '_2'); return M_NAME_2;	1000	2000	3000	2000	2000
Formula 3: set('suffix', '_3'); return M_NAME_3;	1000	2000	3000	3000	3000
M_NAME_1	1000	2000	3000	N/A	1000
M_NAME_2	1000	2000	3000	N/A	2000
M_NAME_3	1000	2000	3000	N/A	3000

Getting User Information

It is possible to obtain information about logged in user for your reporting needs.

Function	Description
String getSignedUserName()	Returns name of currently signed in user.
String getSignedUserSurname()	Returns surname of currently signed in user.
String getSignedUserEmail()	Returns email of currently signed in user.
String getSignedUser()	Returns username of currently signed in user.

Accessing Report Variables

Report variables are accessible using the @ prefix. For example:

Variables can be used in formulas and in custom date/time interval definition.

Return types of different variable types:

Variable type	Return type
Date	LocalDate
String	String
Number	Number
Filter	Array of Strings

Other report variables

Function	Description
DateTime lastDataUpdateTime()	Returns the last data update time as DateTime.
DateTime lastReportUpdateTime()	Returns the last update time of report as DateTime.

Referencing Data From Another Data Set

<code>crossValue(String dataSetCode, String indicator)</code>	Loads the indicator value from the specified data set. Data time context changing functions are available.
<code>crossValue(String dataSetCode, String membersIdentifier String indicator)</code>	Loads the indicator value from the specified data set. Data time context changing functions are available.

Examples:

You can find more examples in [Cross-referencing Values from Different Data Set](#) use cases.

Formula Use Cases



Make sure that you are familiar with [Formula Reference Guide](#) prior to proceeding with **Formula Use Cases**.

Formulas are very powerful and complex scripting language. You can conduct wide variety of calculations and transformations by using and combining advanced functions.

In the following sections, you can find advanced **Use cases** and detailed description of the most common tasks conducted in BellaDati simply by using **Formulas**.

Current **Use cases** include:

- [Filtering in Formulas](#)
- [Calculating Frequencies](#)
- [Calculating Percentual Share in Drill-Downs](#)
- [Calculating With Members On Defined Level](#)
- [Calculating Average Cumulated Values](#)
- [Calculating Percentiles and Quantils](#)
- [Getting Last Available Value](#)
- [Cross-referencing Values from Different Data Set](#)
- [Handling empty \(NULL\) values by formulas](#)
- [Representing numbers in accounting format](#)
- [Overriding Date Interval with Day Order](#)
- [Displaying text values in KPI labels](#)
- [Display Top or Bottom Member Value in KPIs](#)
- [Calculating average across different drill down levels](#)
- [Calculate revenue using unit price times quantity and revenue percentage](#)
- [Calculating with NULL values](#)
- [Link Relative Ratio and Year-on-year Comparison](#)



You can request help or additional use cases by contacting BellaDati analysts team at support@belladati.com

Filtering in Formulas



It is recommended to get familiar with **filter** function before proceeding with this tutorial.

You can set up [filters](#) for view in *Filter* dialog.

Sometimes it is more convenient to add **filter** directly to indicator formula. You can do this when:

- you are cross referencing data from other data sets.
- you do not want your users to change filters.

Filtering Attributes

The filter function supports most of the operators supported by PostgreSQL such as in/not in, ilike, like/not Like, similar to/not similar to etc. "ilike" is similar to "like" but case insensitive to the pattern matched.

- **Filter using operator: in/not in**

Following example filters data to include only **Paris**, **Berlin** and **London** in their **City** attribute. Result returns aggregation of visits in these three cities.



Note the correct usage of apostrophs in the example.

- **Regular Expressions**

Pattern	Description
—	Stands for any single character.
%	Stands for any sequence of zero or more characters.
*	Denotes repetition of the previous item zero or more times.
+	Denotes repetition of the previous item one or more times.
?	Denotes repetition of the previous item zero or one time.
	Denotes alternation (either of two alternatives).

Filtering Indicators

Following example counts number of records which have Indicator's value of visits greater than 3.



Note that filter is applied on every record and not on its aggregated value displayed in the view.

Date filtering

When filtering dates, the only allowed format is 'yyyy-MM-dd'. Following example returns number of students who had an exam on 12th September 2014.

For filtering whole month or year, put date-part function inside the filter. Following example return number of students who had an exam in September 2014.


Time filtering

When filtering time, the only allowed format is 'HH:mm:ss'. Following example returns number of patients who arrived at 10:00

For filtering whole hours, minutes or seconds, date_part function can be put inside the filter function. Following example returns number of patients who arrived between 10:00 and 10:59.

Multiple Filters

Following example returns number of students who had **Exceptional** results from **Math** subject.

 You can combine multiple filter conditions with **AND** or **OR** conjunction.

Calculating Frequencies



It is recommended to get familiar with memberSum function and its variations before proceeding with this tutorial.

The goal of this tutorial is to display the frequencies of visits based on visit counts in individual countries.



Visit frequencies are defined as formula indicators described below.

Download: [Demo structure](#) & [data](#).

Simple Frequencies

Use **memberSum** function to force BellaDati to aggregate data on specific level. Since frequencies for particular countries are calculated, apply **Country** attribute as **memberSum** parameter.



Formula of this indicator will count frequencies of exactly two visits. Create new calculated indicator and change condition to record other frequencies.

Visits count



	Count: Country	
	2012	
	I	
	January	February
+ Austria	2	
+ France	5	3
+ Germany	3	1
+ Poland	1	
+ Ukraine	1	1
Total	12	5

Visit frequency



	2012	
	I	
	January	February
1x visited	2	2
2x visited	1	0
>2x visited	2	1

Frequencies by Categories

Extend **memberSum** function parameters by **Category** attribute to add another dimensionality to your frequency analysis.



Formula of this indicator will count frequencies of exactly two visits. Create new calculated indicator and change condition to record other frequencies.

Visits count by categories

Count: Country				
2012				
I				
January				
	A	B	C	A
Austria		1	1	
France	3		2	
Germany	2	1		1
Poland	1			
Ukraine			1	
Total	6	2	4	1

Visit frequency by categories

2012				
I				
January				
	A	B	C	A
1x visited	1	2	2	1
2x visited	1	0	1	0
>2x visited	1	0	0	0

Frequencies by Categories and Regions

Add another parameter to **memberSum** function in order to further extend dimensionality.



Formula of this indicator will count frequencies of exactly two visits. Create new calculated indicator and change condition to record other frequencies.

Visits count by categories and regions

Count: Country				
2012				
I				
January				
	A	B	C	A
East				
Poland	1		1	
Ukraine	1			
West	5	2	1	1
Austria		1	1	
France	3		2	
Germany	2	1		1
Total	6	2	4	1

Visit frequency by categories - filter East region

2012				
I				
January				
	A	B	C	A
1x visited	1	0	1	0
2x visited	0	0	0	0
>2x visited	0	0	0	0

Visit frequency by categories and regions

2012				
I				
January				
	A	B	C	A
East				
1x visited	1	0	1	0
2x visited	0	0	0	1
>2x visited	0	0	0	1

Visit frequency by categories - filter West region

2012				
I				
January				
	A	B	C	A
1x visited	0	2	1	1
2x visited	1	0	1	0
>2x visited	1	0	0	0



Note that frequencies will not be displayed correctly when adding another drill-down on the right side of the table in current version. Use filters instead. This can be also combined with report variables.

Calculating Percentual Share in Drill-Downs



It is recommended to get familiar with **aggregatePrevLevel** function and its variations before proceeding with this tutorial.

The goal of this tutorial is to display percentual share of children members composing parent node. Formula works across drill-down levels and is universal for any depth. Every member has assigned indicator with absolute value which will be used to calculate share within the level.

Final table looks as follows:

[-] Lean It	[+] Books	7,972	5.1 %
[-] Mobile App	[+] Books	5,353	67.1 %
[+] Twitter Premium	[+] Books	2,626.7	49.1 %
[+] Direct Link	[+] Books	1,345.7	25.1 %
[+] Bing	[+] Books	705.6	13.2 %
[+] AdForum	[+] Books	675	12.6 %
[+] Mobile Kiosk	[+] Books	2,619	32.9 %

- **Lean In** books represents 5.1 % of all Sales.
- 67.1 % of Lean In were sold via **Mobile App** and 32.9 % through **Mobile Kiosk**.
- 49.1 % of mobile users came from **Twitter Premium**, 25.1 % from **Direct Link**, 13.2 % from **Bing** search and 12.6 % from **AdForum**.

Use **aggregatePrevLevel** to obtain aggregated parent value of current member. Divide actual value by received amount to express share in percentage.

Calculating With Members On Defined Level



It is recommended to get familiar with **memberSum** function and its variations before proceeding with this tutorial.

When evaluating formulas, BellaDati proceeds in following fashion:

1. Apply **Member aggregation** function to particular **Indicators**. (SUM, MIN, MAX, AVG, COUNT)
2. Execute user defined **Operations** among **Indicators**. (+,-,*,/).

However, sometimes this behavior is not demanded.



Imagine following situation. You have **Price** and **Quantity** indicators and want to display **Total Sales**. In its standard behavior, BellaDati would sum all prices and quantities and eventually multiply it. Nevertheless, right procedure will be to multiply **Price** and **Quantity** on every row and subsequently consolidate result to display **Total Sales**.

Calculating Total Sales

You can leverage **memberSum** function to force BellaDati to execute defined operation on particular level. Since multiplication of **Price** and **Quantity** is needed on every row, use **unique key** attribute as **memberSum** parameter.



This example is sufficient for indicators without drill-down path applied. Proceed further to find out how to extend this code in case of desired dimensionality.

Calculating Total Sales For Particular Drill-Down

If you want drill-down path to be considered while applying **memberSum** function, you have to explicitly define it in developed formula. Place desired attribute code before unique key definition as shown in the following example. This will ensure that your data are correctly multiplied and subsequently aggregated.



Note that order of parameters in **memberSum** function is important. Also, you still need to select particular attribute in drill-down path definition.



You can extend dimensionality by adding more parameters into the formula.

You can observe result of the applied formula in the following table. Same settings would apply for char data visualization.

To	New indicator
Barcelona	250,094,730
Berlin	120,361,610
London	327,068,920
Paris	97,397,250

Calculating Average Cumulated Values

⚠ It is recommended to get familiar with `cumulateFromDate` function before proceeding with this tutorial.

The result of this tutorial will be the cumulated value of an indicator (eg. payments) divided by number of months from the beginning of the financial year. This can tell you how the total payments average trend changes during the year.

⚠ Changes might be necessary when adding drill-down.

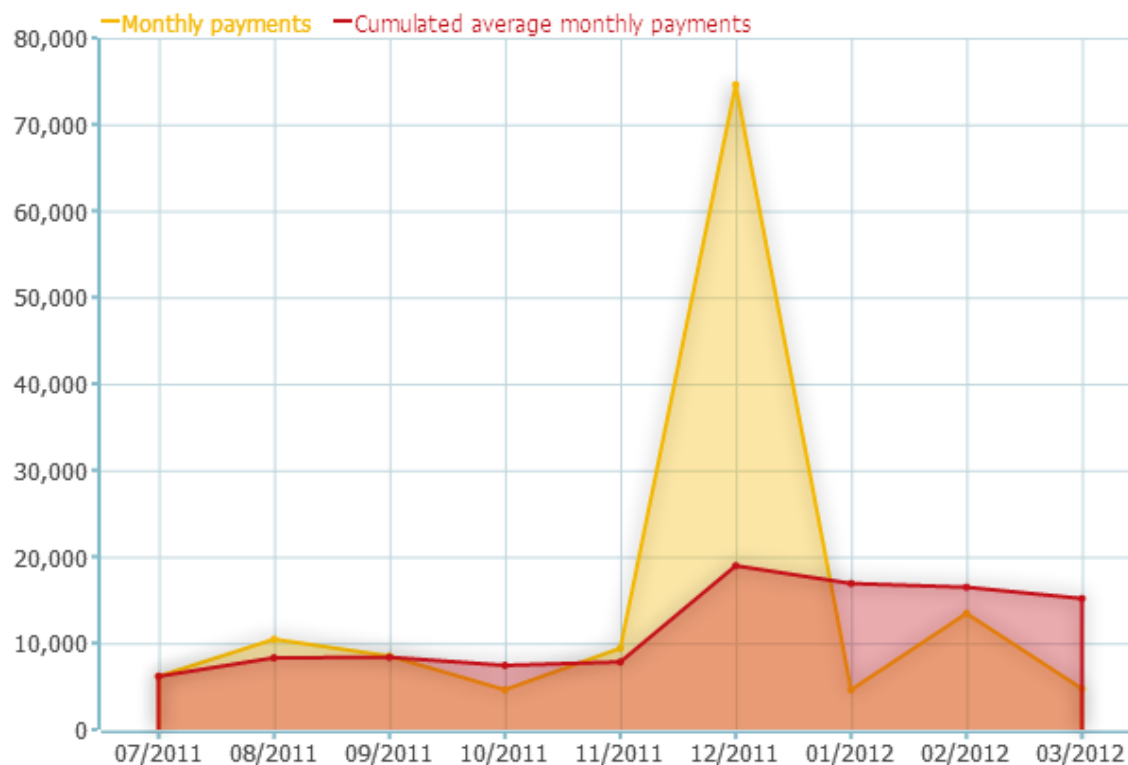
You can observe result of the applied formula in the following chart.

Average payments during the financial year 2011



From [07/2011 to 03/2012](#)

[Monthly payments](#), [Cumulated average monthly payments](#)



Calculating Percentiles and Quantils



It is recommended to get familiar with **rank()** function before proceeding with this tutorial.

For this tutorial, we will leverage **Data Set** loaded with exam scores.
Data set includes two columns:

- Student **ID**
- Student **Score**

Browse data

[Filter](#) [Add record](#) [Delete selected data](#) Order by:

	ID	Score
	1	21.36326
	2	17.58445
	3	86.38824

Percentiles



Percentile (or centile) is the value of a variable below which a certain percent of observations fall. For example, the 20th percentile is the value (or score) below which 20 percent of the observations may be found.

You desire to create table showing percentile next to score for each student.

- Create new table with student **ID** drill-down and **Score** indicator.
- Create new indicator - **Percentile**.
- Add following formula into *Indicators settings*.
- Setup percentage to **Unit** and associate it with appropriate **Format**.

1. line: Store the number of total **records** (students). Since, student drill-down is used, aggregation one level up is needed.
2. line: Obtain **rank** for each record.
3. line: Recalculate rank to **percentile**. For example, if rank is 5 from 100 students, the percentile will be: $1 - (5/100) = 95\%$.

Percentile		
	Score	Percentile [%]
30	97.5	98 %
34	97.4	96 %
26	95.7	94 %
28	94	92 %
42	93.9	90 %
40	87.6	88 %
3	86.4	86 %
46	86.2	84 %
47	85.4	82 %
23	84.7	80 %
29	84.5	78 %
33	84.3	76 %
5	84.1	73 %
22	82.3	71 %
20	81.4	69 %

Quantiles



A value which divides a set of data into equal proportions. Examples are median, quartile and decile.

You desire to create KPI label showing the median of exam scores.

- Create new KPI label.
- Create new indicator - **Quantile**.
- Add following formula into *Indicators settings*.
- Create **quantile** variable, to be able to dynamically change observed quantile.

1. Use first three lines to convert provided **quantile** variable and find the corresponding **position** within the set of scores.
2. Obtain **rank** for each score, aggregated to the level of student's **ID**.
3. If the current **rank** equals the **position**, store **score** to the **median** variable.

Quantile ✂ 🔍 📄 ★ ▶

93.9
Quantile

84.3
Quantile

54.4
Quantile

Variables

quantile 90
Save Reset Edit

quantile 75
Save Reset Edit

quantile 50
Save Edit

Getting Last Available Value

Cross-referencing Values from Different Data Set



It is recommended to get familiar with **crossValue**, **memberValue** and **filter** function before proceeding with this tutorial.

CrossValue function is used to access **Indicators** from other **Data Sets**.

It receives two parameters:

1. Cube name
2. Indicator name

Following example returns **Students** count from **Results** data set.



Access cross reference function from **Formulas help** to predefine **Cube name**.

Cross Reference with Drill-down

When cross-referencing values, BellaDati does not take in consideration applied drill-downs. It is possible that it will display same value for each member. Therefore, you need to explicitly tell BellaDati, which member is currently being processed.

You can achieve this by inserting **memberIdentifier** as second parameter of **crossValue** functions.

Observe below the differences between indicators without and with **memberIdentifier** inserted. Notice also return value of the function in third column.



This will also automatically handle drill-down paths.

plant	Man Hours (Cross-referenced WITHOUT Member Identifier)	Man Hours (Cross-referenced WITH Member Identifier)	Member Identifier Function
Amos	29,241,197	815,197	[L_PLANT={Amos}]
0	29,241,197	685,127	[L_PLANT={Amos}][L_UNIT={0}]
Painters	29,241,197	10,523	[L_PLANT={Amos}][L_UNIT={0}][L_CRAFT={Painters}]
1	29,241,197	4,326	[L_PLANT={Amos}][L_UNIT={1}]
2	29,241,197	125,616	[L_PLANT={Amos}][L_UNIT={2}]
Boilermakers	29,241,197	64,656	[L_PLANT={Amos}][L_UNIT={2}][L_CRAFT={Boilermakers}]

Cross Reference with Filter

BellaDati also does not take in consideration filters applied through view settings. Therefore, you need to explicitly tell BellaDati in formula definition which filters and how do you want to use them.

Cross Reference with Date

When filtering date, date attribute must be in format 'yyyy-MM-dd'.

filtered date can also defined inside the crossValue function:

You can find more about **filters** and their combinations in [Filtering in Formulas](#).

Handling empty (NULL) values by formulas

Sometimes it happens that some members of your table do not have any data to display. This can be caused by:

- Specific selection of date interval.
- Application of specific filters combination.
- Missing data in underlying data set.

By default, BellaDati will display empty cell.

		PeriodIntervalQuarter1_2		Quarter Difference\$
		2010		
	Item	I	II	
Total ▾	Binder	999.5	838.8	838.80001
	Pen Set			
	Pencil	536.13	1,111.13	1,111.13
	Pen	539.73		
	Desk			

Nevertheless, you can use **Replace empty value** option to define specific numeric value, which will be displayed in the cells.

		PeriodIntervalQuarter1_2		Quarter Difference\$
		2010		
Item		I	II	
Total	Binder	999.5	838.8	838.80001
	Pen Set	0	0	0
	Pencil	536.13	1,111.13	1,111.13
	Pen	539.73	0	0
	Desk	0	0	0

Following code can be used to catch all empty cells:

Using accounting dash ('-')

Use following code to replace empty value will desired symbol (String):

		PeriodIntervalQuarter1_2		Quarter Difference\$
		2010		
Item		I	II	
Total	Binder	999.5	838.8	-160.69999
	Pen Set	0	0	-
	Pencil	536.13	1,111.13	575.00001
	Pen	539.73	0	-
	Desk	0	0	-

Representing numbers in accounting format

Sometimes, you may want to represent the numbers on the report in **accounting format**.

If the numbers are negative, it will be represented within a bracket with the positive number, e.g. - 500 is represented as (500) in accounting format.

If the numbers are 0, it will be shown as "-".

Here is the sample code for you to transform your number to accounting format. The code of the indicator using is "M_AMOUNT" and the aggregation is SUM.

Overriding Date Interval with Day Order



Make sure to get familiar with BellaDati [formulas](#) before proceeding with this section.

There are cases that you want your chart x-axis to display number of days for a particular event, rather than the calendar date.

For example, if you are running a marketing campaign from 2013-08-01 to 2013-10-01, you want to display:

- 2013-08-01 to be "Day 01"
- 2012-08-02 to be "Day 02"
- 2013-10-01 to be "Day 62" and so on.

This use case is going to show you how you could make it if you only have the date in your data set.

Preparing Data with Transformation script

Create a new attribute called **campaign_day** in the data set and apply transformation script as seen below:

This will return the number of days in terms of the campaign starting time in the format of 00 to 62.

Creating formula

At the report level, set the **campaign_day** as the drill down and calculate the indicator as seen below:

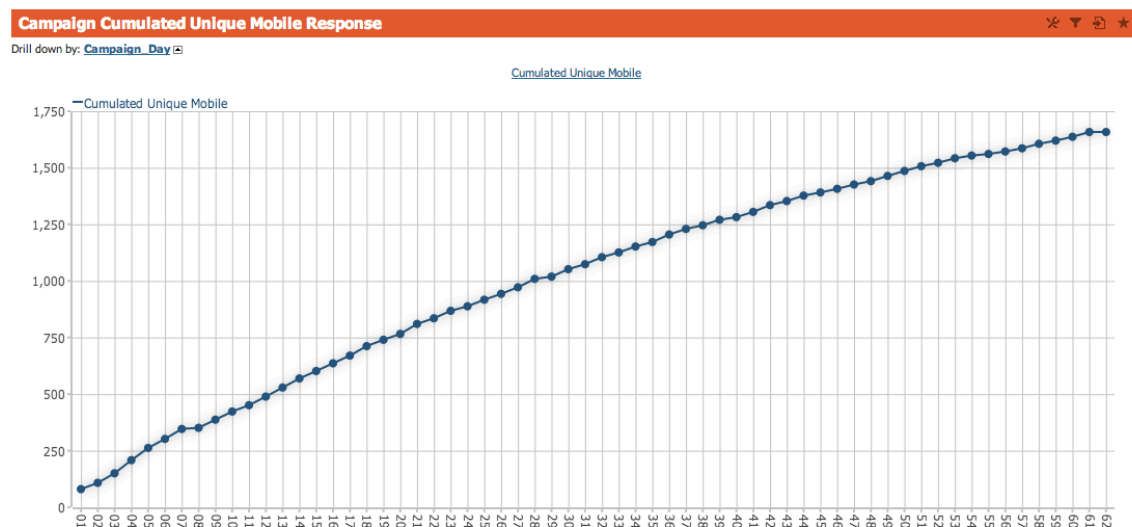


It is recommended that you get familiar with [Datetime Functions](#).



Remember that Cross Reference does not take drill down into consideration, so you are able to calculate the cumulated value. Here is how you use cross reference magically!

Here is how it looks like in the chart:



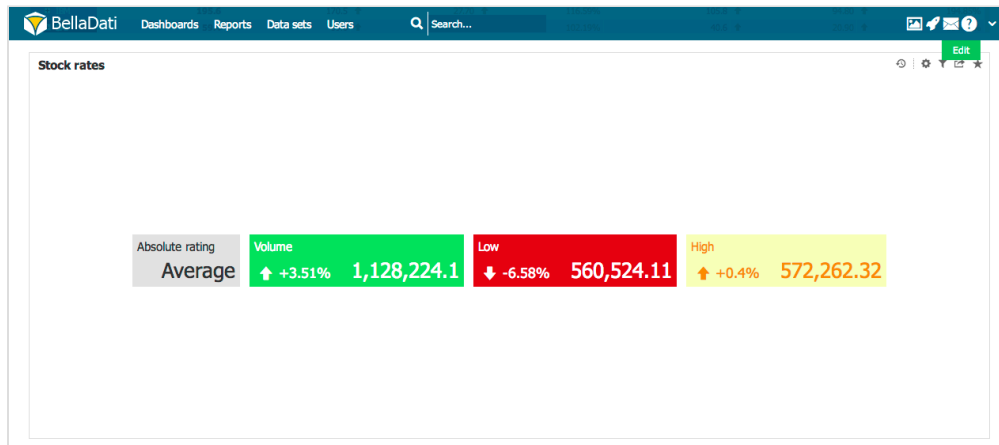
Displaying text values in KPI labels

BellaDati allows you to display **text-based KPI labels**. Use this feature to translate rates info easily readable tiers or as alternative to conditional formatting.

Example:

M_RATING@AVG indicator stores average rating of the institution. Instead of displaying numbers with conditional formatting, report requires to have text-based evaluation including *'At-Risk'*, *'Below Average'*, *'Average'*, *'Good'*, and *'Excellent'*. Use code bellow to obtain such result.

Result:



Display Top or Bottom Member Value in KPIs

You can display the top/bottom member value as KPIs by using BellaDati formulas.



It is recommended to get familiar with membersSum and memberValue function before proceeding with this tutorial.

Example:

If you want to display the name of the top city according to number of units sold, here is code you could use. Here we use membersSum to loop through each city for the total units.

You can also write the formula in this way:

If you want to display the bottom city with least total units. Here it is:

Here is how the result will look like in the view:



Calculating average across different drill down levels

It happens very frequently that we use a few drill downs in our tables, and we want to calculate the average of indicators at different levels.



It is recommended to get familiar with memberValue and crossValue functions before proceeding with this tutorial.

In the table below, you can see two drill down levels (**Employee Name** and **Product**). **Avg. Rating** is calculating the average rating for that employee on that product.

Employee Specialisation Analysis



Employee Name	Avg. Rating	Avg. Benchmark
Sam	3.24	
Product C	3.47	3.03
Product A	3.36	3.18
Product E	3.31	3.2
Product B	3.25	3.13
Product D	2.87	2.78
Jason	3.05	
Product A	3.56	3.18
Product C	3.33	3.03
Product E	3	3.2
Product D	2.78	2.78
Product B	2.5	3.13
Peter	3	
Product E	3.43	3.2
Product D	3.25	2.78
Product B	3	3.13
Product A	2.75	3.18
Product C	2	3.03
Sunny	3	
Product B	3.5	3.13
Product E	3.33	3.2
Product A	3	3.18
Product D	3	2.78
Product C	1.75	3.03
Jasmine	2.33	
Product B	3.33	3.13
Product C	3	3.03
Product A	2.2	3.18
Product E	2	3.2
Product D	1.5	2.78

But you may want to compare the employee's average rating for a product with the overall employee rating for the same product as a benchmark, as it is shown in the **Avg. Benchmark** column. This could be added as a conditional formatting, so if the employee's avg. rating on a product is higher than overall employee's rating on this product, we can mark it as green and show an upper arrow there.

Here is how you could calculate this.



It is recommended to get familiar with crossValue formula before proceeding with this tutorial. As crossValue will not take drill downs applied in the context view.

In this table below, now **Product** is being placed as the first level of drill downs, and then **Employee Name** is the second level. **Avg. Benchmark** is still the overall employee average on the product. Changing the order of the drill downs will make calculating "Avg. Benchmark" a little bit different.

Product Best Employee Analysis



Product	Avg. Rating	Avg. Benchmark
Product E	3.2	3.2
Peter	3.43	3.2
Sunny	3.33	3.2
Sam	3.31	3.2
Jason	3	3.2
Jasmine	2	3.2
Product A	3.18	3.18
Jason	3.56	3.18
Sam	3.36	3.18
Sunny	3	3.18
Peter	2.75	3.18
Jasmine	2.2	3.18
Product B	3.13	3.13
Sunny	3.5	3.13
Jasmine	3.33	3.13
Sam	3.25	3.13
Peter	3	3.13
Jason	2.5	3.13
Product C	3.03	3.03
Sam	3.47	3.03
Jason	3.33	3.03
Jasmine	3	3.03
Peter	2	3.03
Sunny	1.75	3.03
Product D	2.78	2.78
Peter	3.25	2.78
Sunny	3	2.78
Sam	2.87	2.78
Jason	2.78	2.78
Jasmine	1.5	2.78

The code is as below:

Calculate revenue using unit price times quantity and revenue percentage

There are times that the data set only contains the unit price and sold quantity without the calculated revenue. This tutorial will show you how to calculate the revenue and revenue percentage towards total revenue.



It is recommended to get familiar with memberSum function before proceeding with this tutorial.

The table we are going to build will have two drill down levels, **Product Group** and **Product Name**. Here is how the table will look like.

Product	Product Name	Unit Price	Quantity	Revenue (Unit Price * Quantity)	Percentage (Revenue / Total Revenue) (%)
Group A	Name1	100	30	3,000	8.2 %
	Name4	400	11	4,400	12.02 %
	Name6	600	10	6,000	16.39 %
Group B	Name1	100	6	600	1.64 %
	Name2	200	35	7,000	19.13 %
	Name7	700	6	4,200	11.48 %
Group C	Name1	100	4	400	1.09 %
	Name3	300	12	3,600	9.84 %
	Name5	500	10	5,000	13.66 %
Total	Name8	600	4	2,400	6.56 %
		280	128	36,600	100 %



Make sure you are displaying the **Unit Price** correctly. The members aggregation should be **Average** rather than **Sum**.

The **Revenue**, which equals to **Unit Price** times **Quantity** is calculated using **membersSum**,

Make sure you place the order of the drill down levels in the parameters of membersSum correctly. It should follow the order of the drill downs in the table, so **Product Group** first and then **Product Name**. The last drill down level **Transaction ID** will make sure it aggregates total revenue for all transactions belongs to the same product group and same product.

In order to calculate the revenue percentage, we need to calculate the total revenue first.

So the code for calculating the revenue percentage is as below:

Calculating with NULL values



Download [data file](#) for this use case and upload it in your domain.

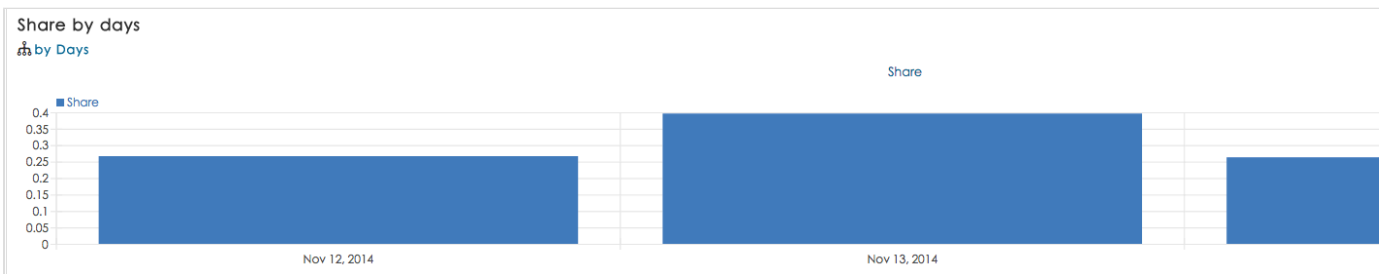
This tutorial gives you an example how to handle formulas which contain calculations with 'null' result. Otherwise when you are counting with null values the result is always 0.

```
def a = (filter("L_PRODUCT in ('TV')") {M_SALES})
def b = (filter("L_PRODUCT in ('Radio')") {M_SALES})
def c = (filter("L_PRODUCT in ('PC')") {M_SALES})

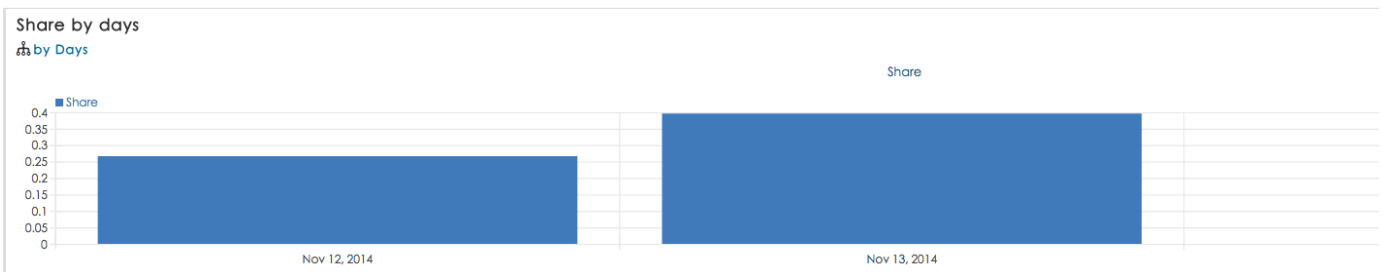
if(a == null || a == Double.NaN) {
a = 0
}
if(b == null || b == Double.NaN) {
b = 0
}
if(c == null || c == Double.NaN ) {
c = 0
}
return a/(a + b + c)
```

You can observe result on the visualisations below.

Result:



Before:



Link Relative Ratio and Year-on-year Comparison



It is recommended to get familiar with **dateAt** and **dateInterval** function before proceeding with this tutorial.

Link relative ratio

Link relative ratio refers to comparison between the current data and data of the previous period, for example value of indicator before one month. This can be done in BellaDati with very simple formula:

If you need to aggregate the value for a period of time, use this formula instead:

Year-on-year Comparison

Year-on-year comparison is made between the results for a current period and the same period in the previous year. This can be done in BellaDati with very simple formula:

If you need to aggregate the value for a period of time, use this formula instead:

Setting Date Interval

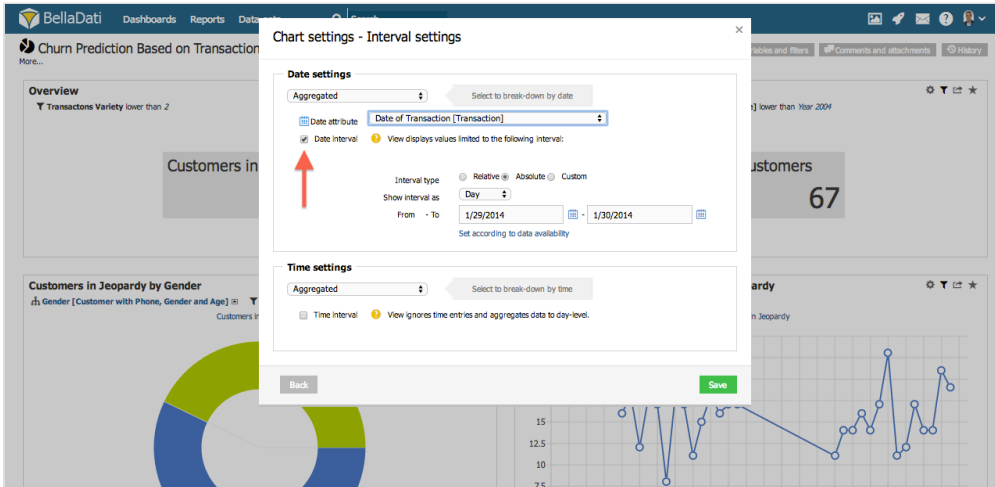


This option is related to views. Always refer to [views](#) or particular view types ([table](#), [chart](#), [Geo map](#) or [KPI label](#)) before proceeding with this section.

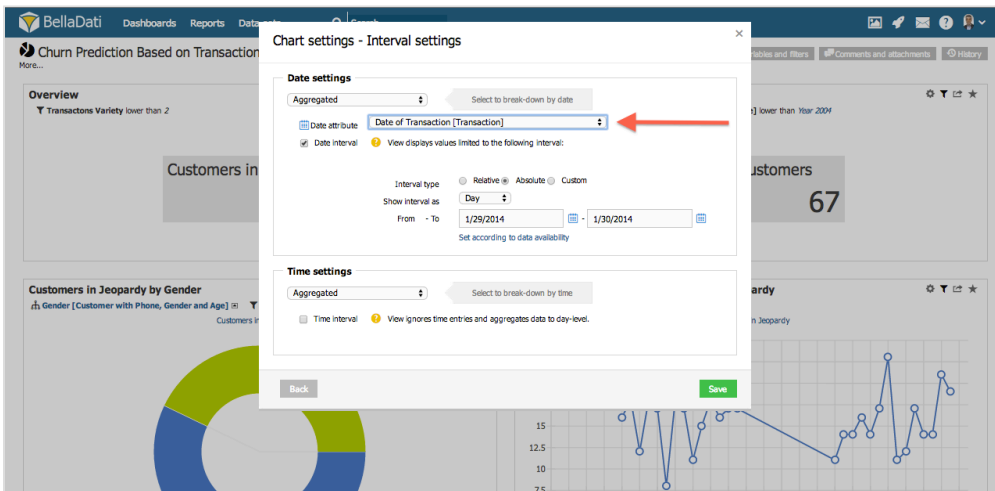
You can access **Date interval** options from the *Add view* dialog by selecting the "Date interval" checkbox.



If left blank, view displays values aggregated for the whole date interval.



When multiple date attributes are available, BellaDati will prompt you to choose the desired one.



There are three different types of intervals:

- **absolute** - (common setting) displays time from the first to the last day of the selected time unit (granularity)
- **relative** - indicates the range of the time interval from actual day according to selected time unit (granularity)
- **custom** - allows user to choose the interval on daily basis and keep arbitrary time granularity

You can set up following options:

- **Unit**: granularity of time period. Available options include: **Day**, **Week**, **Month**, **Quarter** and **Year**.
- **Interval type**: type of the interval used to restrict data. Available options include: **Absolute**, **Relative** and **Custom**.
- **From - To**: starting and ending boundary of desired time period. Format depends on selected **Unit** and **Interval type**.
- **Aggregate**: aggregates the data according to the indicator's time aggregation type
- **Set according data availability**: populates **From** and **To** inputs with frontal date values.
- **Filter day of week**: filters displayed values only for selected day of the week.



For [Geo map](#) and [KPI label](#) the **Time interval** option is available through view settings.

Entering custom date/time values


- you can enter time (date) **absolutely in two different formats**: **dd.MM.yyyy** (e.g. 1.12.2010), or **yyyy-MM-dd** (e.g. 2010-12-01)
- it's also possible to enter date **relatively**:
 - **now** - represents actual date
 - **availableFrom** - represents the date the data are available from
 - **availableTo** - represents the date the data are available to
 - **actualyear** - represents the first day of actual year (1.1.20XX). For example actualyear selected on 21.9.2010 represents date 1.1.2010
 - **actualquarter** - represents the first day of actual quarter (1.1.20XX, 1.4.20XX, 1.7.20XX, 1.10.20XX). For example actualquarter selected on 21.9.2010 represents date 1.7.2010
 - **actualmonth** - represents the first day of actual month (1.1.20XX, 1.2.20XX, ...). For example actualmonth selected in 21.9.2010 represents date 1.9.2010
 - **actualweek** - represents first day of actual week (Monday). For example actualweek selected on 21.9.2010 represents date 20.9.2010 (Monday of this week in calendar)
 - relative and absolute enterig of date can be adjusted by operators using this syntax: **date +/- n[d|w|m|q|y]**, where **n** is integer, **d** represents day, **w** represents week, **m** represents month **q** represents quartal and **y** represents year. We can for example define time in this way: *actualyear + 2m -4d*. Today is 21.9.2010, so this value represents 1.1.2010 + 2 months - 4 days, which means date 25.2.2010.
- similar to date, for specifying the time **relatively**, use the following syntax:
 - **now, actualTime** - represents the actual time
 - **availableFrom** - represents the time the data are available from
 - **availableTo** - represents the time the data are available to
 - **actualhour** - represents the actual hour, e.g. 11:00:00 AM
 - **actualminute** - represents the actual minute, e.g. 11:23:00 AM
 - **actalsecond** - represents the actual second, e.g. 11:23:45 AM

Using Filters

Filters serve for analysing subsets of your data. BellaDati support three types of filters:

1. **View Filters** - filter is applied on the current view only, doesn't affect other views in the report
2. **Report Filters** - filter is applied on all views within the report
3. **Indicator Filters** - filter is applied on the desired indicator

Using Filters in Views

 **Filter** option is available for Chart, Table, Geo map and KPI label view.

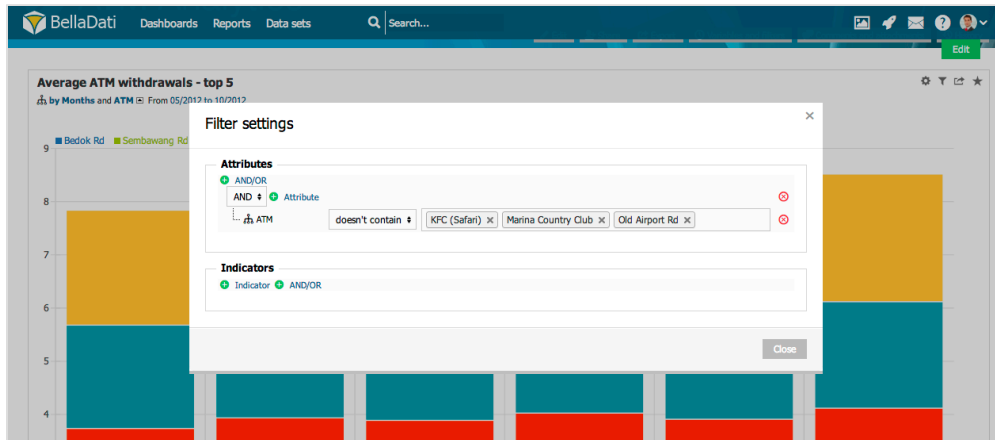
You can access **Filter** option from toolbox in the upper right corner of the view.

Filter dialog includes list of attributes and indicators used in the view. You can filter either by:

1. **Attribute** members
2. **Indicator** values

To apply a filter, you have to:


1. Select **Attribute** or **Indicator** for filtering.
2. Select **Condition**.
3. Provide **Values** for the condition.



Filtering by Attribute Members

There are four conditions available:

- **contains**: Allows to select single or more members, which will be displayed within the attribute.
- **doesn't contain**: Allows to select single or more members, which will not be displayed within the attribute.
- **count**: Allows filtering members by their count.
- **not empty**: Hides members with empty (blank) values.
- **empty**: Displays members with empty (blank) values.

 You can combine different attributes or even more conditions for one attribute. More filters are combined with **AND** condition. Therefore all conditions must be fulfilled to display data.

Filtering by Indicator Values

There are the following conditions available:

- equals the value specified
- not equal to value specified
- lower than value specified
- lower than or equal to value specified
- greater than value specified

- greater than or equal to value specified
- not empty
- empty



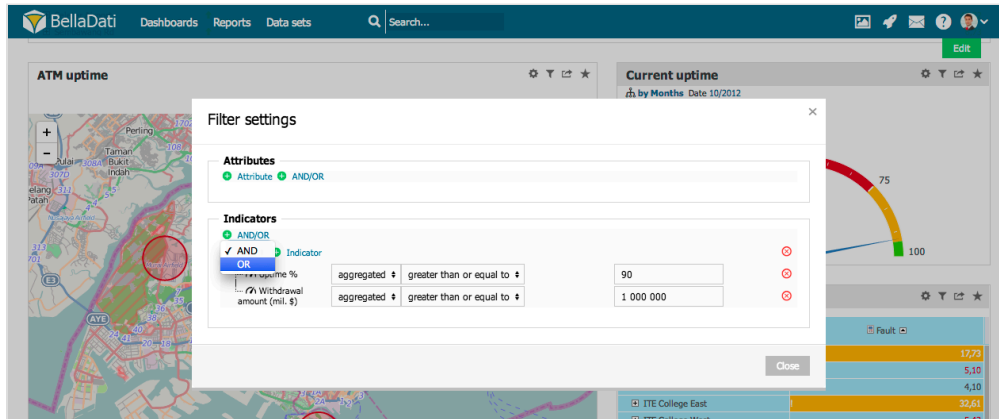
Filtering by indicator values is defined only for basic cases. It can produce unpredictable result if you use complex drill-down paths or date/time dimension.



Hint: See **crossValue()** function to access all data in the view even if the filter is active. This allows you to calculate eg. ration of some value to total which is not affected by current view filter.

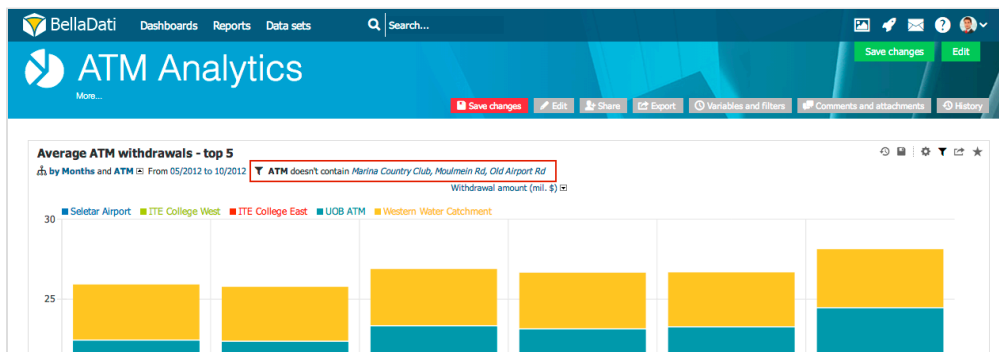
Multiple filters

You can combine different attributes/indicators or even apply more conditions for one attribute/indicator. Multiple filters can be merged with **AND** or **OR** condition.



Modifying filter

Applied **filters** are listed under the view title and labeled with **Filter** tag. In case of **attribute** filter, you can click on the condition to modify values.



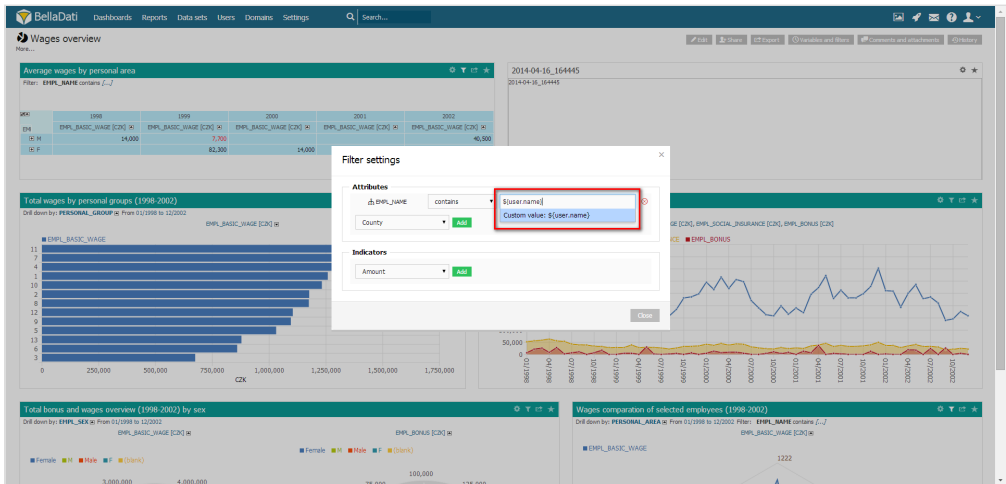
Referencing current user attributes in filter

You can access the current signed user attributes to build the filter. In order to do this, add a custom expression as a filter value:

Name	Description
<code>\${user}</code> , <code>\${user.username}</code>	Returns the username of currently signed user
<code>\${user.email}</code>	Returns the email of currently signed user
<code>\${user.name}</code>	Returns the first name of currently signed user
<code>\${user.surname}</code>	Returns the last name of currently signed user
<code>\${user.locale}</code>	Returns the user's locale of currently signed user
<code>\${user.info}</code>	Returns value from user field additional info.

<code>\${domain}</code> , <code>\${domain.name}</code>	Returns the name of currently signed user's domain
<code>\${domain.locale}</code>	Returns the locale of currently signed user's domain
<code>\${domain.timezone}</code>	Returns the timezone of currently signed user's domain

See example here:



Using wildcards in filters

You can use wildcards in BellaDati which allow you to substitute any number of characters in the custom value.

Name	Description
<code>%</code>	Replace zero or any number of characters.
<code>_</code>	Replace one single character


Using Variables in view filters

See [Report Variables and Filters](#) components

Using Report Filters

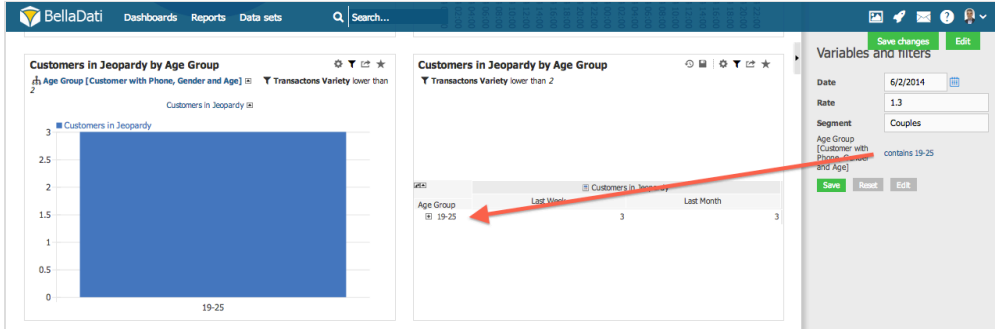
Report filters are leveraging the [report variables](#).

When **Filter** variable is set up to drill-down path, all **Views** including this **Drill-down** will be filtered according the definition.

 Click on **Filter** values in *Variables and Filters* panel to change them.

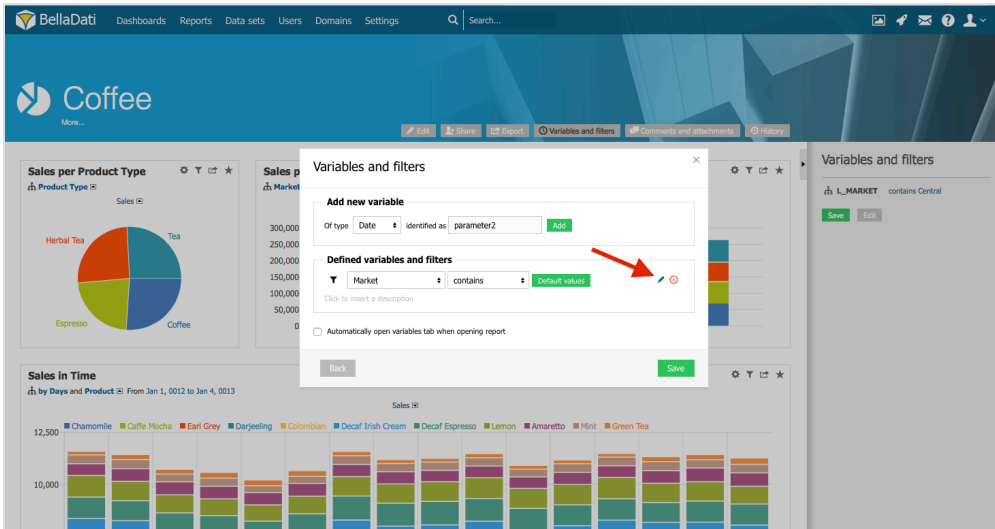
In such dynamic definition, the **Views** including the filtered **Drill-down** will be refreshed after every modification done to variable in right *Variables* panel.

For more information about **Setting Filters** on views - continue by [Using Filters](#).



Filter and variable names

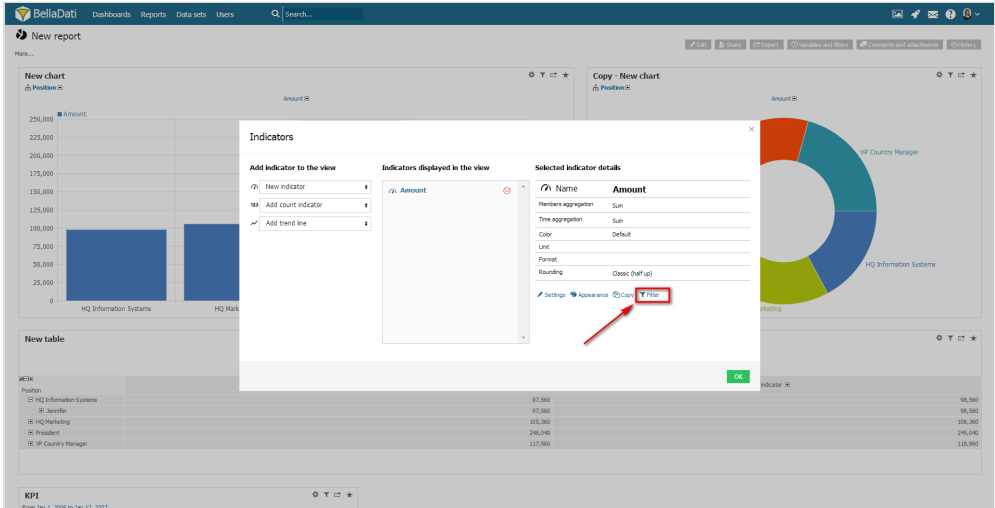
When creating a filter or variable you are able to give it a name or a code. Just click on the settings sign and give a name to your variable or filter.



Using Indicator Filters

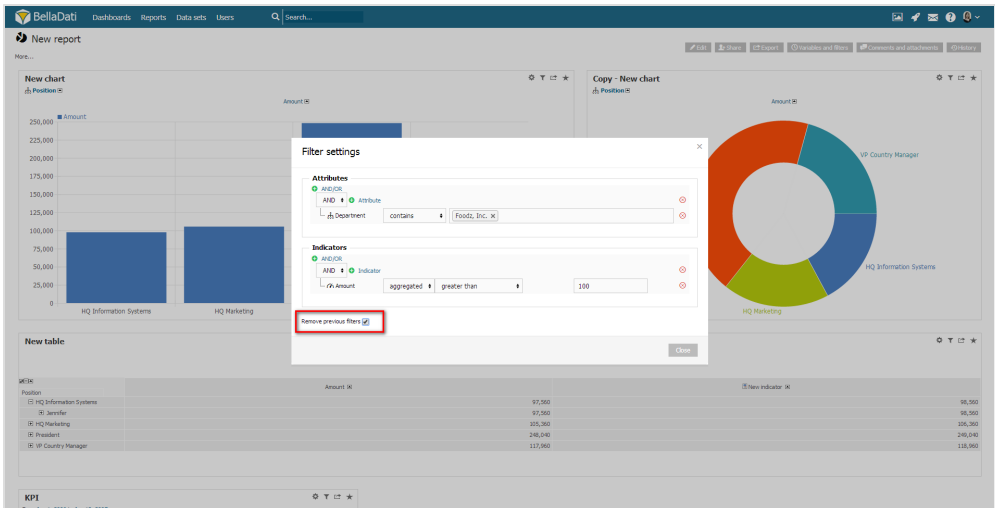
In addition to [view filters](#) and [report filters](#), you can add specific filter for the desired indicator. There are two ways - using [the formula](#), or more user-friendly, using the Indicator Filter UI.

To configure the indicator filter, open the Indicators list and select the desired indicator. Then just click on the "Filter" link and set the requested filter.



Ignoring view and report filters

There can be a case, when you don't want to apply the view and report level filters. In this case, you can just select the "Remove previous filters" option and the filters will not be applied for the Indicator value calculation.

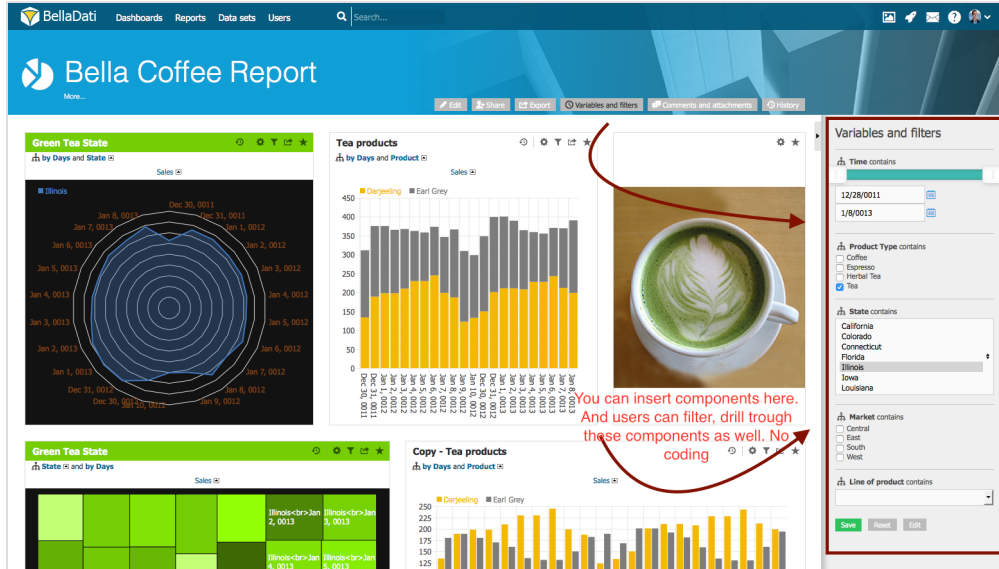


Report Variables and Filters components

Variables and Filters allows you to dynamically **modify content** of the report. It is feature, that makes filtering and drilling into the report very straightforward for report viewers. Since all filter options, data and time are accessible via custom components. Report creator can apply these components into the report. See screenshot below.

On this page:

- [Editing Variables](#)
- [Applying variables in indicator settings](#)
- [Applying variables in time interval settings](#)
- [Applying variable as relative date](#)
- [Applying variables as global report filters](#)
- [Using Variables in view filters](#)
- [Using Variables in Report and View Titles](#)
- [Next Steps](#)



You need to be in [view mode](#) in order to add variables and filters.

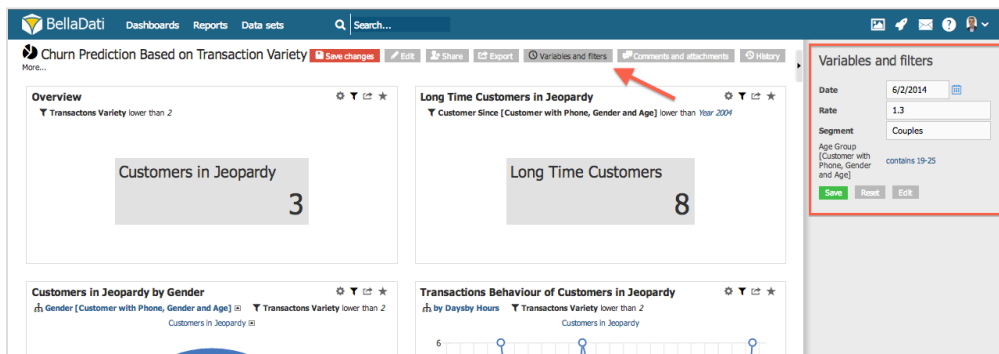
Only [report author](#) or [report editor](#) can create report variables and set their default values. Every other user that has access to the report, can only change report variable values.

Select **Variables and Filters** from the report toolbox list in the upper right corner to open *Variables* panel.

Variables and Filters dialog allows you to:

- **Change** variables values.
- **Reset** variables values.
- **Save** variables values.
- **Edit** variables definition.
- **Build** custom components (selects, radio buttons etc.) for filters

Variables are set up on report level and work only within this space.



Editing Variables

Click on **Edit** button to enter *Variables* dialog.

BellaDati allows you to:

- Select variable **type** including:
 - **Number**
 - **Text**
 - **Date**
 - **Filter**
- Edit variable **name**
- Specify **description**
- Setup **default** value
- Edit variable **value**



Select **Show variables settings panel to report users** if you want variables settings to be visible to report users.

	Variable	Value	Unit	
<input checked="" type="checkbox"/>	USDtoEUR	1.3		✕
<input checked="" type="checkbox"/>	From	1/1/2012	Day	✕
<input checked="" type="checkbox"/>	To	2/29/2012	Day	✕

Applying variables in indicator settings

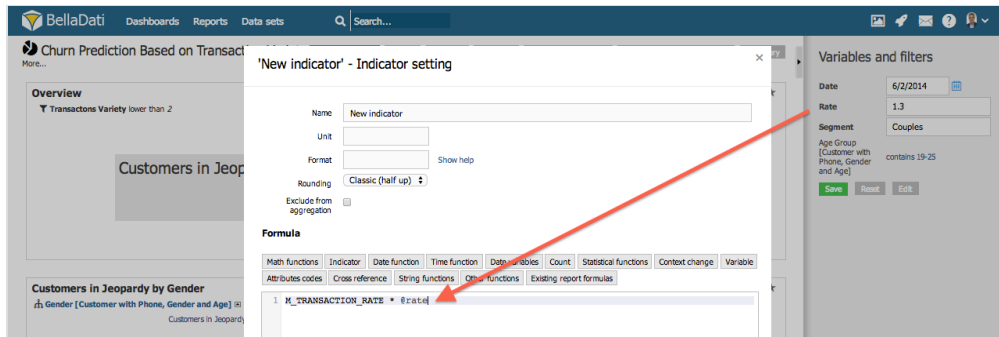
You can leverage variables in indicators **formulas**.

In such dynamic definition, the **Views** including the **Indicator** will be recalculated after every modification done to variable in right *Variables* panel.

For more information about **Displaying Indicators** - continue by [Displaying Indicators](#).



Prefix variable with '@' in order to refer to variable value.



Report **Variables** could also serve as a basic **planning and forecasting tool**. Analyst can setup **formulas** for various **versions** or **time intervals** and then change variables values in order to observe impact of such changes.

Applying variables in time interval settings

You can leverage variables in time interval settings. Variable has to have proper date format in order to be able to be used as date interval.



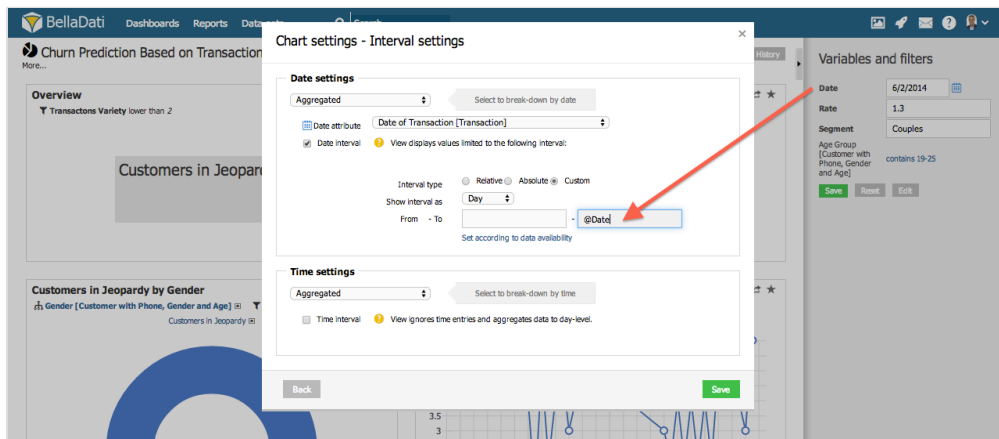
Select **Custom interval** to apply variables.

In such dynamic definition, the **Views** including the **Time interval** will be recalculated after every modification done to variable in right *Variables* panel.

For more information about **Setting Date Interval** - continue by [Setting Date Interval](#).



Prefix variable with '@' in order to refer to variable value.



Applying variable as relative date

You can also use relative dates in variables. To allow this option, go to *Variables and filters*, create date variable, go to advanced settings for this variable.

This window allows you to:

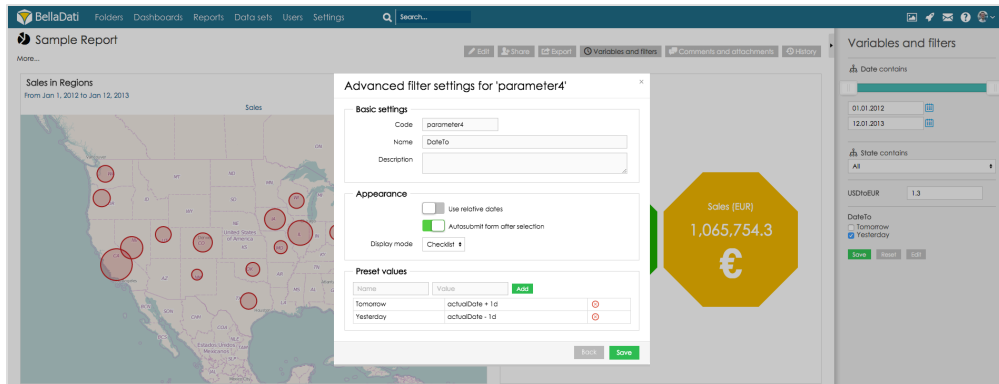
- Display default relative dates by selecting option **Use relative dates**
- Allow **Autosubmit after selection** (every time user selects a value, the setting will be automatically saved)
- Select **Display mode** (Radio, Select, Checklist)
- Set **Preset values** (displayed **Name** of the value, and **Definition** of the value - g.e. "now-2d" as the day before yesterday)

In the definition of the value can be used following date strings:

- **now** - represents actual date
- **actualyear** - represents the first day of actual year (1.1.20XX). For example actualyear selected on 21.9.2010 represents date 1.1.2010
- **actualquarter** - represents the first day of actual quarter (1.1.20XX, 1.4.20XX, 1.7.20XX, 1.10.20XX). For example actualquarter selected on 21.9.2010 represents date 1.7.2010
- **actualmonth** - represents the first day of actual month (1.1.20XX, 1.2.20XX, ...). For example actualmonth selected in 21.9.2010 represents date 1.9.2010
- **actualweek** - represents first day of actual week (Monday). For example actualweek selected on 21.9.2010 represents date

20.9.2010 (Monday of this week in calendar)

- **availableFrom**, **availableTo** - represents the first and last available date entry
- relative and absolute enterig of date can be adjusted by operators using this syntax: **date +/- n[d|w|m|q|y]**, where **n** is integer, **d** represents day, **w** represents week, **m** represents month **q** represents quartal and **y** represents year. We can for example define time in this way: *actualyear + 2m -4d*. Today is 21.9.2010, so this value represents 1.1.2010 + 2 months - 4 days, which means date 25.2.2010.

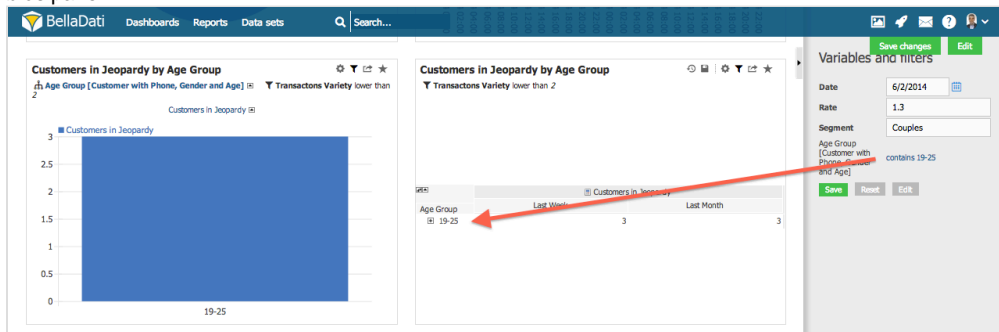


Applying variables as global report filters

You can leverage variables also as global filters. When **Filter** variable is set up to drill-down path, all **Views** including this **Drill-down** will be filtered according the definition.

 Click on **Filter** values in *Variable* panel to change them.

In such dynamic definition, the **Views** including the filtered **Drill-down** will be refreshed after every modification done to variable in right *Variables* panel.



Using Variables in view filters

For more information about **Setting Filters** on views - continue by [Using Filters](#).

Variables can also be used in view filter.

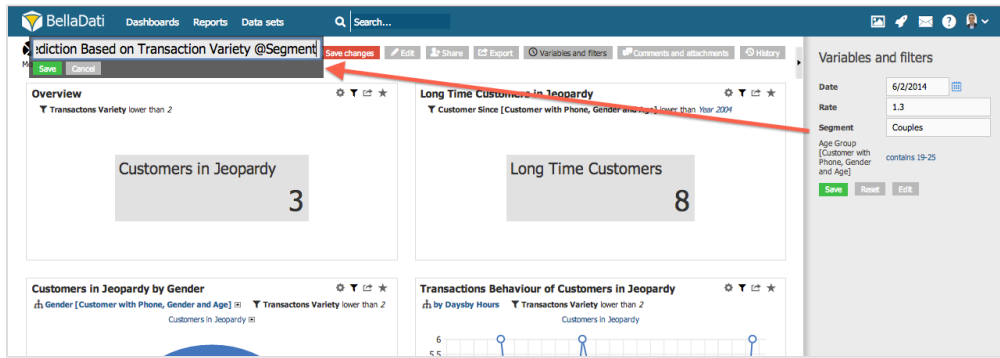
1. Create new variable, for example "var".
2. Add filter to view.
3. Type **\$_{@var}** and click on **Custom value: \$_{@var}** to confirm.
4. Close the window and Save filter and report.

Using Variables in Report and View Titles

You can use **variable values** in reports' and views' titles. To do so:

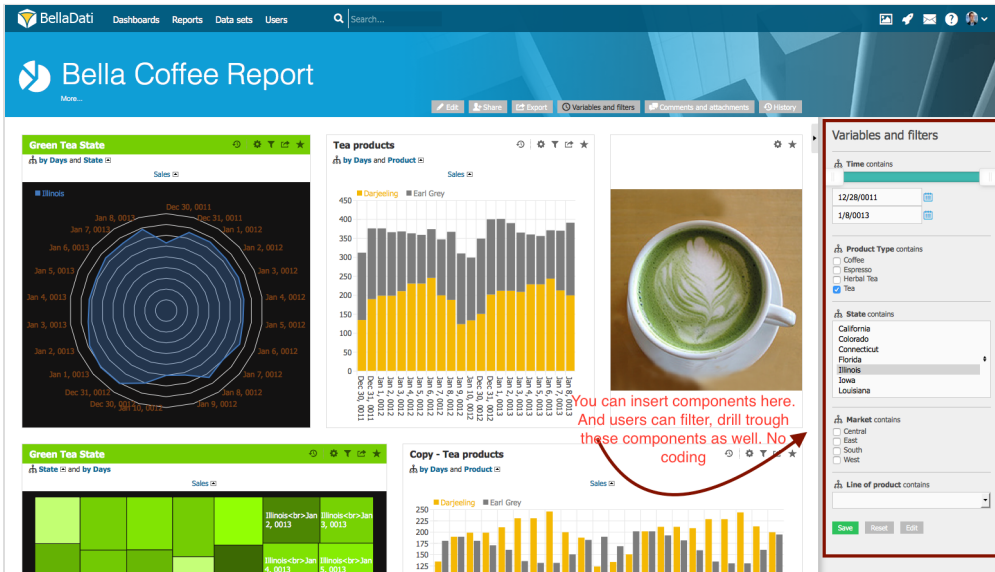
1. Click on **Title** you would like to modify
2. Type **@** followed by **Variable Name**. For example: 'Patient Details: @patient_name'.
3. Click **Save** or hit **Enter**

 This feature is handy especially when creating **Global Filters** or **drill-thoughts** with **URL Appearance**.



Custom Filter Components

Apart from filters available in Charts, Tables menu. Report creator can create filters, that are accessible for report viewers in right panel through custom components. Result for viewer can look like screen below

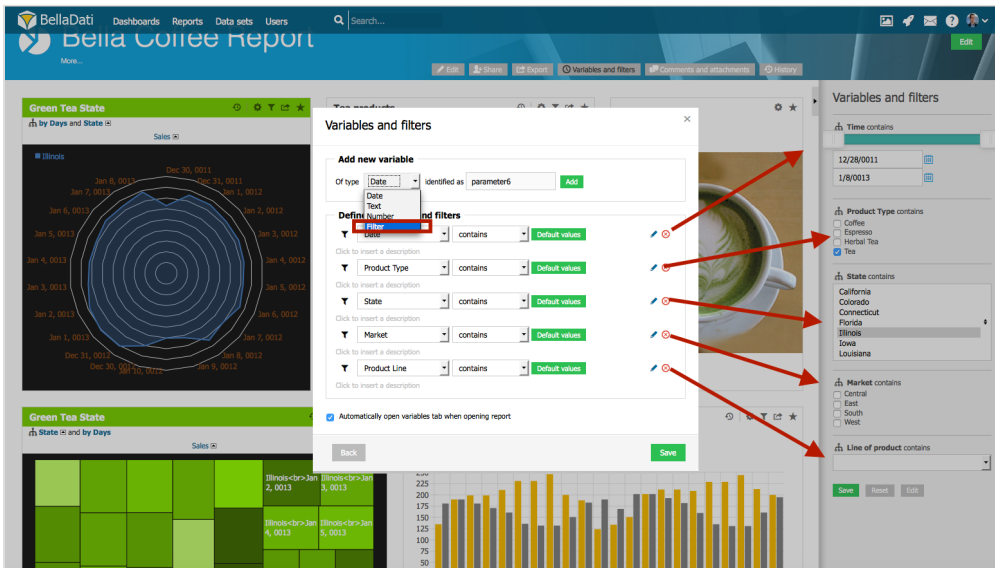


List of custom Filter components includes.

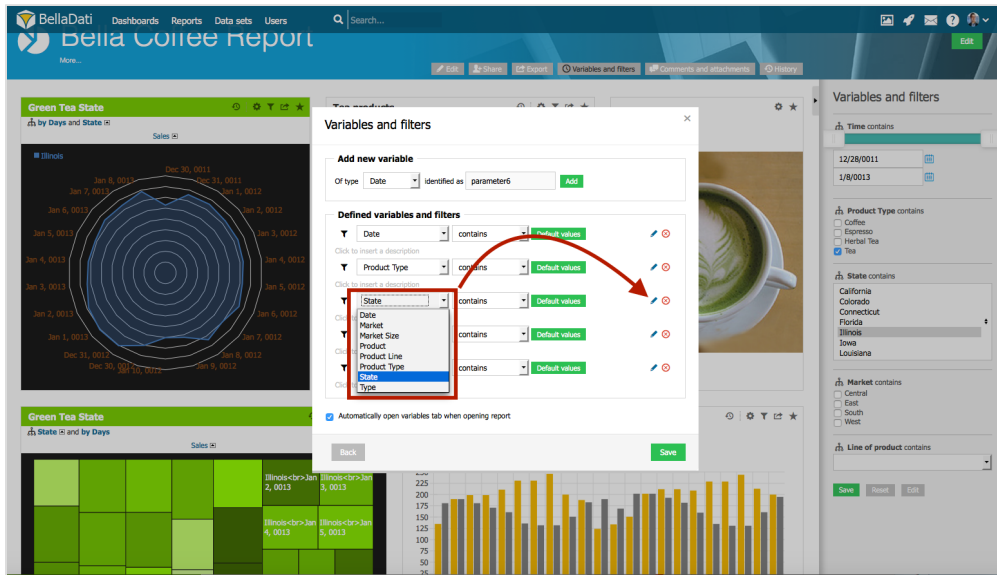
- Slider
- Multiple select
- List
- Check box
- Radio buttons
- Text

Creating the filter and GUI component

Select Variables and Filters option. As a Variable type select Filter option



Select available filter attribute from which you want to build your component.

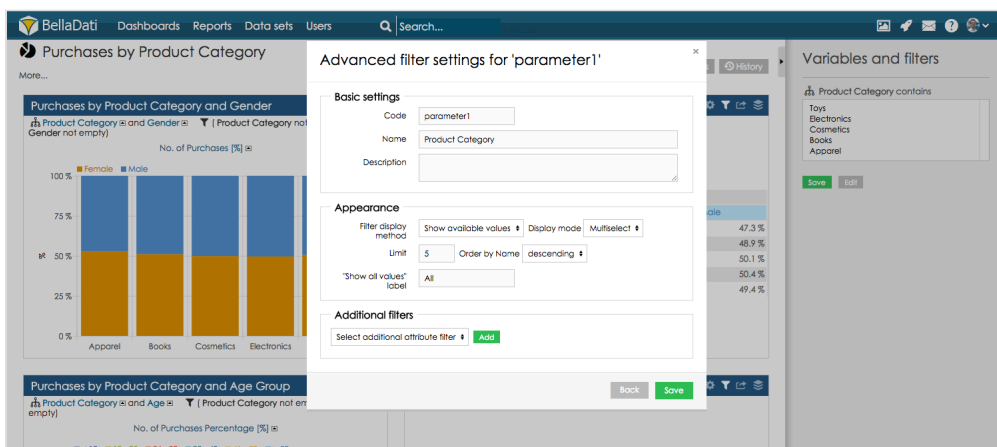


Selecting the component

Select name for your component.

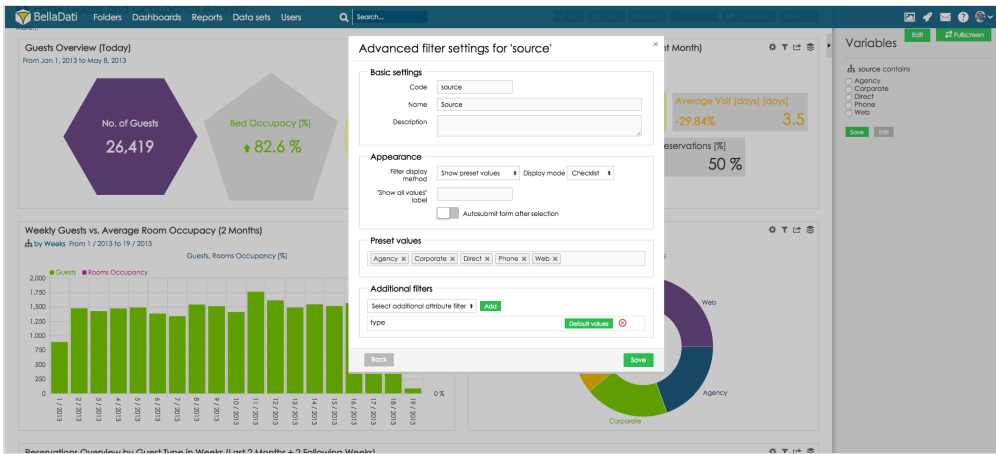
Section Appearance

- **Filter display method:** choose option *Show preset values* to display only selected values (select them in section **Preset values**). Select option **Show available values** to display selected number of available values.
- **Display method** - select which filter component will be used (multiselect, select, checklist or radio)
- **Limit** - set how many values will be displayed in the filter
- **Order by Name** - select how values will be ordered (ascending or descending)
- **"Show all values" label** - set value which will be displayed as option for all of the values
- **Autosubmit form after selection** - will confirm every option user selects



Multi structure filter

Select additional attribute to create multi structure filter. This additional attribute is displayed and allowed to select when main filter attribute is selected (only available values are displayed). For each of additional filters can be also selected Default values.



Passing Parameters in URL



It is recommended to get familiar with [Report Variables](#).

On this page:

- [Basic information](#)
- [Passing parameters in URL](#)
- [Passing parameter in embed view](#)
- [Passing parameter in dashboard](#)

Basic information

Variables can be used in many ways in BellaDati. In some cases it might be required to open report or dashboard with values of variable already set. When only one value is needed, default value of variable can be used. If the value needs to be set dynamically, passing parameters in URL is the best option.

Passing parameters in URL

This example shows how to add variable to indicator formula and then set its value through URL.

1. Create new report
2. Open Variables and filter panel
3. Add new number variable identified as "var" and click on Save.

Variables and filters

Add new variable

Of type identified as

☐ Automatically open variables tab when opening report

4. Add new table
 - a. Add formula indicator - see example formula below. Then click on Save.

'New indicator' - Indicator setting

Basic settings

New indicator

Unit Format Rounding ☐ Exclude from aggregation

☐ Replace empty values

Show help

Advanced settings

Math functions | Indicator | Aggregation suffix | Date variables | Count | Statistical functions | Context change | Variable

Attributes codes | Cross reference | String functions | Other functions | Existing report formulas

1 M_PRICE*@var //indicator Price * Variable "var"

Numeric transformation

- b. Add drill-down and save the table and also the report. Since default value of variable is not set, the indicator values should be empty.
5. Add some value to variable and click on Save. Correct table should be now displayed.

The screenshot shows the BellaDati interface. On the left, a table titled 'New table' displays data for various countries. On the right, the 'Variables and filters' sidebar is visible, featuring a 'var' input field containing the number '10'. A red circle highlights the 'Save' button below the input field.

Country	New indicator
China	42,000,000
France	93,980,000
Germany	79,400,000
Israel	60,000,000
Japan	94,000,000
Russia	62,500,000
South Korea	88,000,000
United Kingdom	97,500,000
United States	93,800,000

6. Value of this variable can be set by opening this URL: <https://URL/bi/report/detail/IDofReport?var=20>
 - a. URL - URL (IP) of your BellaDati installation
 - b. IDofReport - numerical ID of report (for example 279)

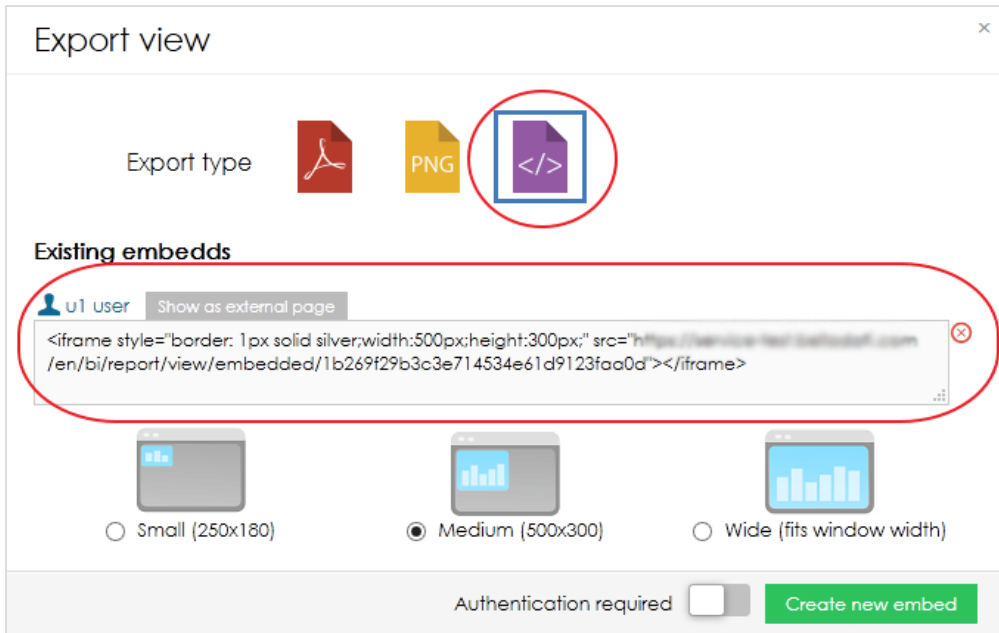
Passing parameter in embed view

It is also possible to pass a value of variable into embed view (iframe).

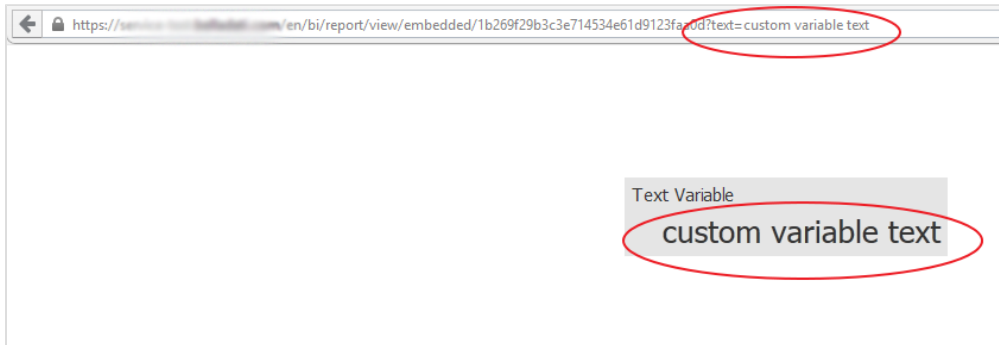
1. The scenario is the same as in report. In this example a text variable "text" is used in KPI label.

The screenshot shows the 'Text Variable' - Indicator setting dialog box. It has two main sections: 'Basic settings' and 'Advanced settings'. The 'Basic settings' section includes a 'Text Variable' input field, 'Unit', 'Format', 'Rounding', and 'Classic (half up)' options. The 'Advanced settings' section includes a list of functions (Math functions, Indicator, Aggregation suffix, Date variables, Count, Statistical functions, Context change, Variable) and a 'Numeric transformation' dropdown. The 'Save' button is highlighted in green at the bottom right.

2. Create new embed view by clicking on export icon and selecting embed code.

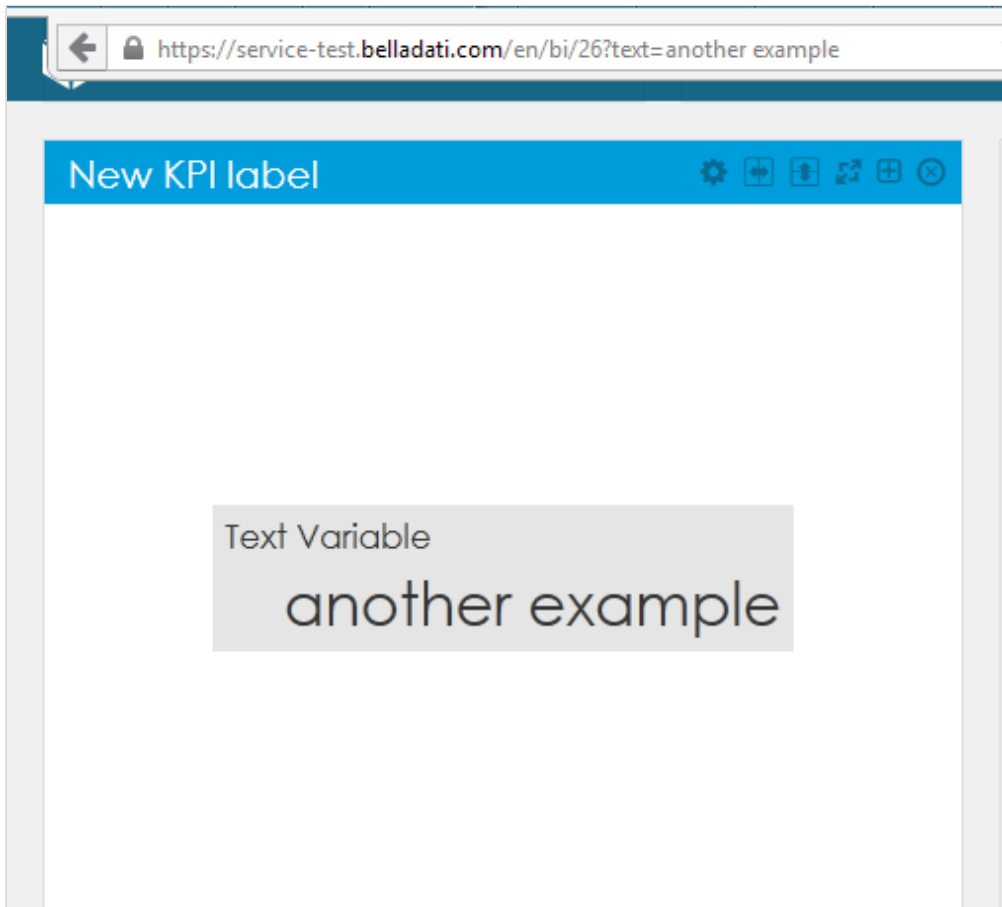


3. Click on "Show as external page".
4. Add code "text=custom text variable" at the end of the URL and click on Enter. The value should be now displayed.



Passing parameter in dashboard

1. Add view with variable in dashboard. The same view with variable "text" is used in this example.
2. Add code "text=another example" at the end of the URL and click on Enter. The value should be now displayed.



Defining Drill-through URL



Make sure to get familiar with applying **Drill-downs** before proceeding with this section.

Drill-through via attribute members

BellaDati allows you to mask attribute's member with custom URLs. This feature is useful to:

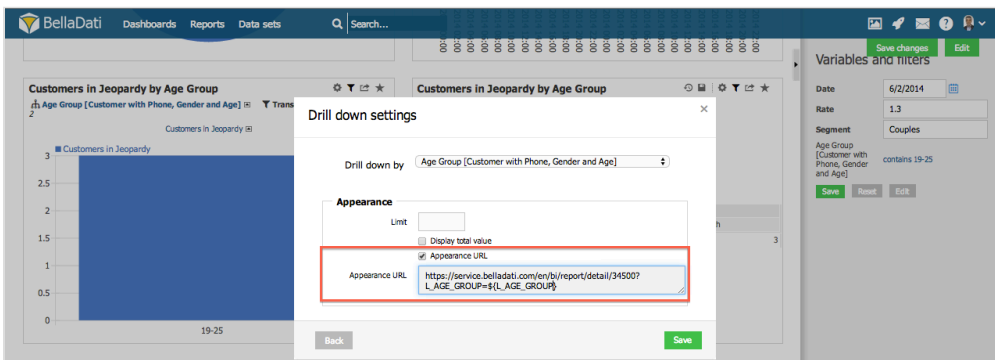
- create **Drill-throughs** - redirection from master report to detailed one based on clicked member.
- redirect to other report or resources.

Masking members

1. Open drill-down settings
 - a. In tables: Go to **Table settings** and click on existing attribute or select **Add drill-down path**.
 - b. In charts: Go to **Chart settings** and click on **Drill down path**. In charts with a drill-down on the X/Y axis, click **Advanced settings**.
 - c. In maps: Go to Map settings, click **Drill-down and map properties**.
2. Check **Show as URL link**
3. Define custom **URL**



- In the URL, you can refer to the current drill-down value using `${}`.
- To refer to other drill-down attribute values, use `${L_ATTRIBUTE_NAME}`.
- You can create report links using `${reportLink(report_id)}`



Drill-through via indicator

BellaDati allows you to mask indicators with custom URLs. This feature is useful to:

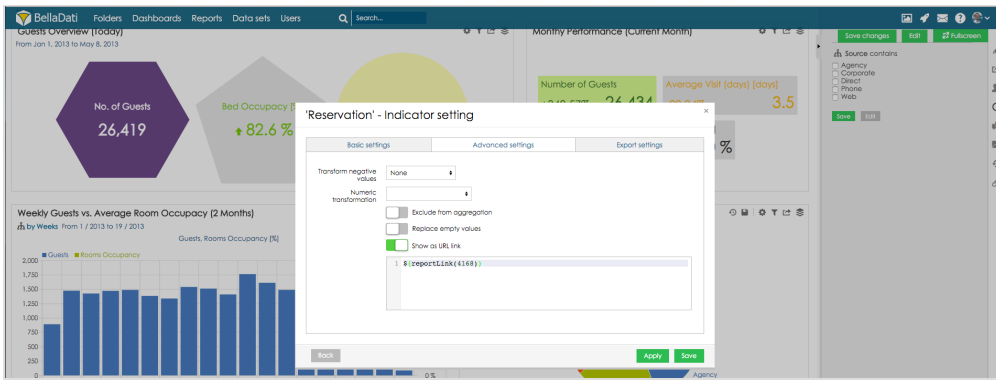
- create **Drill-throughs** - redirection from master report to detailed one based on clicked indicator.
- redirect to other report or resources.

Masking members

1. Go to indicator settings
2. Select tab **Advanced settings**
3. Check **Show as URL link**
4. Define custom **URL**



- You can create report links using `${reportLink(report_id)}`
- To refer to current indicator, use `${indicatorCode}`



Adding Comments and Attachments



You need to be in [view mode](#) in order to add comments and attachments.

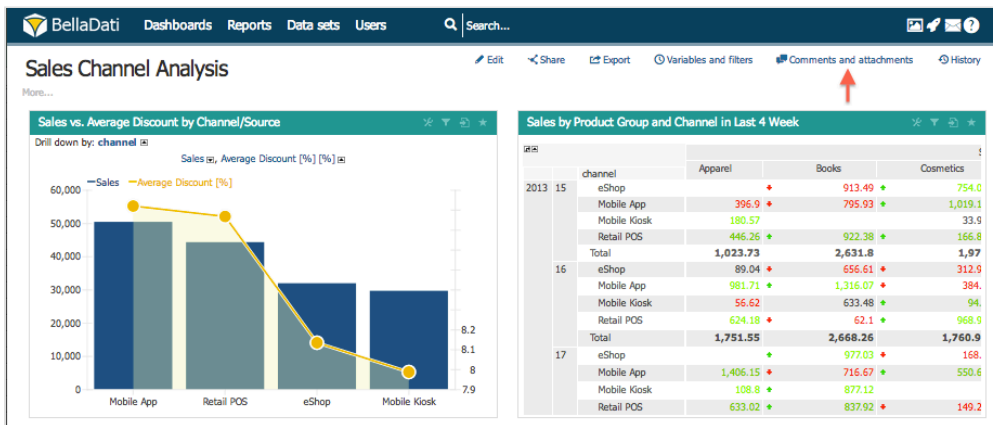
Select **Comments and Attachments** from the report toolbox list in the upper right corner to open *Attachments* panel.

Attachments dialog allows you to add:

- **Attachments:** Click on **Add attachment** to open Attachments section.
 - Click **Browse** to select desired file (double click needed if you are using Internet Explorer 8).
 - Click **Add attachment** to save the attachment.
- **Comments:** Type your **Comment** into text area and submit it.



Maximum attachment size is 20MB.

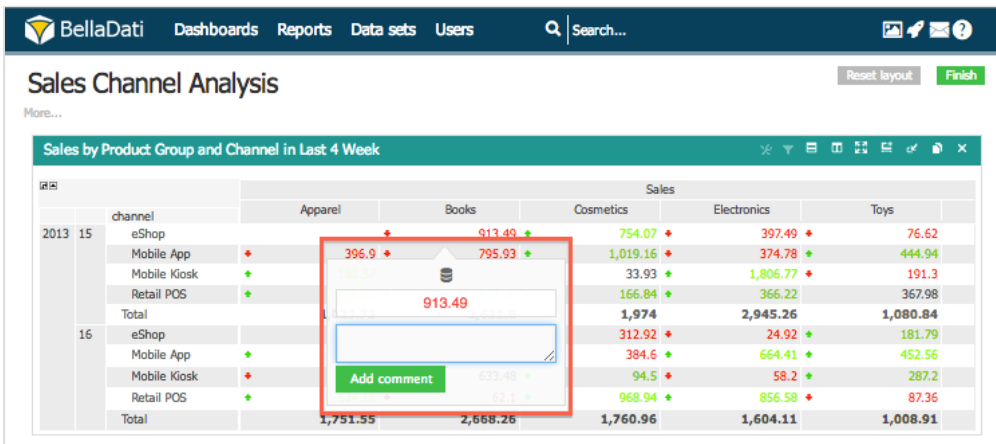


Adding Comments in Table Cells

Hover over desired table cell and select **comment** to attach comment to the data.



Red triangle marks cells with attached comments. Hover over the cell to see the **bubble** with the comment.

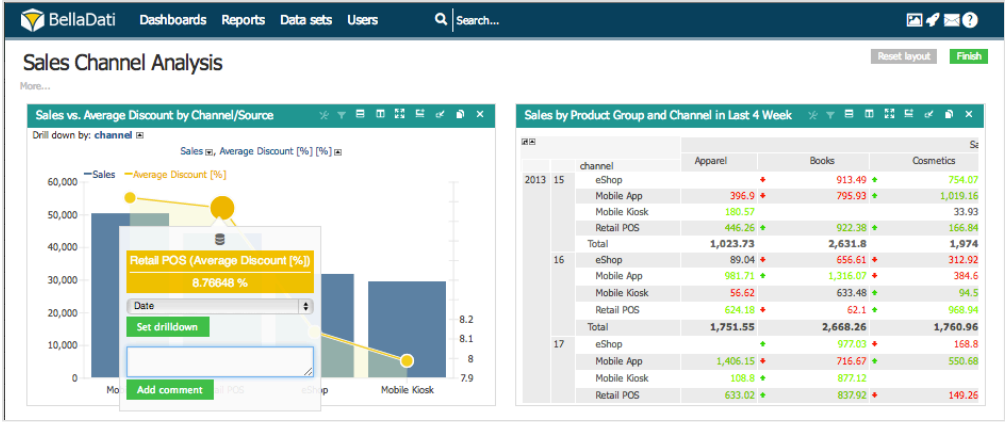


Adding Comments in Charts

Hover and click on desired values in the chart. The *value settings popup* will be opened. Type comments and hit **Add**.

Comments will be recorded:

1. on the **right chat sidebar**
2. on the **particular value**. Hover over it to see all related comments.




Publishing to Dashboard

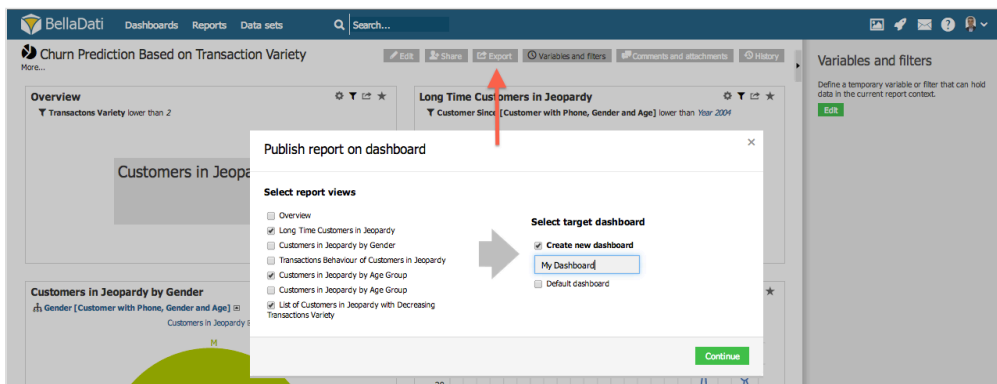
Publish allows you to pin the **Report/View** to selected **Dashboard**.

You can access **Publish** option from the toolbox in the upper right corner of the view or by selecting **Add to dashboard** from **Export** report option.

From *Publish report to dashboard* dialog you can:

- Select more **Views** to be saved to dashboard.
- Specify destination from list of all available **Dashboards**.
- Enter **Name** of the new dashboard.

 BellaDati will preselect desired object when publishing from view.



Navigate to destination **Dashboard** in order to observe published views.


Searching and Filtering Reports

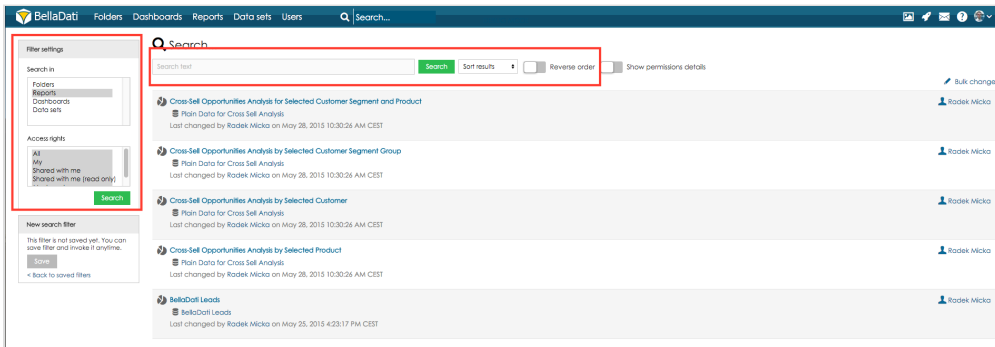
BellaDati allows you to **search** for specific report and **filter** reports by various criteria.

Click on **Reports** in the top toolbar to enter the *Search reports* window. By default, BellaDati will list all available reports.

Left panel of the *Search* window includes **Filter settings** panel. Panel allows you to:

- Insert **search text**.
- Define **sort order** of the results.
- Request **reverse sorting**.
- Restrict search by **object type**.
- Restrict search by **access rights**.

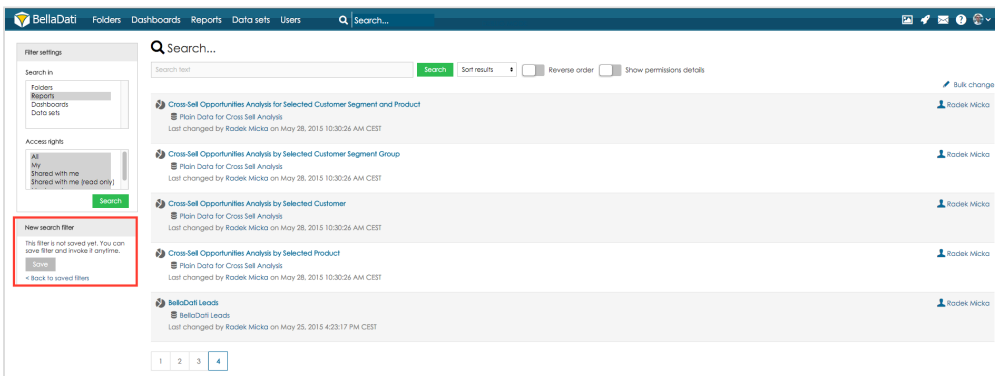
 Hold **Shift** key to select multiple **object types** or **access rights**.



Saving Filter


You can save defined **filter** for future usage or for its displaying in [filter dashlet](#).

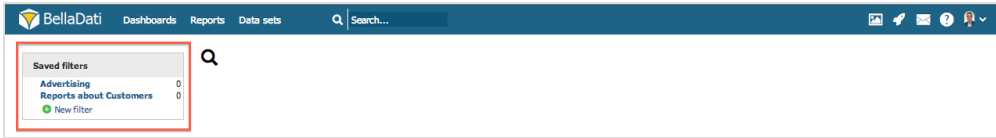
Click on **Save** button in the upper part of the **Filter settings** panel to save current settings.



Click on **Back to saved filters** link to view all stored filters. *Saved filters* windows allows you to:

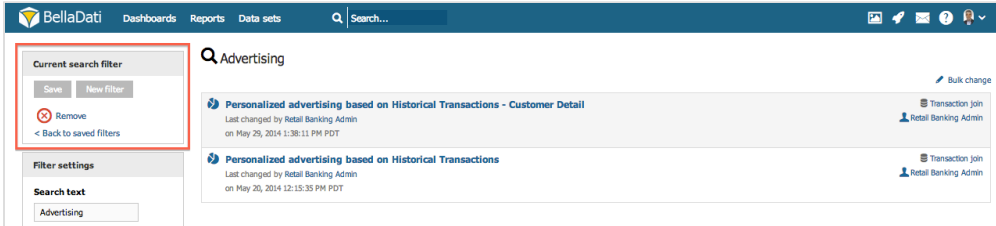
- Display **filters**
- Create **new filter**.

 Number next to the filter name shows how many objects meet the entered condition.



Removing Filter

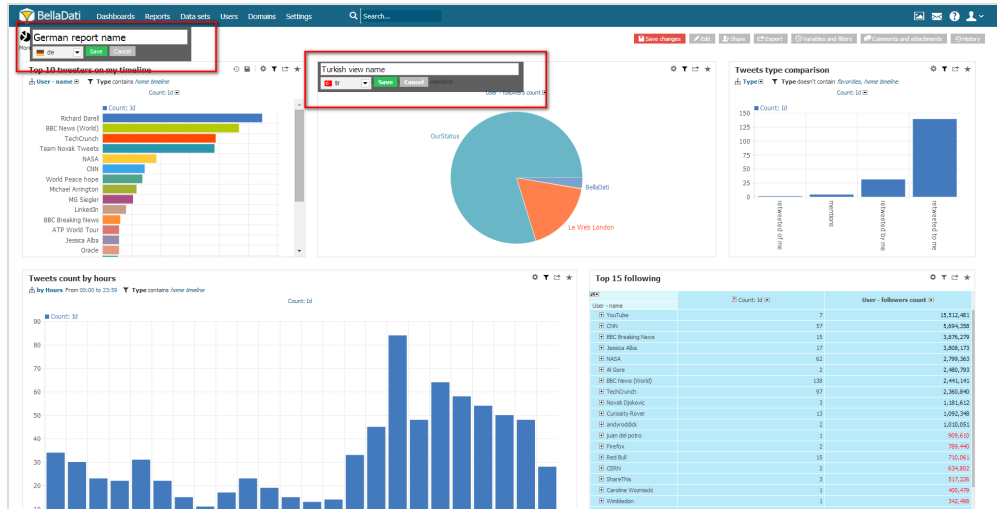
From *Saved filters* windows click on the filter name you want to remove. Click on the **Remove** icon to erase the filter.




Translating Reports

With multilingual names support you can make your reports and views available in various languages.

You can just **click** on the desired report or view **name** and choose the language you want to translate the name to.



 Names will be displayed in the signed user's language. Language can be set in user profile, eventually in the domain detail.

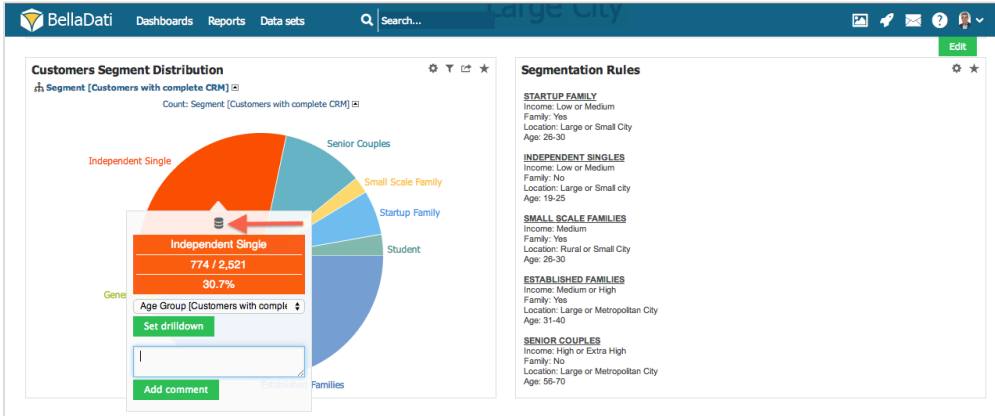
See also

- [Translating Reports](#)
- [Translating Indicators](#)
- [Translating Attributes and Members](#)

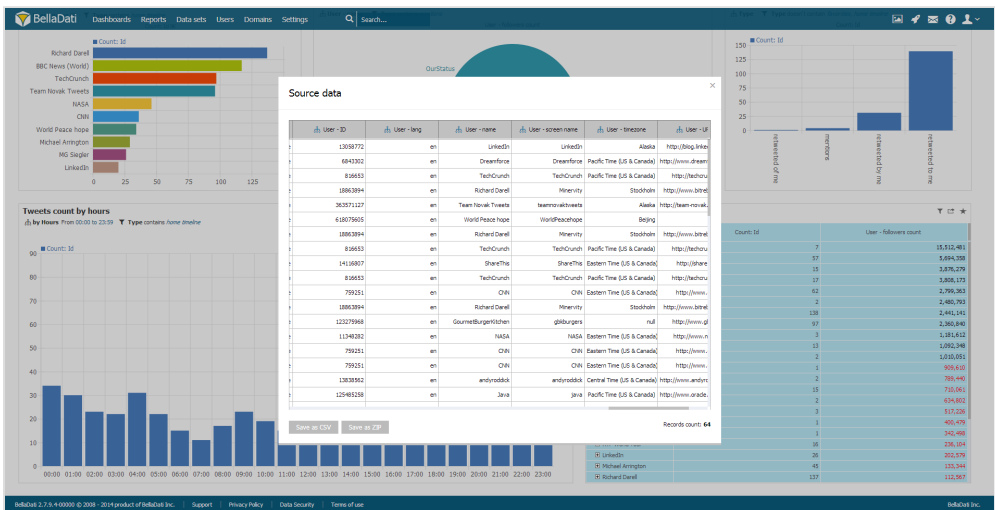
Displaying Source Data

Displaying of source data is available in **charts, tables** and **maps**.

You can display all data which composes specific chart item in *Source data* dialog. Click on the desired section and select **Source data icon** to open the dialog.



And the source data which is the value composed of are displayed in the popup window.

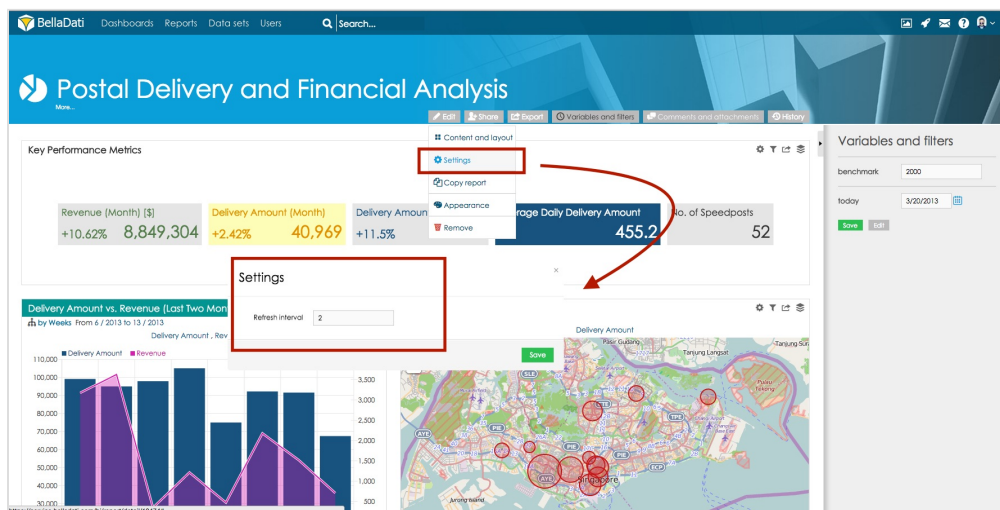


Exporting source data

You can export the displayed source data into a CSV or ZIP file.

Auto update interval

User can setup autoupdate interval for reports. Report is connected to dataset. Once the data in dataset are update, report will be updated according the Update interval settings. Update interval settings is in seconds. Only views (charts, tables) in which new data are pushed are refreshed. Other views are kept unchanged. Follow steps below to setup update interval from within the Report Menu Edit ----> Settings ----> Insert desired interval in seconds




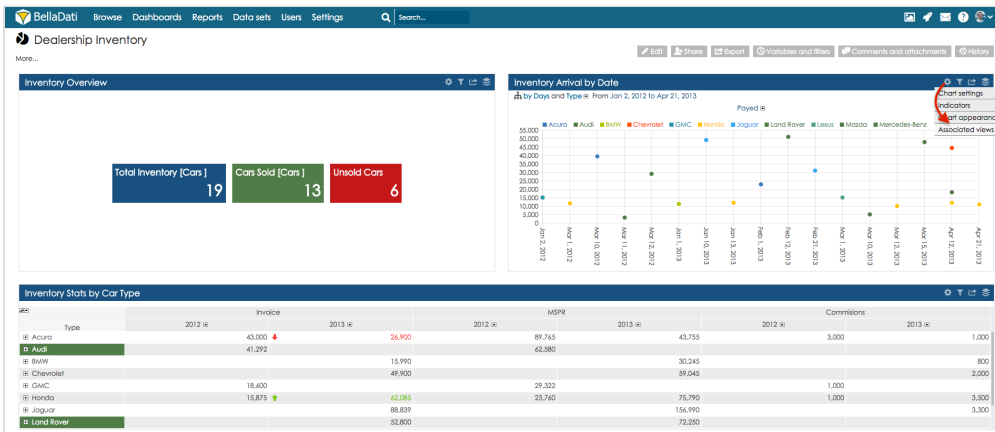
Associated views

Associated views serve to create association between selected views. If one view (master) is filtered, other views (slaves) are filtered with the same values.


Creating associations

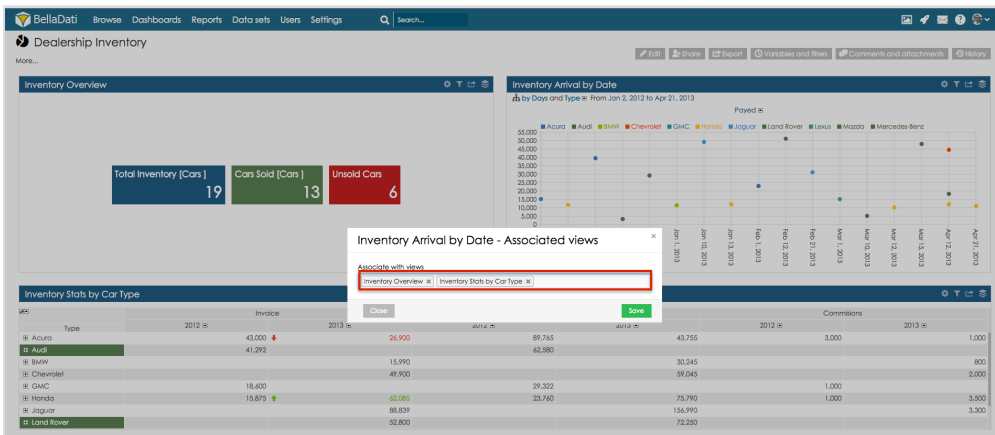
First need to be views associated with master view. Hover your mouse over settings button and select option *Associated views*.

 Master view can be only table or chart.



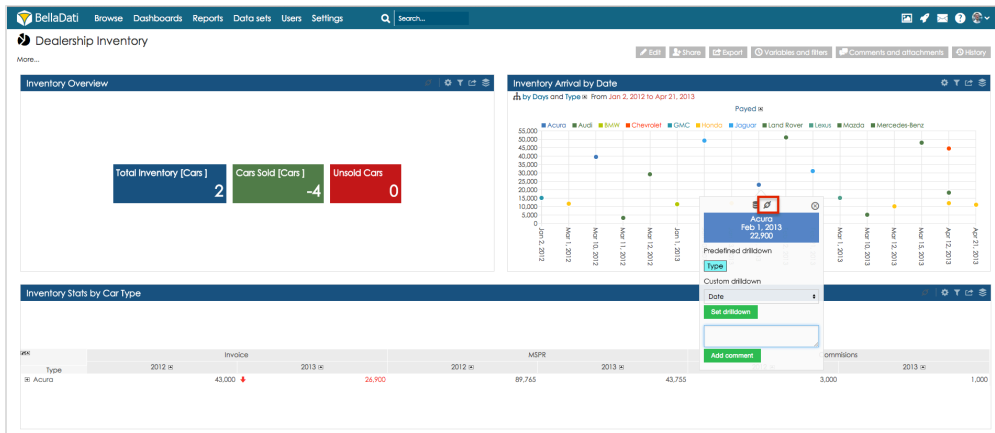
In the field *Associate with views* select the slave views which will be associated with this master view.

 One slave view can have association only with one master view.

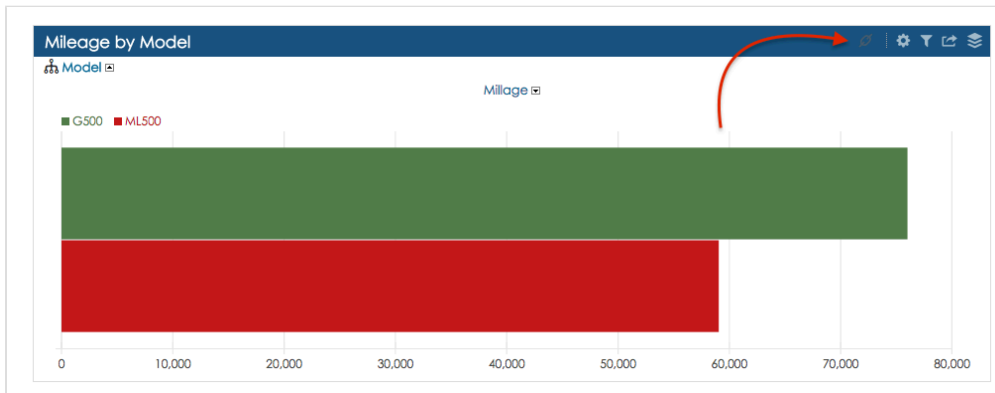


Create association

Now hit a click on a value in master view and click on association icon. All associated views will according to selected attribute member.



Views which are filtered by associated (master) view are marked with association icon:

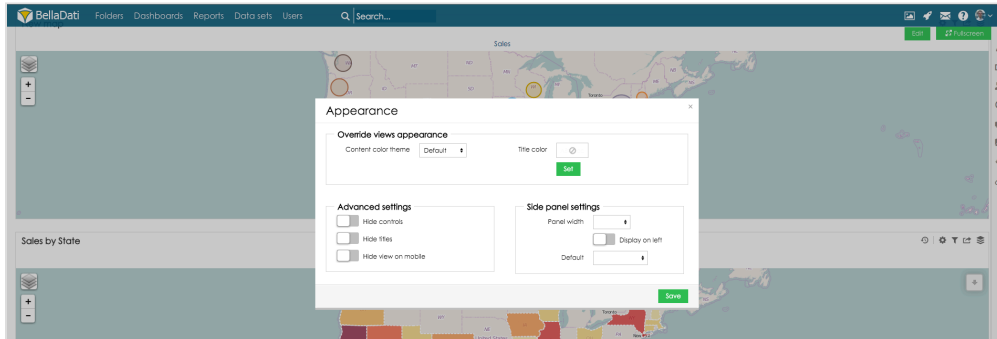


Clicking on this icon will remove the filter from the view.

 You can watch Associated Views tutorial in Video Tutorial [section](#).

Report appearance

To access report *Appearance* dialogue window, hover your mouse over button *Edit* and select option *Appearance*.



This dialogue window allows you to:

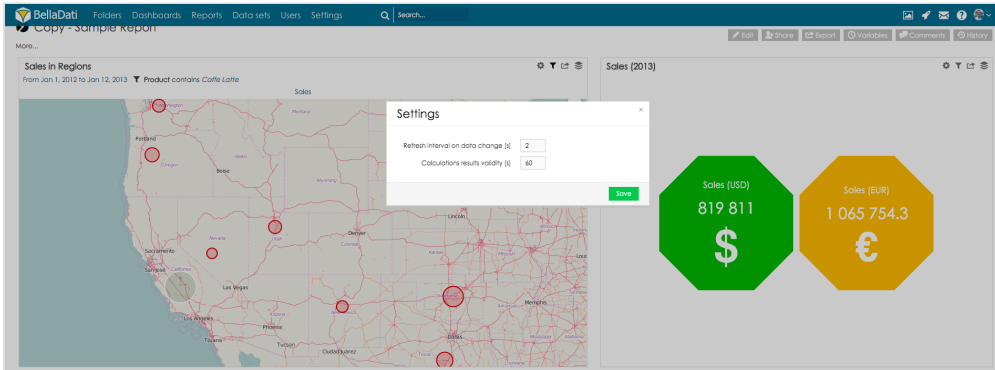
- Change **Content default color**
- Set **Title color** of all views
- Select, if view controls should be displayed (option **Hide controls**)
- Select, if view titles will be displayed (option **Display titles**)
- Select, if the report should not be displayed in **mobile** application
- Set side panel settings

Side panel settings allow you to:

- Set panel width
- Select side on which panel will be displayed (right or left)
- Select default tab - if any option is selected, side panel with this tab will be automatically opened when user opens the report

Report settings

To access report *Settings* dialogue window, hover your mouse over button *Edit* and select option *Settings*.



Report settings dialogue window allows you to:


- set **Refresh interval on data change**
- set **Calculations results validity**

Refresh interval on data change parameter serves to set how often will be changes in the source data set checked. When change in source data is detected, report is automatically refreshed. Default value is set to 2 seconds.

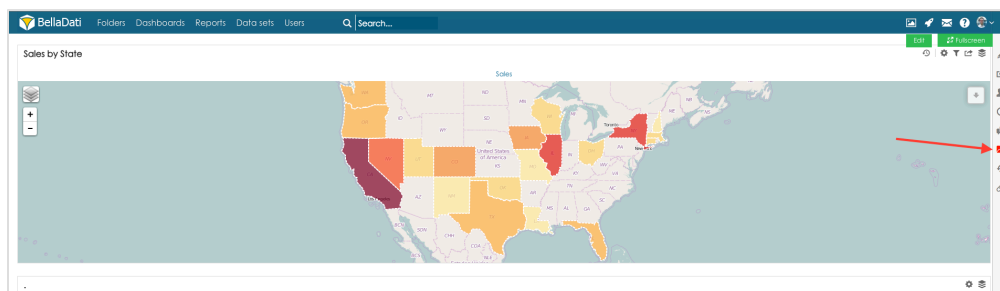
Calculations result validity parameter is dedicated to speedup reports re-loading time. However data are being calculated in-memory and some of them are already cached on database subsystem layer, there is quite huge CPU overhead while computing formula.

There are several reasons why it is not used automatically everytime - each formula might contain some parameters evaluated in runtime (cross values, date/time modifiers etc), so we cannot cache just by the formula context (date/time/drilldown/filter) and formula definition itself. On the other hand, formula calculation cache automatically expires after desired period of time [s] or on the data or context change. Default value is set to 60 seconds.

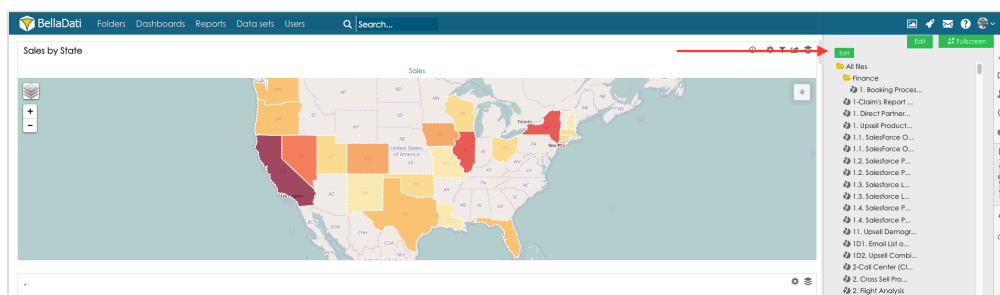
Report navigation

 You need to be in **view mode** in order to access report navigation.

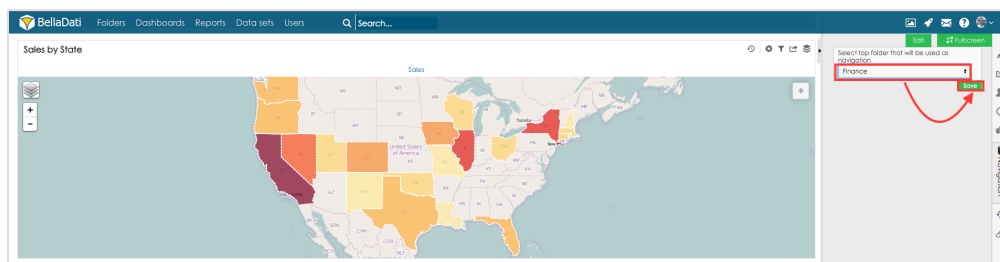
Select option navigation from the control panel.



All folders and reports are displayed by default. In case you would like to display only reports from selected folder click *Edit*.



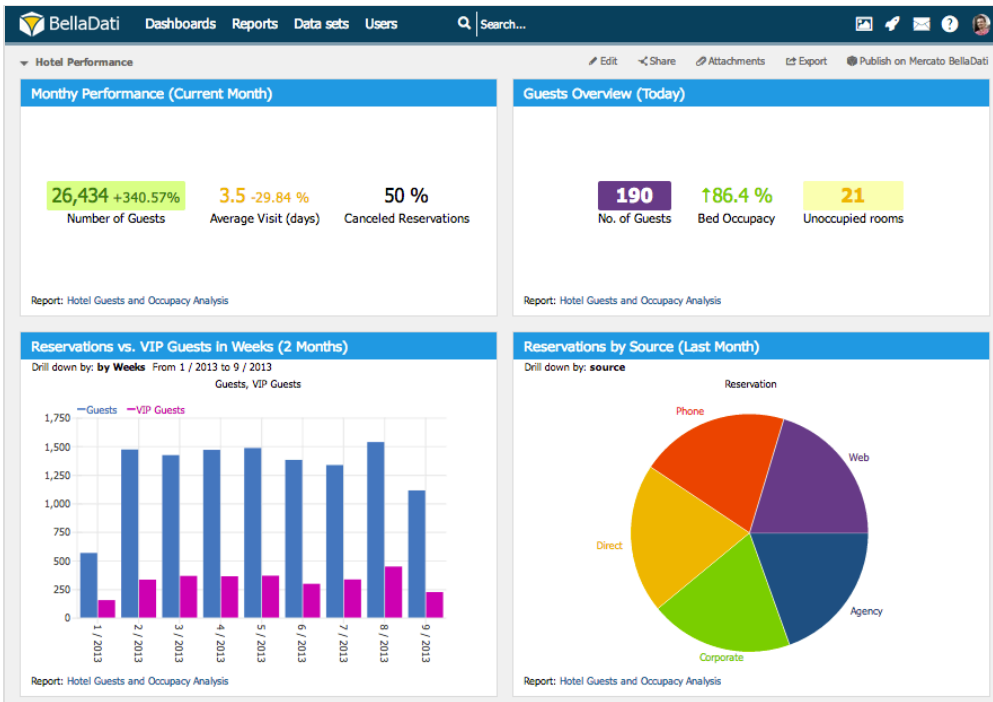
Select desired folder and save the settings.



Dashboards

Overview

Dashboards are the entry point of BellaDati Business Intelligence solution. They typically comprise of the most important charts and tables pulled out from the more detailed **reports**. Therefore, they are primary designed for **managers**, who need brief overview of company's actual performance.



Dashlets

Every **dashboard** is composed of several **dashlets**. **Dashlet** is a piece of content, including either report's views or navigation items. **Dashboard** can be customized by adding **dashlets** and adjusting their visualization.

Dashboards can be enriched with arbitrary attachments as well as shared for public via web.

Use cases

Dashboards allow to visually compare data from different reports and therefore data sources. Consider using [Reports](#) if you are an analyst and need to perform detailed data analyses.

Dashboards

Reports





Dashboards are determined for uses who prefer consuming prepared insights. They usually combine existing **views** and **navigation items** to create their own customized workspaces. **Dashboards** are not designed for direct work or analyses of data. Consumed content is prepared in **reports** by **analysts**.

Typical user includes:

- Executives
- Managers

To learn more about **Dashboards** proceed in this chapter.

Reports are determined for users who work with data, prepare reports and execute analyses. Users usually select **Indicators**, define **Drill downs** and visualize them in appropriate **View types**. Work with **reports** requires deeper knowledge of data. Created content can be consumed in **dashboards** by **managers**.

Typical user includes:

- Analysts
- Specialized employees

To learn more about **Reports** - continue by [Reports](#)

Following actions are supported on dashboards:

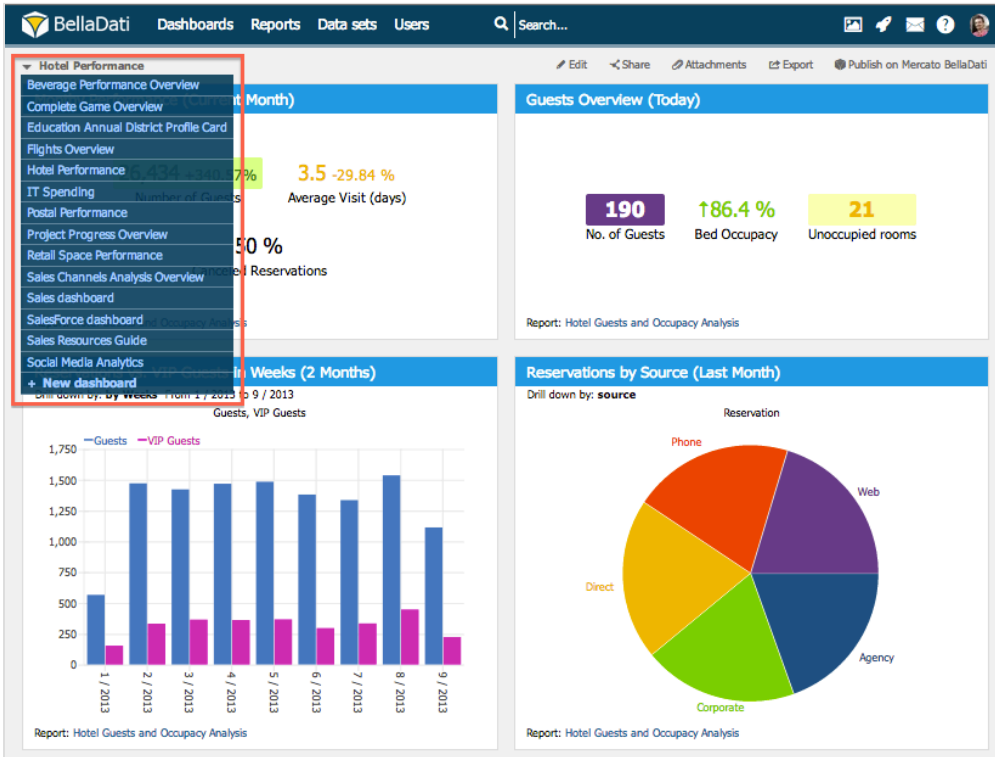
- [Creating Dashboard](#)
- [Managing Dashboard Layout](#)
- [Creating Dashlet](#)
 - [Adding View](#)
 - [Adding Navigation](#)
 - [Adding Filter](#)
 - [Adding Other Content](#)
- [Sharing Dashboard](#)
- [Adding Attachment](#)


Creating Dashboard

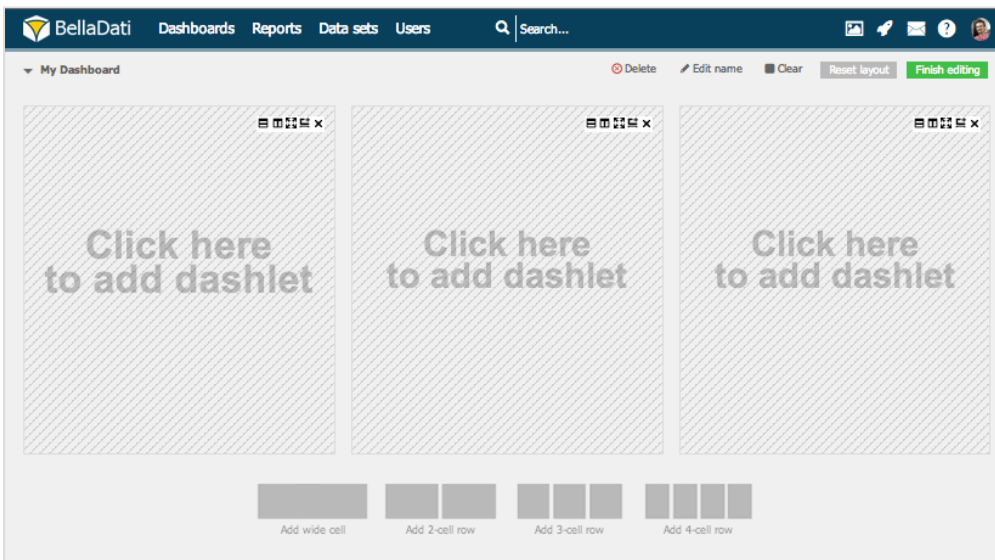
Point to the **Dashboard** name in upper left corner of **Dashboard** window and select **New Dashboard** from the list.

1. Enter name of the new dashboard.

New **Dashboard** will be created.



 Dashboard will be automatically turned into [edit](#) mode.



Continue by [Managing Dashboard Layout](#) and [Creating Dashlet](#) to populate the **Dashboard**.

Edit mode



You can switch to **Edit** mode by clicking on "Edit" in top dashboard menu.

Edit mode allows you to:

- [Manage Dashboard Layout](#)
- [Adding Dashlets](#)
- **Exit edit mode:** Click on **Finish Editing** to enter dashboard **View** mode.
- **Edit Dashboard Name:** Click on **Edit name** to change the name of current dashboard.
- **Clear Dashboard:** Click on **Clear** to remove all dashlets from the dashboard.
- **Remove Dashboard:** click on **Remove** to erase current dashboard.

Managing Dashboard Layout



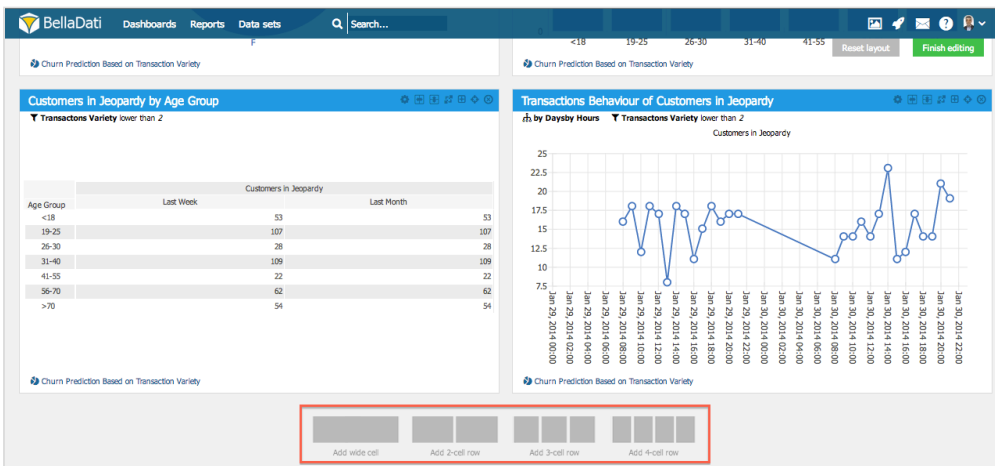
You need to be in [edit mode](#) in order to manage dashboard layout. Click on "Edit" in top dashboard menu to activate edit mode.

You can modify **Dashboard** layout to your needs by:

- adding new **rows**.
- changing **size** of existing dashlets.
- **manipulating** dashlet.

Adding rows

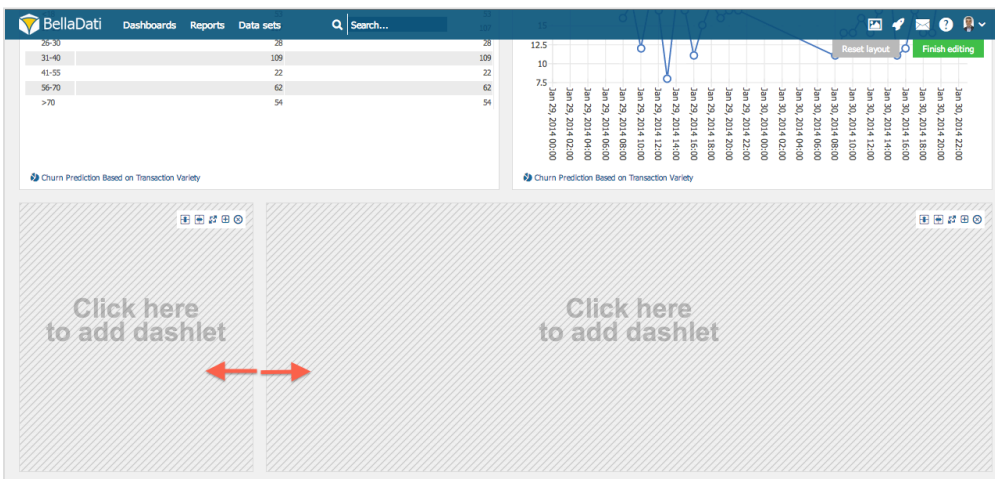
Scroll to the bottom of the **Dashboard** and select number of columns you require in the new row.



Changing size

Point your mouse among two dashlets and move with bar to change their size. You can change:

- dashlet **width**
- dashlet **height**

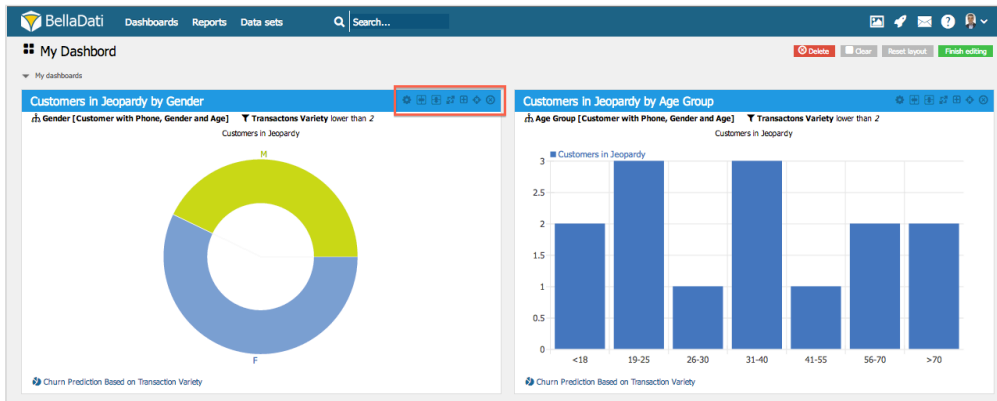


Manipulating dashlet

Point your mouse to manipulation buttons in upper right corner of the **dashlet**. They allow you to:

- **move** dashlet

- **split** dashlet **horizontally**
- **split** dashlet **vertically**
- **insert new row**
- **enlarge** dashlet
- **remove** dashlet



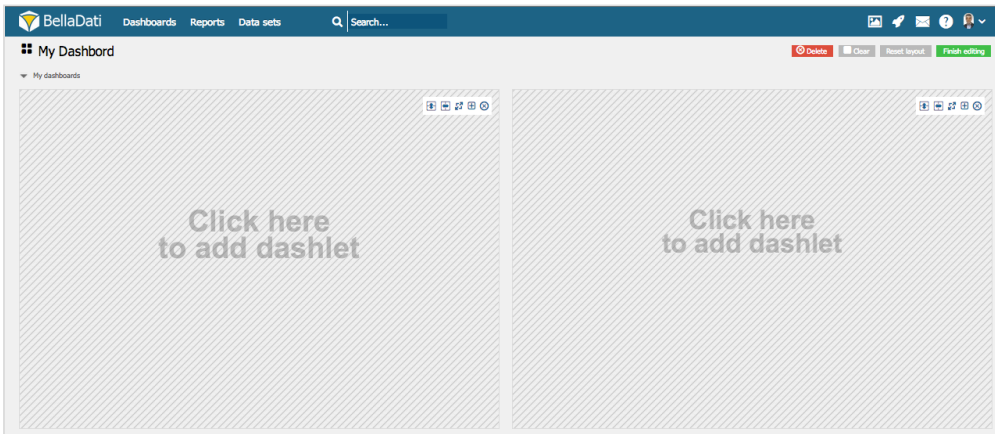
Creating Dashlet



You need to be in [edit mode](#) in order to create new dashlet. Click on "Edit" in top dashboard menu to activate edit mode.

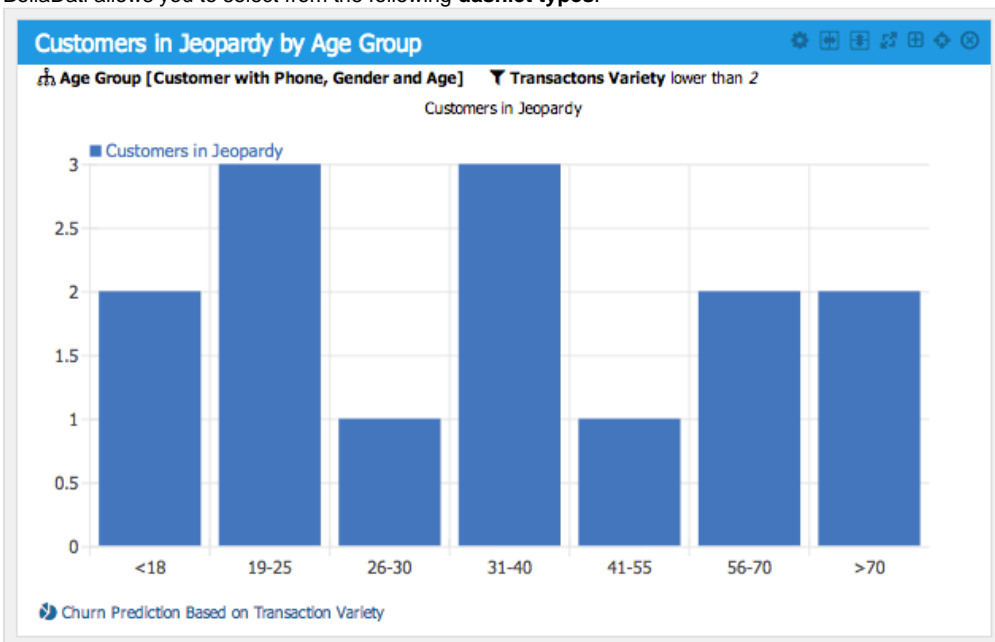
Dashlet is a peice of specific content positioned on the **Dashboard**.

To add a new **Dashlet** hover over free place and click on **Add Dashlet**. The *Insert dashlet* dialog box will appear.



Dashlet types

BellaDati allows you to select from the following **dashlet types**:



View

View dashlet allows you to display view from available **reports**. To find a desired view you can:

- **Search** by name.
- **Filter** by report.
- **Browse** views.

To learn more about **View** dashlets continue by [Adding View](#).

Owned reports	
Name	
Paired Transactions for Promotions to be Sent Last changed by Retail Banking Admin on May 15, 2014 1:50:17 AM PDT	Paired Promotions join Retail Banking Admin
Post Location-based Promotion Evaluation and Analysis Last changed by Retail Banking Admin on May 13, 2014 6:28:38 AM PDT	Paired Promotions join Retail Banking Admin
Visitors Demographics Last changed by Retail Banking Admin on May 13, 2014 4:02:35 AM PDT	Branch Visits join Retail Banking Admin
Operational Efficiency for Selected Branch Last changed by Retail Banking Admin on May 15, 2014 3:57:12 AM PDT	Branch Visits join Retail Banking Admin
Operational Efficiency Last changed by Retail Banking Admin on May 13, 2014 5:13:11 AM PDT	Branch Visits join Retail Banking Admin

Navigation

Navigation dashlet allows you to display dashboard navigation.

BellaDati offers following navigation types:

- **Most visited**
- **Owner reports**
- **Shared reports**

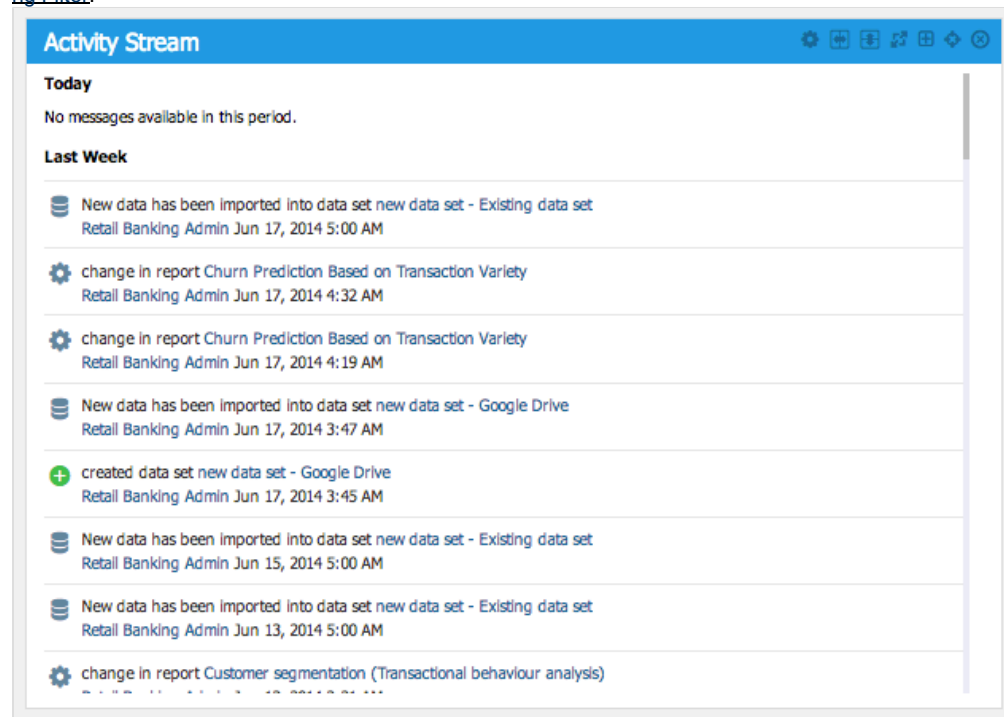
To learn more about **Navigation** dashlet continue by [Adding Navigation](#).

Reports about Customers	
Customer segmentation (Transactional behaviour analysis) Last changed by Retail Banking Admin on June 13, 2014 2:21:37 AM PDT	Product Ownage join Retail Banking Admin
Copy - Customer segmentation (Transactional behaviour analysis) Last changed by Retail Banking Admin on June 13, 2014 1:51:26 AM PDT	Product Ownage join Retail Banking Admin
Personalized advertising based on Historical Transactions - Customer Detail Last changed by Retail Banking Admin on May 29, 2014 1:38:11 PM PDT	Transaction join Retail Banking Admin
Customer Engagement - Shared Wallet Last changed by Retail Banking Admin on May 29, 2014 3:50:02 AM PDT	Shared Wallet Survey join Retail Banking Admin
Cross-Sell Opportunities Analysis for Selected Customer Segment and Product Last changed by Retail Banking Admin on May 20, 2014 12:20:17 PM PDT	Plain Data for Cross Sell Analysis Retail Banking Admin
Cross-Sell Opportunities Analysis by Selected	Plain Data for Cross Sell Analysis

Filter

Filter dashlet allows you to display filter window. Filter window shows reports meeting predefined search conditions.

To learn more about **Filter** dashlet continue by [Adding Filter](#).



Other content

Other content dashlet allows you to display arbitrary content.

You can add:

- **News**
- **Custom content**
- **RSS**
- **Help and tutorials**
- **Begin with BellaDati**

To learn more about **Other content** dashlet continue by [Adding Other Content](#).

Adding View



You need to be in **edit mode** in order to add new view. Click on "Edit" in top dashboard menu to activate edit mode.

To add a new **View**, hover over free place and click on **Add dashlet**. The *Insert dashlet* dialog box will appear.

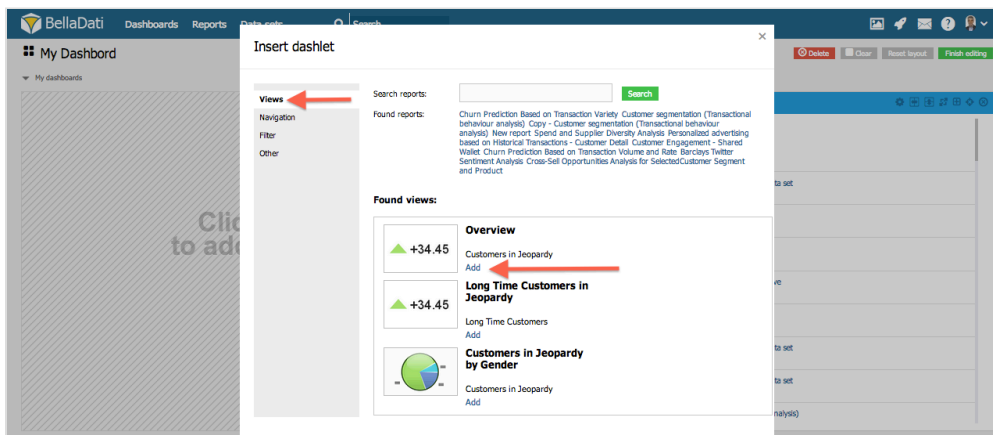
1. Select **View** from the left navigation panel.
2. **Browse** for the desired view.
3. Click **Add** to append the view to the dashboard.

In case of many view, BellaDati allows you to:

1. **Search** for target report.
2. **Filter** target report.



List of views displays **view type**, its **name** and employed **indicators**.



Desired **View** will be appended to the **Dashboard**.

Adding Navigation



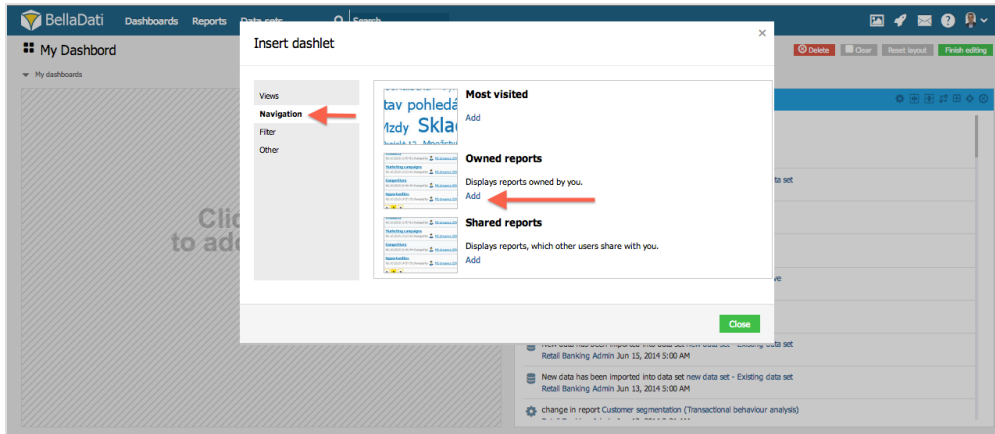
You need to be in [edit mode](#) in order to add navigation. Click on "Edit" in top dashboard menu to activate edit mode.

To add a new **Navigation**, hover over free place and click on **Add dashlet**. The *Insert dashlet* dialog box will appear.

1. Select **Navigation** from the left navigation panel.
2. Click **Add** to append desired navigation type to the dashboard.

BellaDati offers following navigation types:

1. **Most visited**
2. **Owned reports**
3. **Shared reports**



Navigation will be appended to the **Dashboard**.

Adding Filter



You need to be in [edit mode](#) in order to add filter. Click on "Edit" in top dashboard menu to activate edit mode.

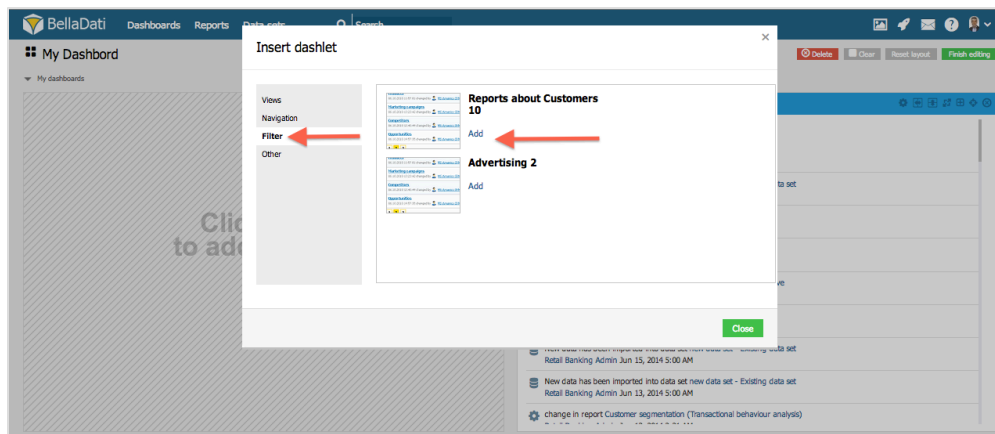
To add a new **Filter**, hover over free place and click on **Add dashlet**. The *Insert dashlet* dialog box will appear.

1. Select **Filter** from the left navigation panel.
2. Click **Add** to append desired filter to the dashboard.

For more information about creating **Report filters** continue by - [Searching and Filtering Reports](#)



Number at the end of the filter name displays **total items** meeting the filter condition.



Filter will be appended to the **Dashboard**.

Adding Other Content



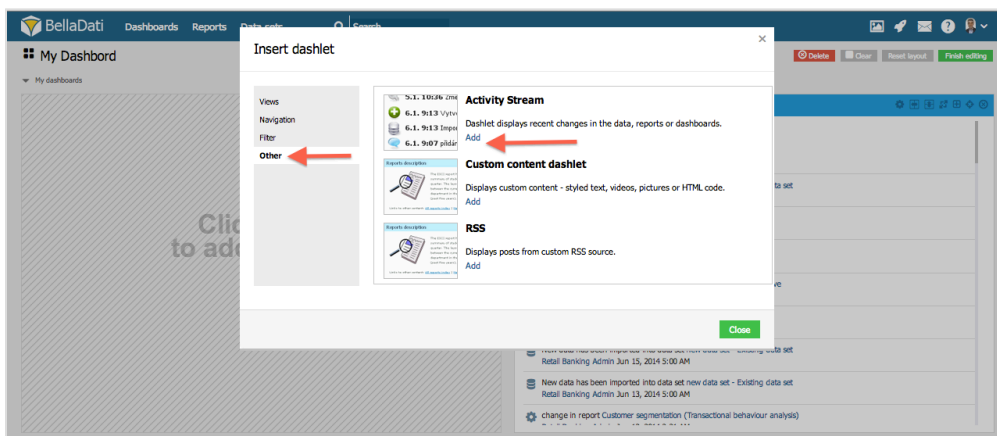
You need to be in [edit mode](#) in order to add content. Click on "Edit" in top dashboard menu to activate edit mode.

To add a new **Content**, hover over free place and click on **Add dashlet**. The *Insert dashlet* dialog box will appear.

1. Select **Other** from the left navigation panel.
2. Click **Add** to append desired content to the dashboard.

BellaDati offers following types of content:

- **Activity Stream**: Dashlet displays recent changes in the data.
- **Custom content**: Displays custom content - styled text, videos, pictures or HTML code.
- **RSS**: Displays posts from custom RSS source.
- **Help and tutorials**.
- **Begin with BellaDati**.



Content will be appended to the **Dashboard**.

Sharing Dashboard

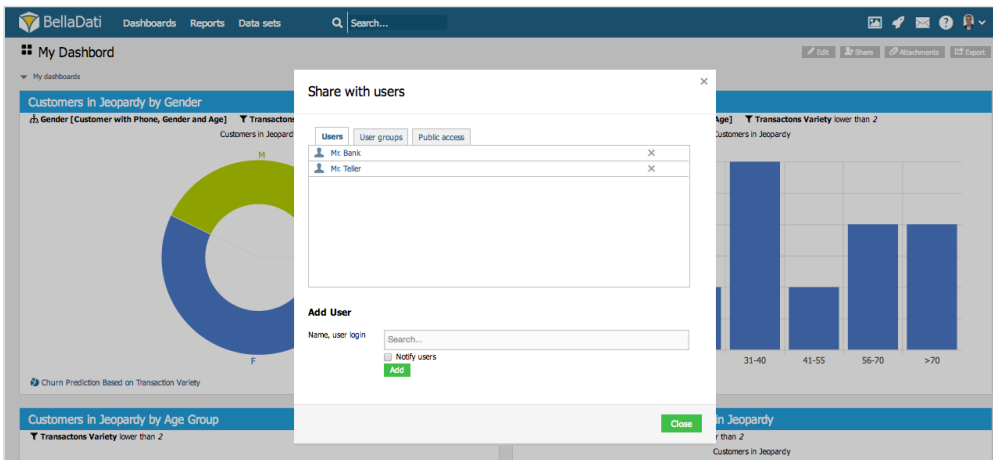


Data set sharing functions are only available for the owners of the particular dashboard.

Click on **Share** in the upper right corner of the **Dashboard** window to enter *Share with users* dialog.

Dashboard sharing functions allows you to perform following actions:

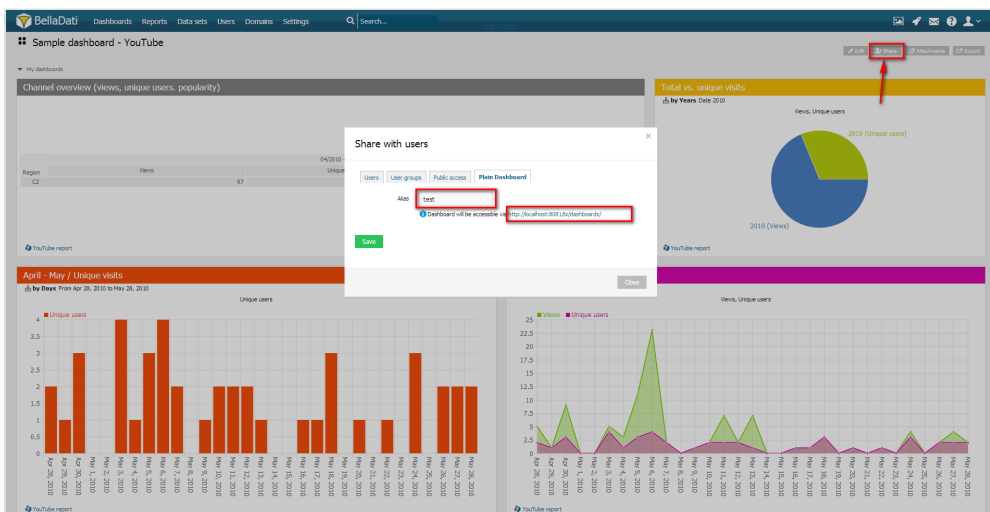
- **Grant access** to the dashboard for selected [users](#) or [user groups](#).
- Optionally **notify** users about granted access to dashboard.



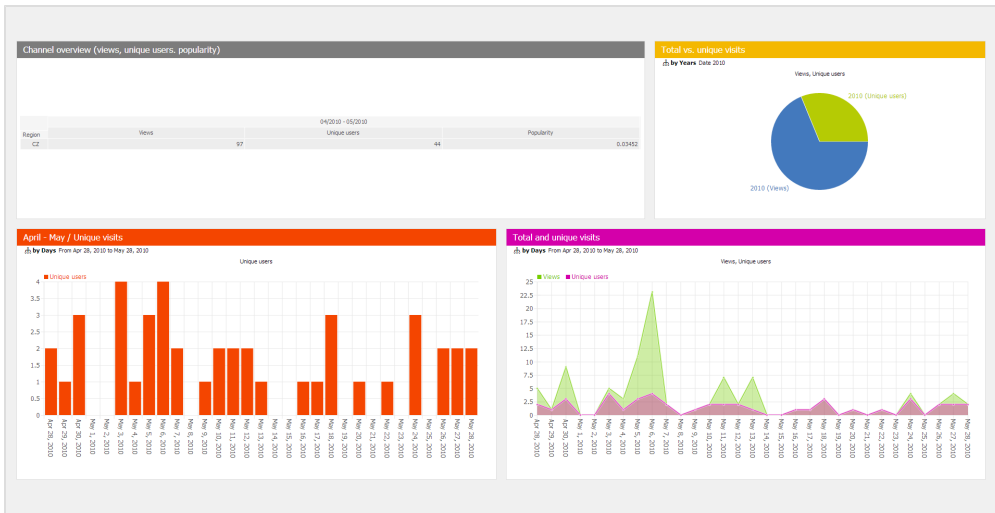
Advanced options

As advanced way to share the dashboard you can use the option to share your dashboard using an alias. Dashboard shared in this way will not display the top menu and links to the reports, but it will check the access rights as original dashboard. All other functions (drill-down, source data, comments, etc) are available.

1. step - set the alias (must be unique in domain)



2. step - get the link and open the dashboard (*relative path is /bi/dashboards/alias*)



Sharing via Sharing console

BellaDati also allows you to share the **Dashboard** via Sharing console. The list contains all the **Dashboards** with users and user groups who can view the dashboard.

 Only **Domain administrator** can access Sharing console.

Sharing console can be found in the Users section. To access Sharing console:

1. Point to the Users in the main menu and click the Sharing console item.
2. Choose Dashboards in the Actions box

To change the sharing settings click the Edit button.

BellaDati Dashboards Reports Data sets Users

Sharing console

Search dashboards...

Name	Owner	Read-only users	Groups for read-only	Public access	Allow	Action
Reference Dashboard	Radek Micko					Edit
Rights Overview	BellaDati BellaDati					Edit
Default dashboard	Radek Micko					Edit
CrossUp Sell Dashboard	Radek Micko2					Edit
Default dashboard	test test					Edit
Retail Banking App	Radek Micko2					Edit
Retail Banking App	BellaDati BellaDati					Edit
Insights into Booking Rates and Process	Radek Micko2					Edit
Sample dashboard - Healthcare	Radek Micko					Edit
MAQ dashboard	release test					Edit
Hospitality Analysis Dashboard	Radek Micko					Edit
Retail Banking App	Radek Micko2					Edit
MAQ dashboard	James Bond					Edit
MAQ dashboard	Bart Simpson					Edit
MAQ dashboard	Homer Simpson					Edit
Default dashboard	BellaDati BellaDati					Edit
Default dashboard	Radek Micko2					Edit
demo dash	Radek Micko					Edit
MAQ dashboard	tester testovic					Edit
Telco Dashboard	Radek Micko2					Edit
Customer Segmentation Insights	Radek Micko2					Edit
Hospitality Analysis Dashboard	Radek Micko					Edit
Revenue and Emission for Illinois Region	Radek Micko2					Edit
Retail Banking App	Radek Micko2					Edit
Performance by Marketing Source	Radek Micko2					Edit

Adding Attachment

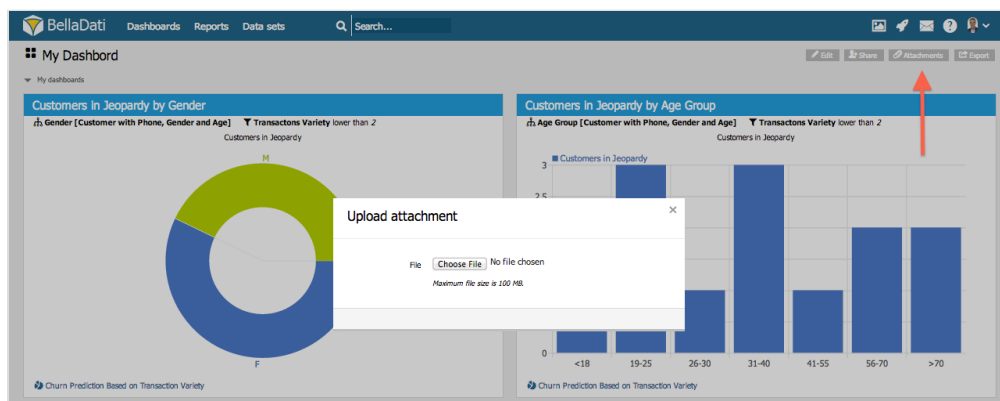
Hover over **Attachment** in the upper right corner of the **dashboard** and select **Upload attachment** to enter *Upload attachment* dialog.

In the dialog you can:

- Click **Browse** for the desired attachment (double click needed if you are using Internet Explorer 8).
- Click **Create** to upload the attachment.



Maximal **size** of the attachment is **20MB**.



You can simultaneously upload **numerous** attachments.

Attachments can be accessed from the attachment list. Hover over **Attachment** in the upper right corner to view all attachments.

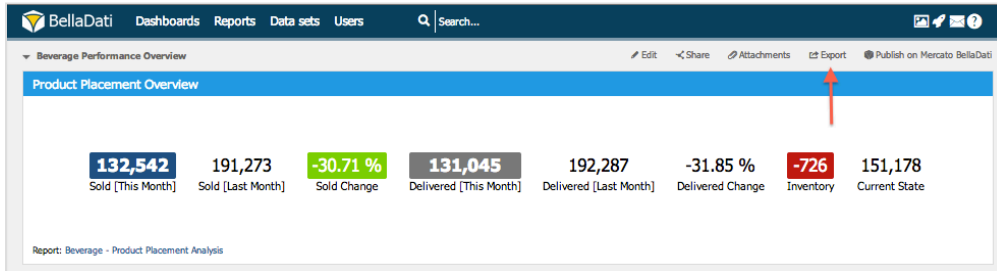
Exporting Dashboard

BellaDati allows you to export **Dashboards** into:

- PDF
- PPT
- Excel

To export:

1. Make sure that you are in Dashboard **viewing mode** (if not hit **Finish editing** green button in upper right corner of your screen)
2. Click **Export** from **action menu** in upper right corner

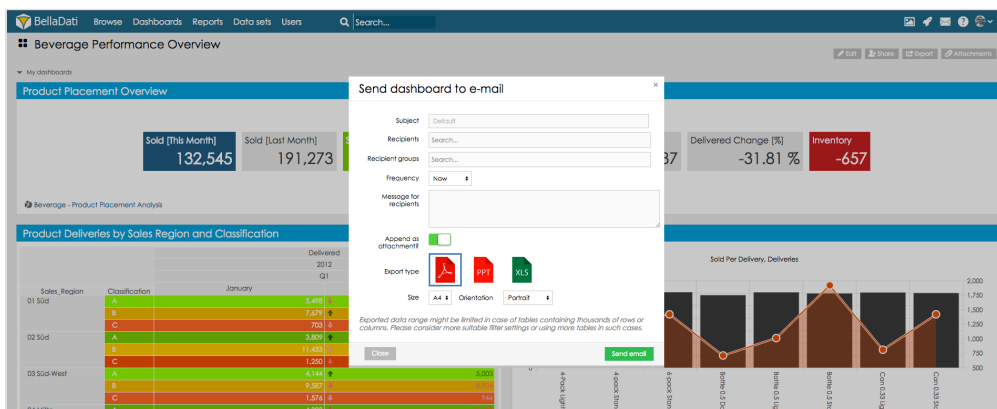


Emailing dashboard

Select **Schedule email** from report toolbox list to open *Send report as e-mail* dialog.

Dialog allows you to:

- Create **Subject**.
- Add **Recipients**.
- Add **Recipients groups**.
- Set up **Frequency** of delivery.
- Append **Message**.
- Select if report should be attached as attachment (if the checkbox is not selected, report content is in the email body - only tables and KPIs can be included).
- Select **Export type**, **Size** and **Orientation**.



You can also send multiple dashboards via one email. For more info proceed to [Export schedule](#).

Fullscreen mode

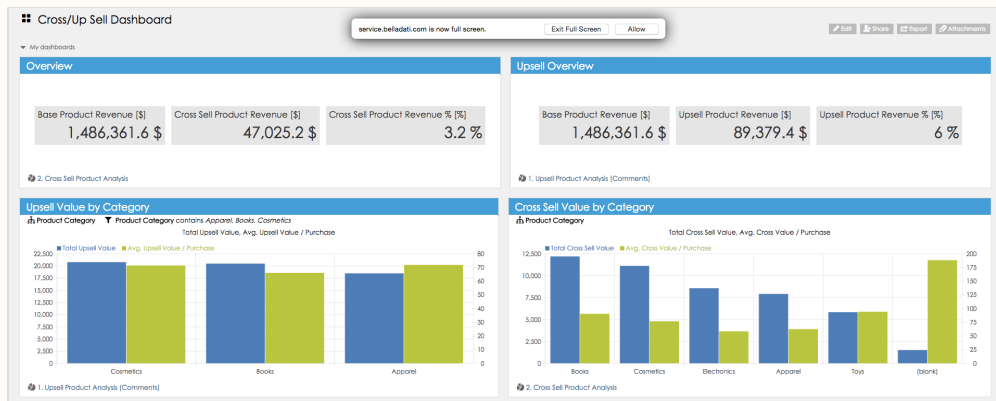
Choose option *Fullscreen* from report toolbox list to enter fullscreen mode, which will display only the report itself. This mode should be used mainly for presentation of dashboards.





Fullscreen mode is available only in following browsers:

- Firefox 10 and newer versions
- Chrome 15 and newer versions
- Opera 12.1 and newer versions
- Internet Explorer 11

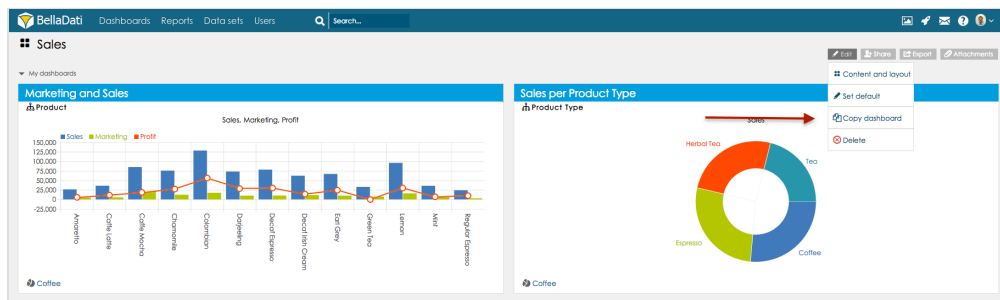


Copying dashboard

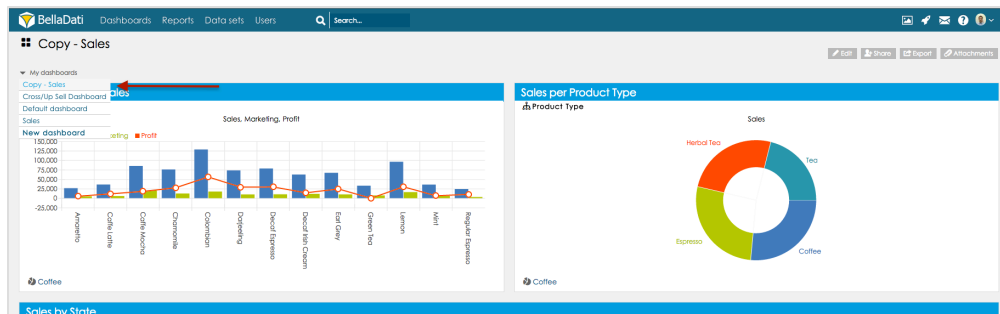


You need to be in [view mode](#) in order to copy dashboard.

Select **Copy dashboard** from dashboard toolbox list in the upper right corner to open *Copy dashboard* dialog.



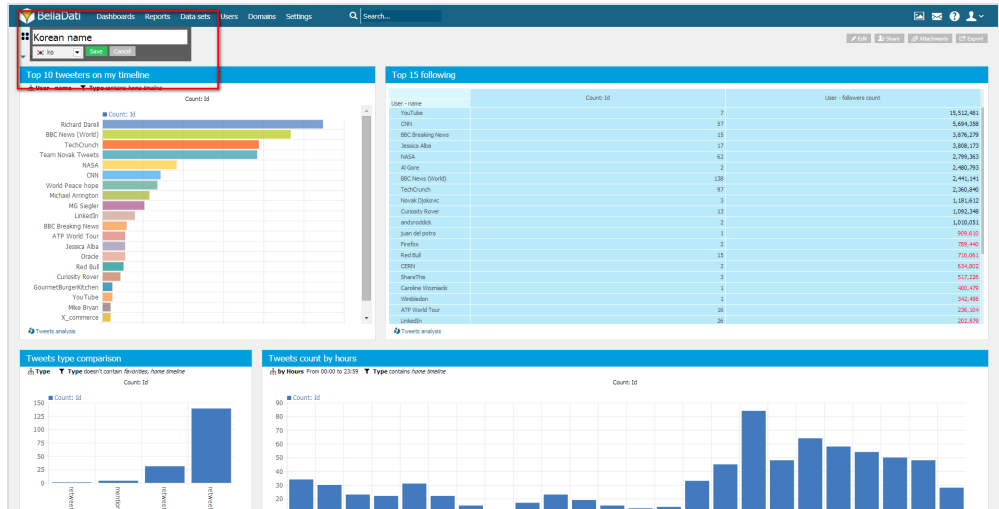
Copied dashboard can be found in **Dashboards** window in My dashboards dropdown and will be prefixed with **Copy -**.



Translating dashboards

With multilingual names support you can make your dashboard names available in various languages.

You can just **click** on the desired report or view **name** and choose the language you want to translate the name to.




Names will be displayed in the signed user's language. Language can be set in user profile, eventually in the domain detail.

See also

- [Translating Reports](#)
- [Translating Indicators](#)
- [Translating Attributes and Members](#)


Folders

Folders in BellaData serve for grouping of reports, dashboards and data sets. These groups can be used to distinguish different areas of your reports (g.e. finance, HR..).

 In the all files folder are included all the reports, dashboards and data sets which are shared with you.

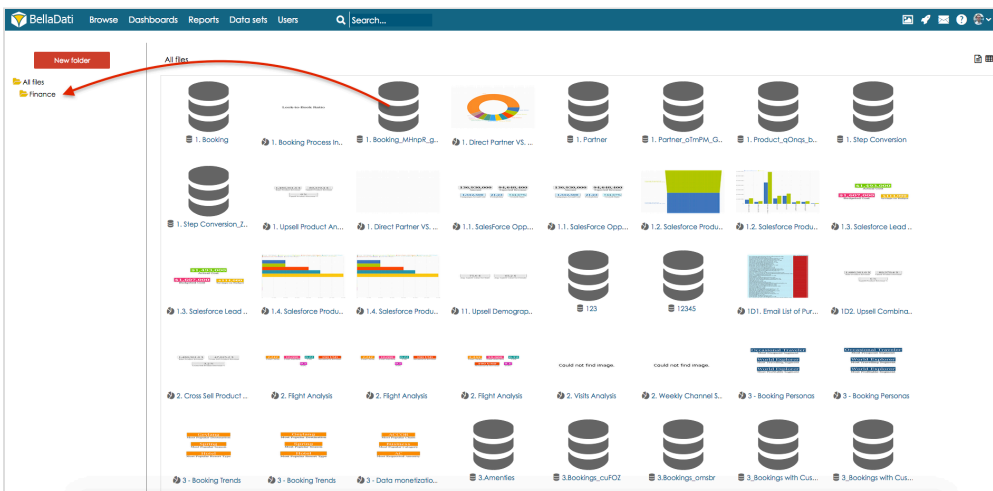
Create new folder


1. Click button *New folder*.
2. Give a name to this folder.
3. Click button *Save* to create new folder.

 Inside one folder can be also created any number of subfolders.

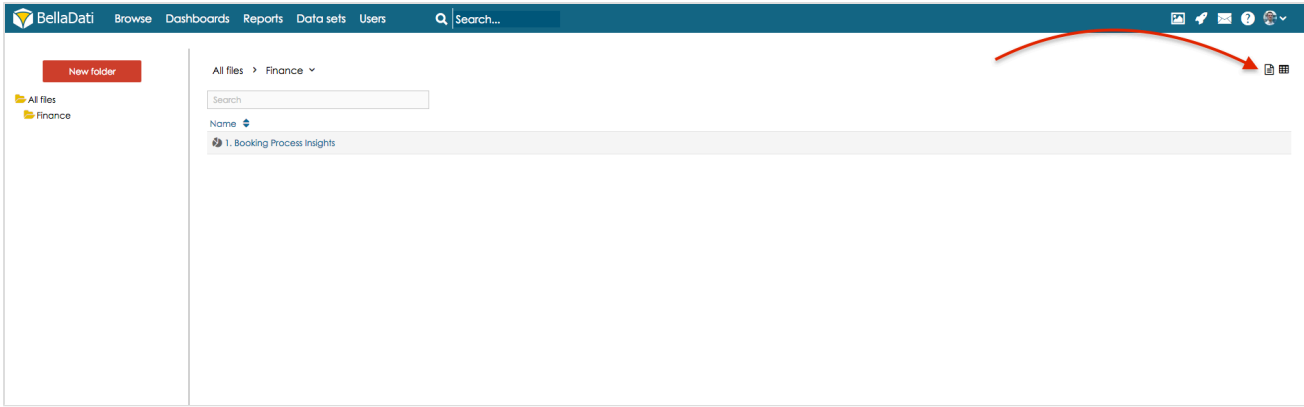
Add content into the folder

1. Grab desired report, dashboard or data set with your mouse.
2. Drop it above the selected folder.



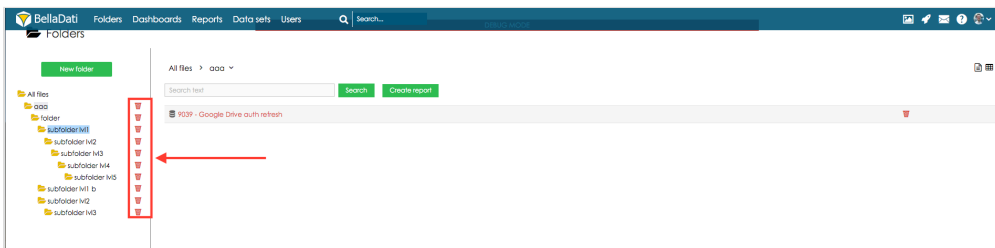
 One report, dashboard and data set can be added only to one folder.

You can change way how the list is displayed (name list or list with previews) in the top right corner.



Delete folder

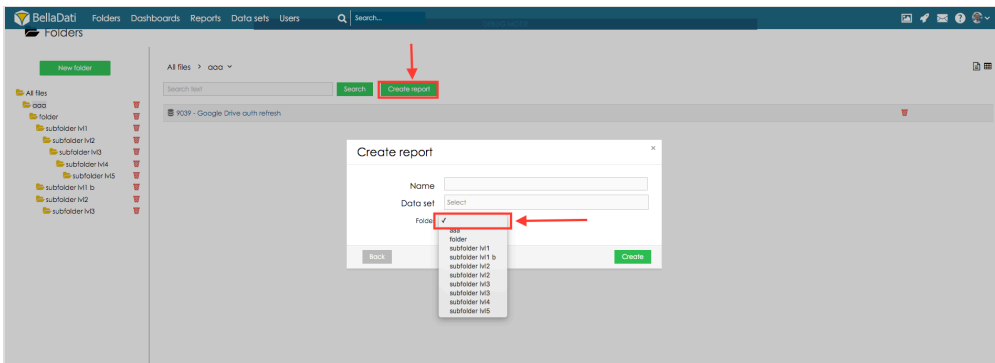
Folder can be deleted by hitting a click on the trash icon displayed next to each folder.



Create report into the folder

In order to create new report directly into selected folder should be followed these steps:

1. Hit a click on button Create report
2. Set report name
3. Select data set this report will be based on
4. Select folder into which newly created report will be added



BellaDati Mobile

BellaDati Mobile allows you to browse and interact with data on **Apple** and **Samsung** devices.



You need to have BellaDati Cloud account or OnPremise license to work with BellaDati Mobile.

Downloading BellaDati Mobile

- BellaDati Mobile for iPhone and iPad [App Store](#).
- BellaDati Mobile for Samsung [Google play](#).

Working with BellaDati Mobile

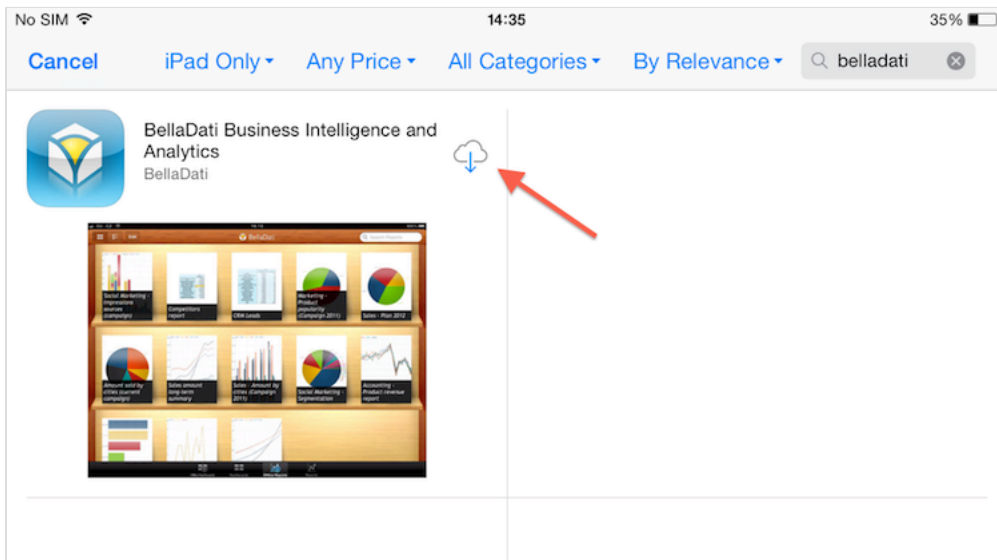
- [BellaDati Mobile for iOS](#)
- [BellaDati Mobile for Android](#)

BellaDati Mobile for iOS

Downloading and Installing BellaDati Mobile

To download and install BellaDati:


1. Go to Apple AppStore and search for **BellaDati**.
2. Click **Install**. BellaDati will be installed to your device. Optionally, you will be asked to provide your AppleID password.
3. Select **Open** immediately to launch BellaDati or tap on BellaDati icon on your homescreen.

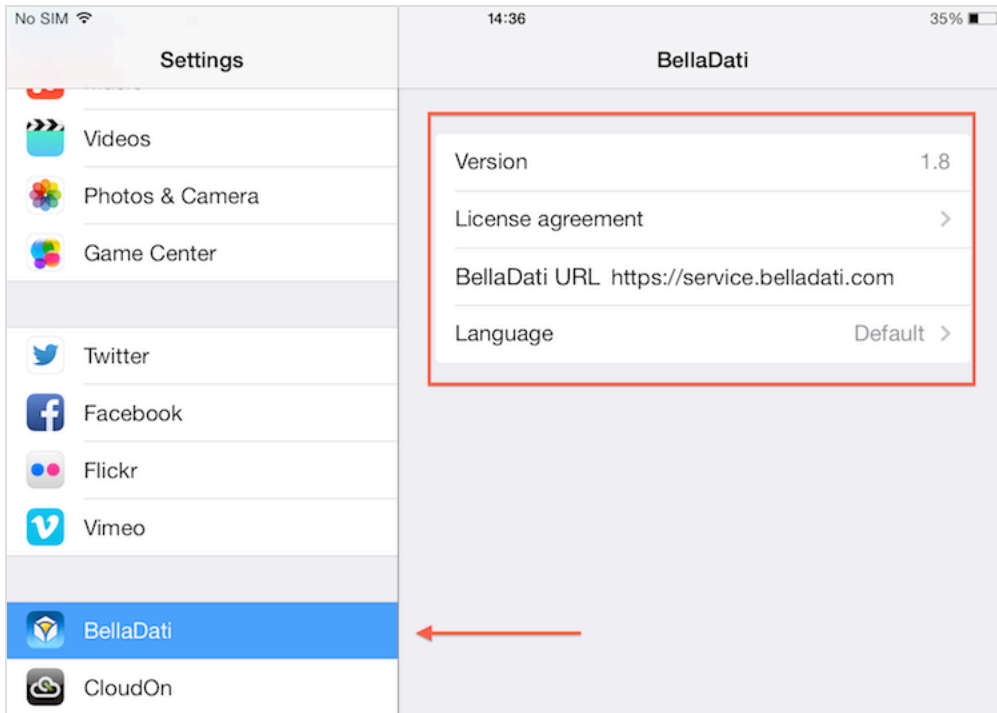


BellaDati Mobile Settings

To edit BellaDati Mobile settings, go to **Settings** and select **BellaDati**. BellaDati allows you to:

- Check current **version**
- Read **license agreement**
- Configure **BellaDati server**
- Set preferred **language**

 Currently, BellaDati offer English and Japanese localisation.



Configuring BellaDati server

By default, BellaDati Mobile points to Cloud server at: <https://service.belladati.com>. Leave this address if you will be connecting to the Cloud.

In case of connecting to BellaDati OnPremise server installed at your location, modify **Server URL** to its address.

BellaDati Mobile Key Features

BellaDati Mobile allows you to:

- Browse **online reports**
- Browse **online dashboards**
- Browse **offline reports**
- Browse **offline dashboards**



Browsing Online Reports



You need to be **logged in** to browse online reports.

In the bottom navigation menu select **Reports** and click **Login**. Provide your valid BellaDati credentials.

BellaDati will list all available reports. Click on Report **thumbnail** to browse its views.



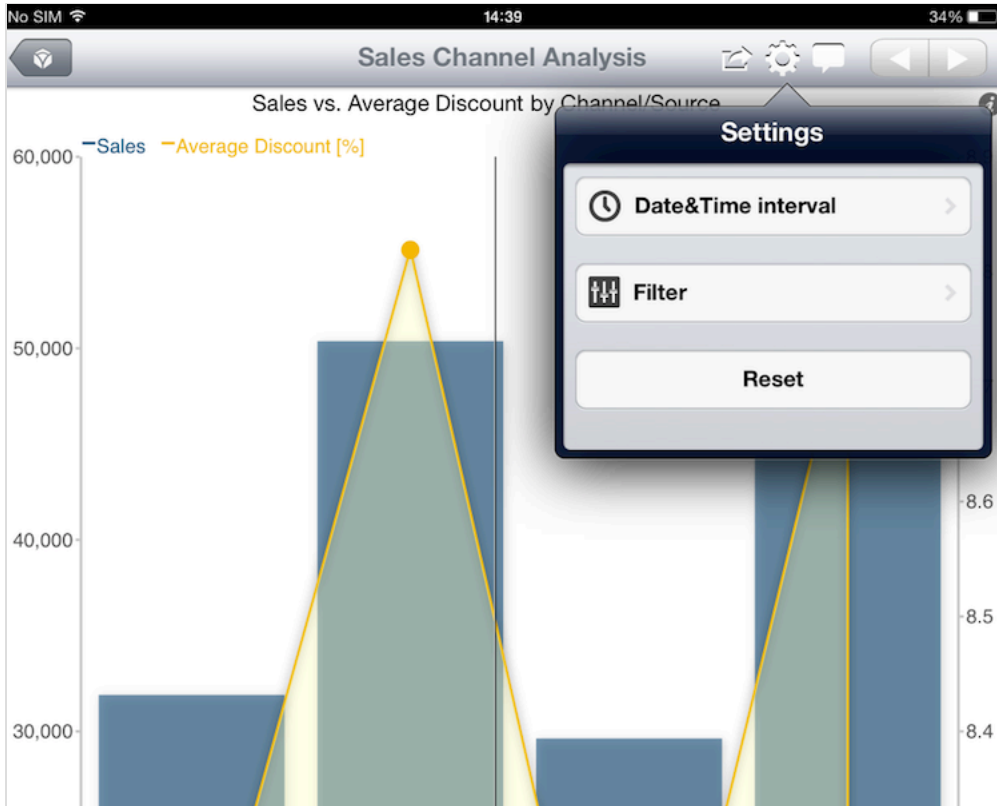
Swipe left/right or use navigation arrows to move between particular views.

Each view in **online mode** allows you to:

- change **Date&Time interval**
- apply **Filter**
- browse or add **Comments**
- download report

Changing Date&Time Interval

To change Date&Time interval tap on **Settings** icon and select **Date&Time**.
To reset changes, click on **Reset** button.



Applying Filter

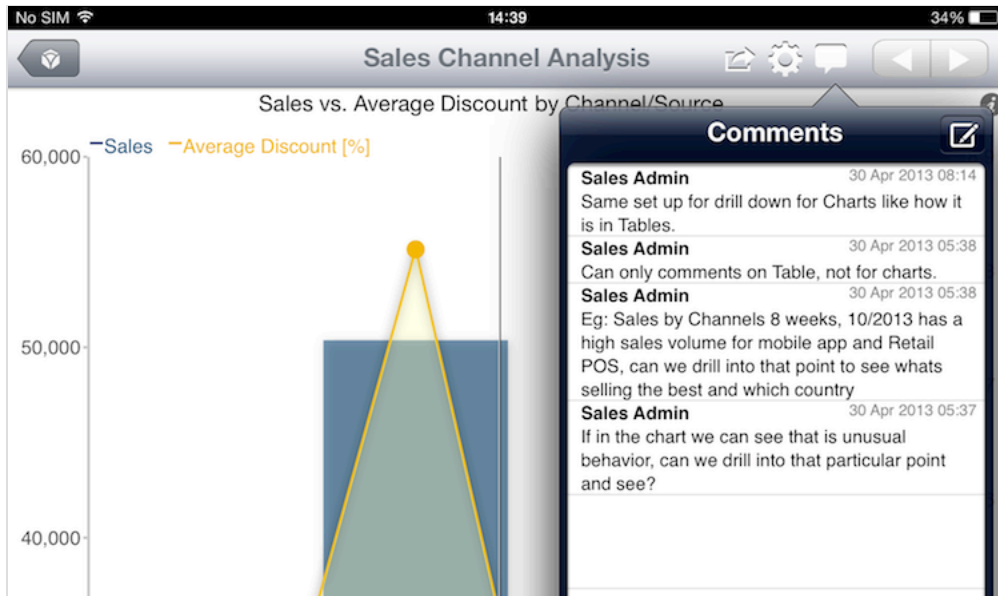
To apply Filter tap on **Settings** icon and select **Filter**. Select:

- Desired attribute
- Filtering condition
- Filtered value

To reset changes, click on **Reset** button.

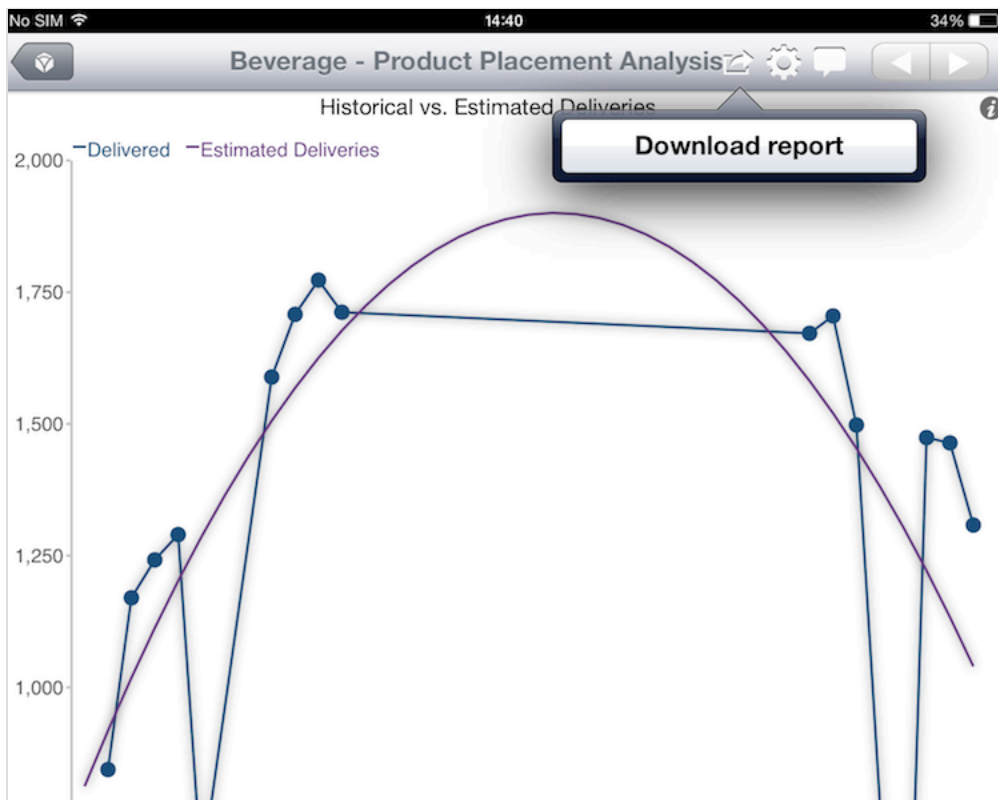
Browsing and Adding Comments

To browse or add comment tap on **Comment bubble**. To add new comment click on **New Comment** in upper right corner.




Downloading Reports

To download reports tap on **Save** icon and select **Download report**. Report will be downloaded and placed among offline reports.




Browsing Online Dashboards

 You need to be **logged in** to browse online dashboards.

In the bottom navigation menu select **Dashboards** and click **Login**. Provide your valid BellaDati credentials.

BellaDati will list all available dashboards. Click on Dashboard **thumbnail** to browse its dashlets.

 Swipe left/right or use navigation arrows to move between particular dashlets.

Each dashlet in **online mode** allows you to **download** dashboard.



Browsing Offline Reports

In the bottom navigation menu select **Offline Reports**.

BellaDati will list all available reports in your device. Click on Report **thumbnail** to browse its views.



To delete reports from your device click **Edit** and tap on black cross in upper left corner of particular report.

Browsing Offline Dashboards

In the bottom navigation menu select **Offline Dashboards**.

BellaDati will list all available dashboards in your device. Click on Dashboard **thumbnail** to browse its dashlets.



To delete dashboards from your device click **Edit** and tap on black cross in upper left corner of particular dashboard.

Searching Reports and Dashboards

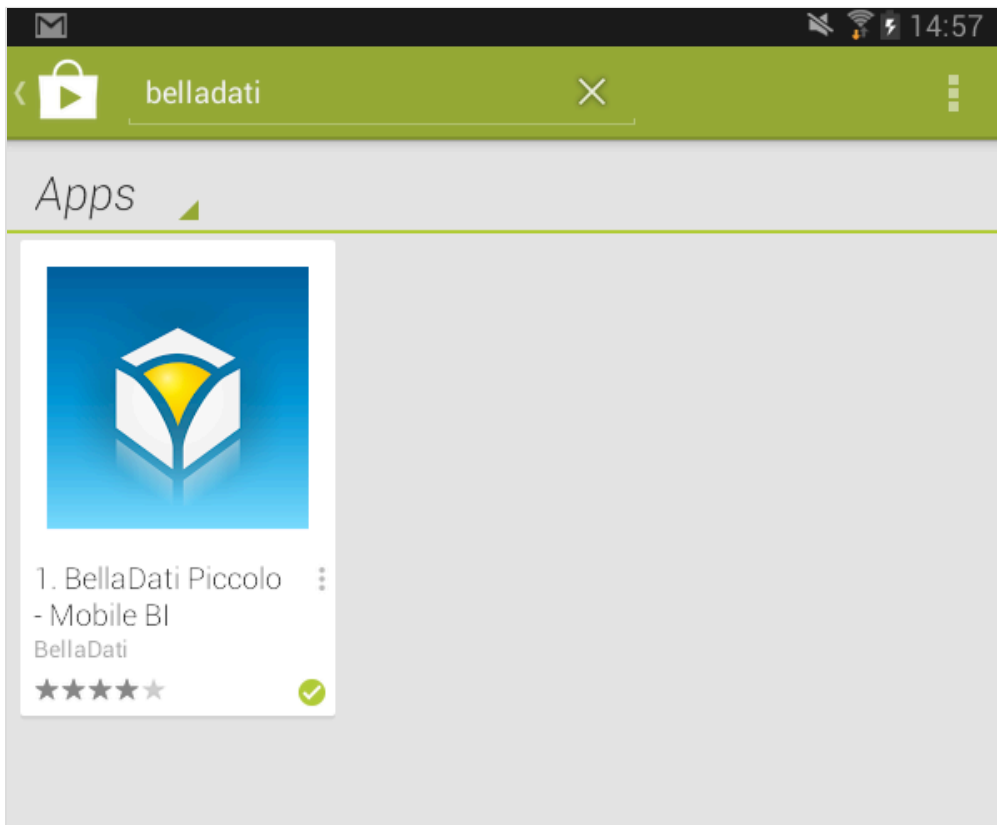
To search reports and dashboards use **Search** box in upper right corner of BellaDati Mobile.

BellaDati Mobile for Android

Downloading and Installing BellaDati Mobile

To download and install BellaDati:


1. Go to Android Google Play and search for **BellaDati**.
2. Click **Install**. BellaDati will be installed to your device.
3. Select **Open** immediately to launch BellaDati or tap on BellaDati icon on your homescreen.



BellaDati Mobile Settings

Depending on your device, you can access the settings by pressing your device's **hardware menu button** or by tapping **settings in the upper right corner** of the app. Here you can:

- Set the **BellaDati server** to connect
- Enable or disable **comment notifications**
- Set preferred **language**

 By default, the language is taken from your Android system settings and doesn't need to be set separately.

Currently, BellaDati offer English and Japanese localisation.

SETTINGS

Service URL

https://service-test.belladati.com/

Receive Notifications

☒

Language

System Default

Configuring BellaDati server

By default, BellaDati Mobile points to Cloud server at: <https://service.belladati.com>. Leave this address if you will be connecting to the Cloud.

In case of connecting to BellaDati OnPremise server installed at your location, modify **Server URL** to its address.

BellaDati Mobile Key Features

BellaDati Mobile allows you to:

- Browse **online reports**
- Browse **online dashboards**
- Browse **offline reports**
- Browse **offline dashboards**



Browsing Online Reports



You need to be **logged in** to browse online reports.

In the top navigation menu select **Reports** and from **Menu** click **Login**. Provide your valid BellaDati credentials.

BellaDati will list all available reports. Click on Report **thumbnail** to browse its views.

✓ Swipe left/right or use navigation arrows to move between particular views.

Each view in **online mode** allows you to:

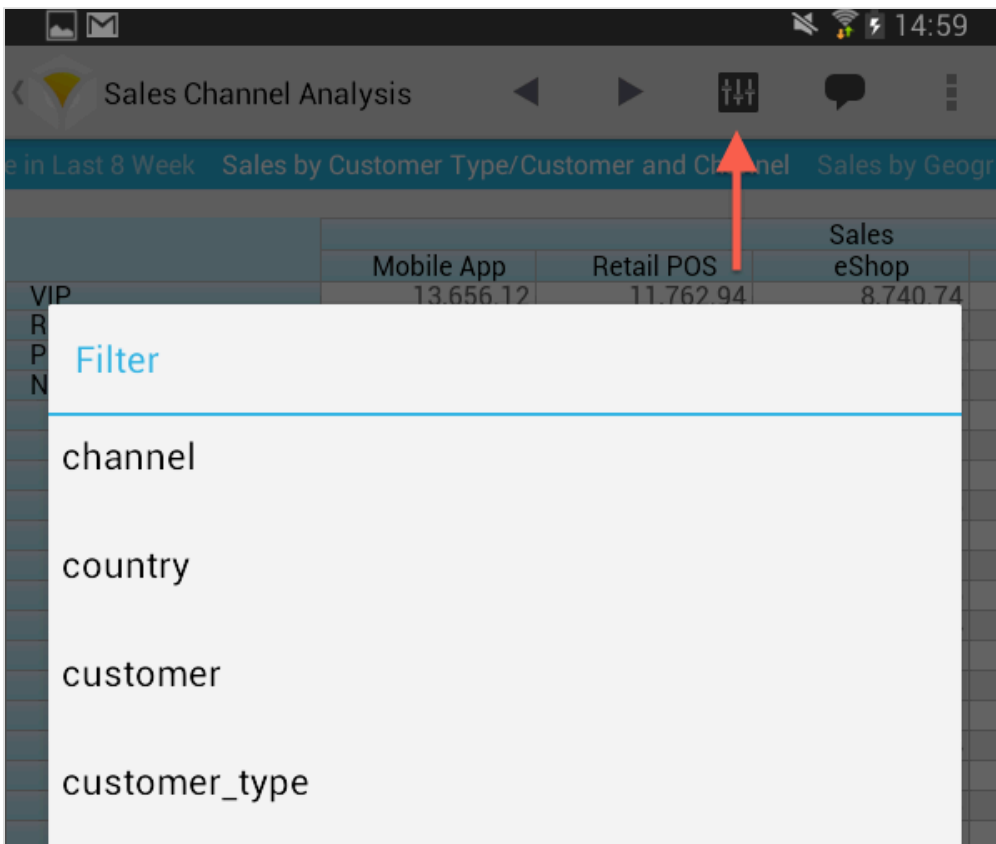
- apply **Filter**
- browse or add **Comments**
- download report

Applying Filter

To apply Filter tap on **Settings** icon and select **Filter**. Select:

- Desired attribute
- Filtering condition
- Filtered value

To reset changes, click on **Reset** button.



Browsing and Adding Comments

To browse or add comment tap on **Comment bubble**. To add new comment click on **Add Comment**.

Sales Channel Analysis

Sales in Last 8 Week Sales by Customer Type/Customer and Channel Sales by Geogra

	Mobile App	Retail POS	eShop
VIP	13,656.12	11,762.94	8,740.74
Returning	17,783.64	15,139.64	5,765.28
Premium	11,086.41	9,056.52	9,417.78
New Customer	7,847.97	8,306.88	7,980
Jermaine Schultz			
Keaton Mullins	2,393.82		

Comments

Same set up for drill down for Charts like how it is in Tables.
Sales Admin 30/04/2013 06:14

Can only comments on Table, not for charts.
Sales Admin 30/04/2013 03:38

Eg: Sales by Channels 8 weeks, 10/2013 has a high sales volume for mobile app and Retail POS, can we drill into that point to see whats selling the best and which country
Sales Admin 30/04/2013 03:38

If in the chart we can see that is unusual behavior, can we drill into that particular point and see?
Sales Admin 30/04/2013 03:37

Add Comment

Downloading Reports

To download reports tap on **Menu** icon and select **Save**. Report will be downloaded and placed among offline reports.

Sales Channel Analysis				
e in Last 8 Week Sales by Customer Type/Customer an				
Save				
	Mobile App	Retail POS	Sales eShop	
VIP	13,656.12	11,762.94	8,740.71	
Returning	17,783.64	15,139.64	5,765.23	
Premium	11,086.41	9,056.52	9,417.78	
New Customer	7,847.97	8,306.88	7,980	
Jermaine Schultz				
Keaton Mullins	2,393.82			
Hilel Terrell				
Christopher Mercado			1,769.58	
Ulric Vazquez		1,722.72		
Palmer Burton				
Lionel Mullen			1,497.6	
Honorato Benton			1,283.4	
Brenden Dickerson	1,205.82			
Lee Hensley		1,176.12		
Clayton Dunlap	1,134			
Timothy Raymond			1,004.4	
Hasad Wyatt		918.48		
Hall Skinner			861.3	
Chaim Mcmillan	837			

Browsing Online Dashboards



You need to be **logged in** to browse online dashboards.

In the top navigation menu select **Dashboards** and click **Login**. Provide your valid BellaDati credentials.

BellaDati will list all available dashboards. Click on Dashboard **thumbnail** to browse its dashlets.



Swipe left/right or use navigation arrows to move between particular dashlets.

Each dashlet in **online mode** allows you to **download** dashboard.



Browsing Offline Reports

In the bottom navigation menu select **Offline Reports**.

BellaDati will list all available reports in your device. Click on Report **thumbnail** to browse its views.

Browsing Offline Dashboards

In the bottom navigation menu select **Offline Dashboards**.

BellaDati will list all available dashboards in your device. Click on Dashboard **thumbnail** to browse its dashlets.

BellaApps



BellaDati Apps are available since 2.7.4.2 version.



BellaDati App is a package of selected dashboards and reports with related data in single **.bdt** file created from existing domain.

BellaDati App can be created by:

- **BellaDati Data Analysts team.** If you request your industry related BellaApp, contact our data analytics team at support@belladati.com.
- **BellaDati Administrator.** BellaDati App can be exported for back-up or sharing purposes.
- **3rd party developer.** Developers can create and publish their own analyses developed in BellaDati.

BellaDati Official Apps

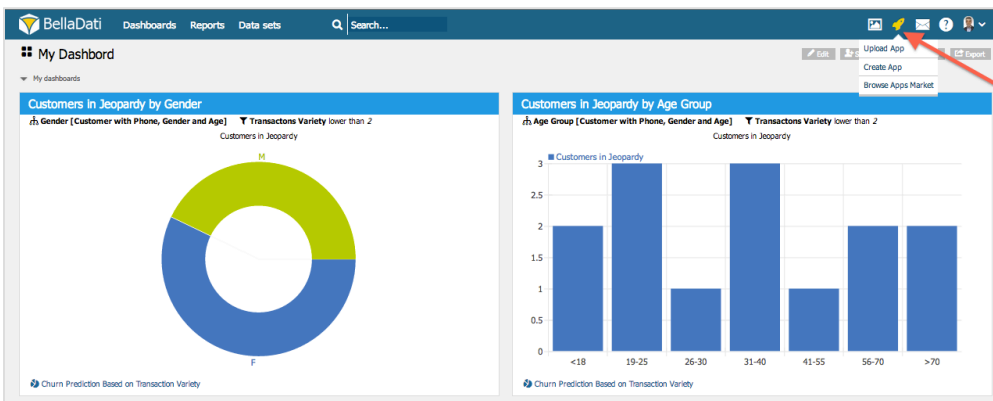
Official BellaDati Apps has following advantages for BellaDati users:

- **Include Best Practices** in KPI design.
- **Suggest common analysis** and **views**.
- **Contains scripts** and **formulas** related to KPIs and analyses.
- **Include general data model.**
- **Suggest Best Practices** for report **design**.
- **Speed up deployment** of new solution.

Working with Apps

BellaDati allows following operations related to **Apps**:

- **Creating App**
- **Importing App**




App operations can be accessed from the **Main Menu** after hovering over icon with **green down arrow**.

Continue with [Creating App](#) for detail guide how to export BellaDati App.

Continue with [Uploading App](#) for detail guide how to import BellaDati App.

Creating App

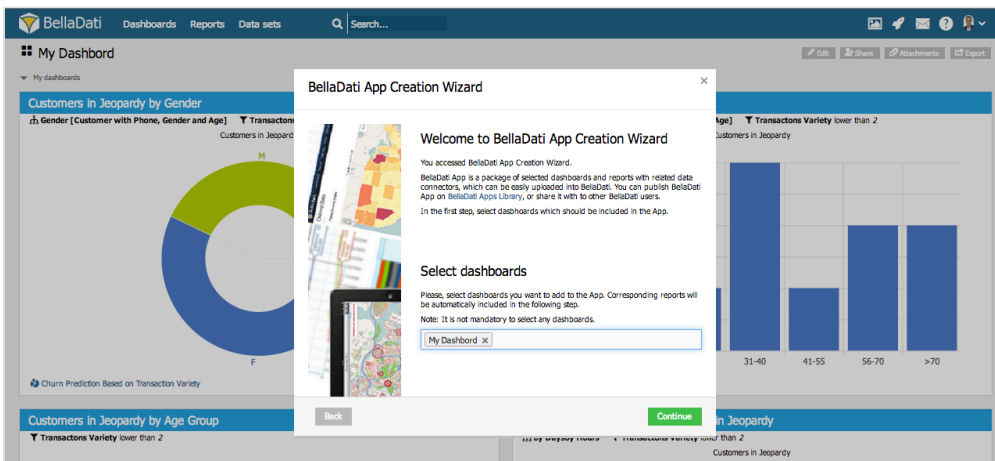
 Make sure to get familiar with **App** concept before proceeding with this section.


Click on **Create App** in BellaDati Main Menu to open *Create Template Wizard*.

Select Dashboards

Search for **Dashboards** you want to include in your App.


 **Reports** with views in selected **Dashboards** will be automatically included in the **App**.

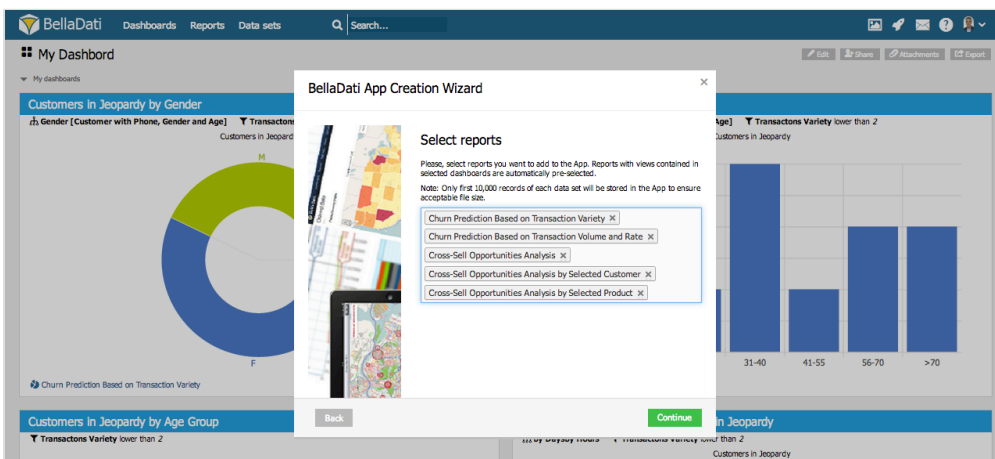


 You don't need to include any **Dashboards**.

Select Reports

Search for **Reports** you want to include in your App.

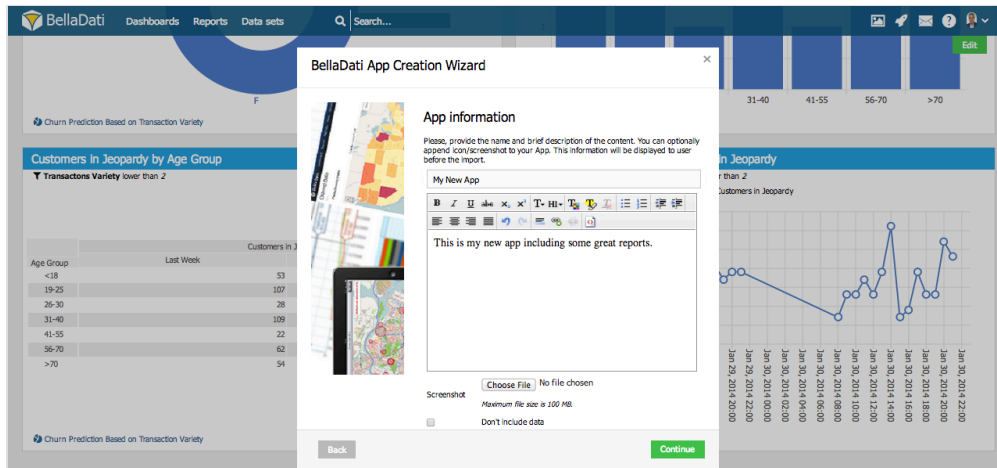
 **Reports** with views contained in selected **Dashboards** are automatically pre-selected.



Edit App Information

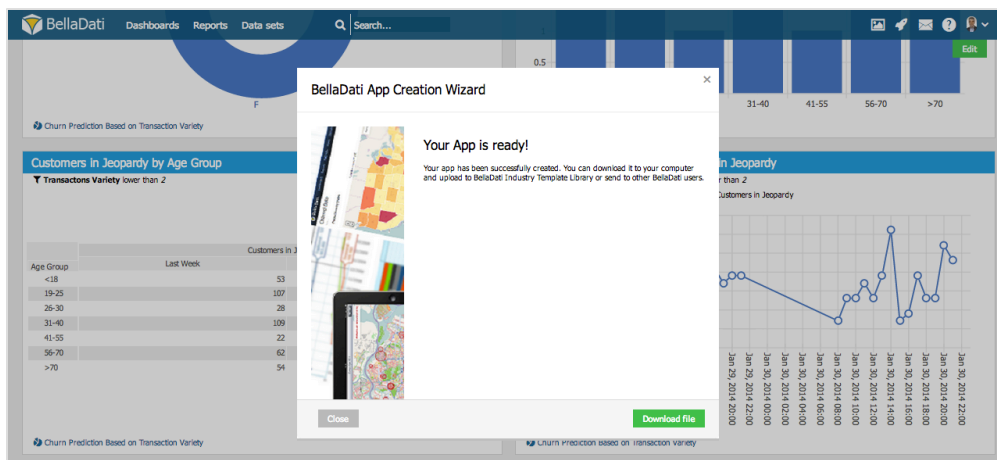
For each App, you can provide following information:

- **Name**
- **Description** in rich HTML editor
- **Screenshot or Icon**



Download App

Click **Download** button to save App to your computer.



[Uploading App Administration](#)

Uploading App

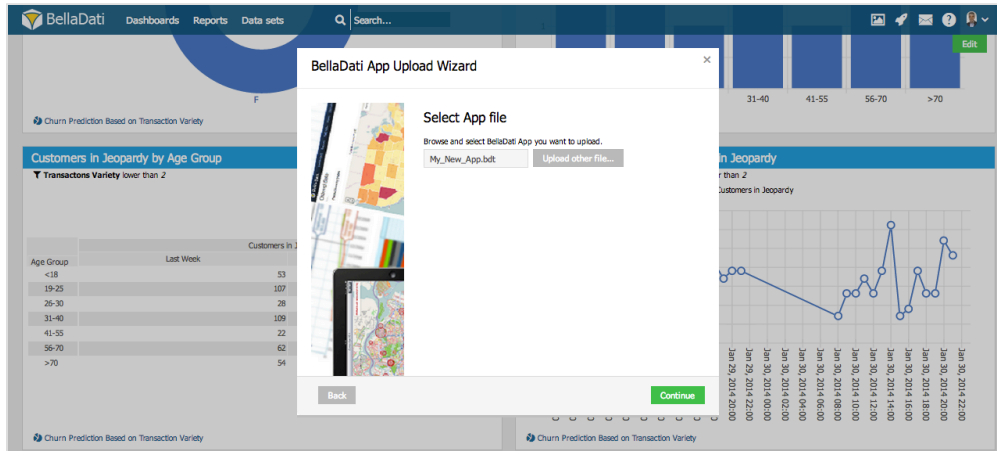


Make sure to get familiar with **BellaDati Apps** concept before proceeding with this section.

Click on **Upload App** in Template button from BellaDati Main Menu to open *Import Template Wizard*.

Select App

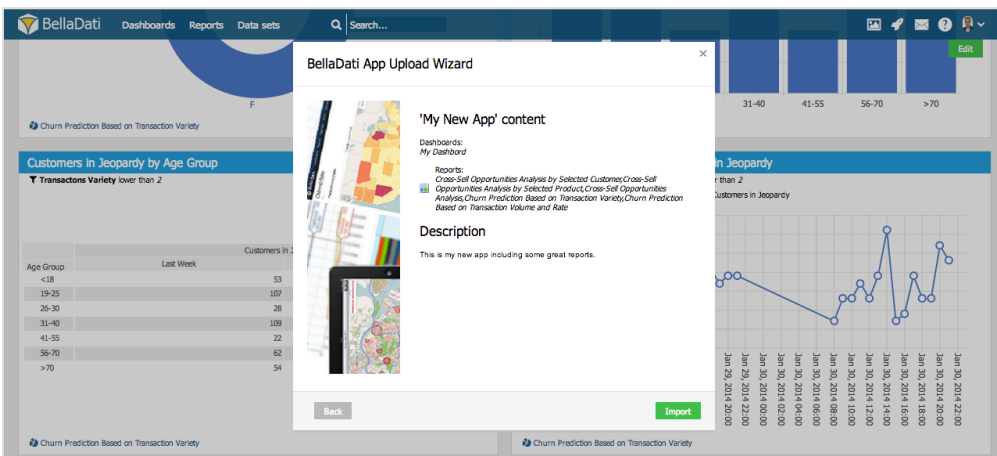
Browse for **BellaDati App** file in your computer.



App Information

In the **App** information, you can find out:

- Included **Dashboards**
- Selected **Reports**
- App **Description**
- Attached **Screenshot** or **Icon**



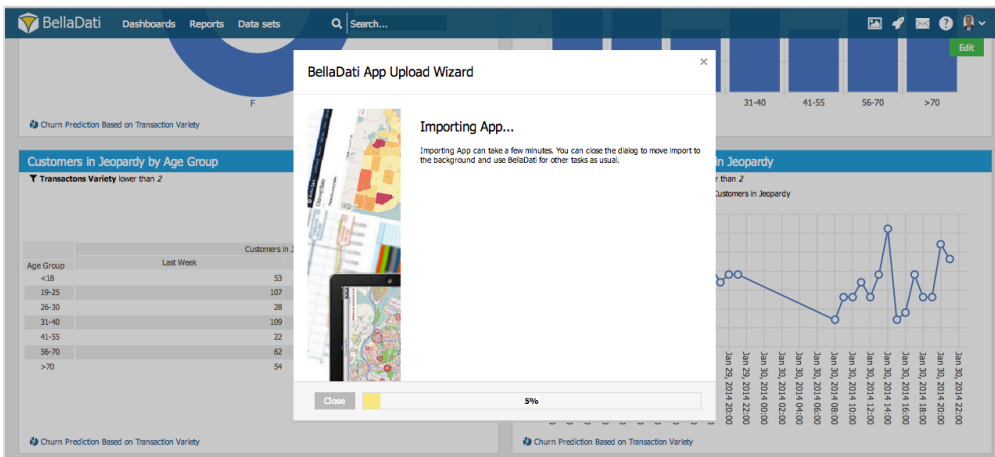
Click **Import** to proceed with App upload.

Import App

App is being imported into BellaDati.



You can close wizard and work with BellaDati. Import will be processed in the background.



After successful import, you can find uploaded **Reports** and **Dashboards** in your domain.

[Creating App Administration](#)

Media Gallery

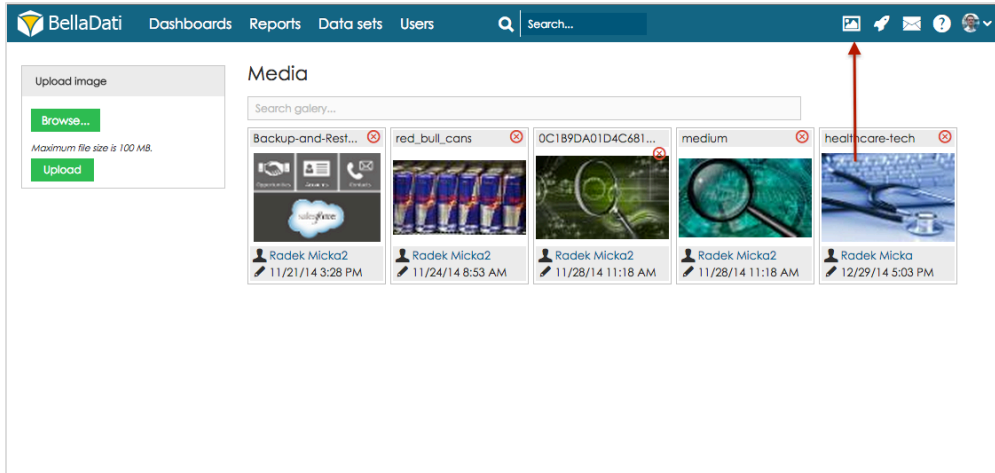
BellaDati allows you to create infographics and extend your reports with visualizations thanks to its extensive **Media Gallery**.

Managing Media

Click on the **folders** icon in the upper right corner to enter **Media Gallery**.

Media Gallery allows you to:

- **Browse** images
- **Upload** new images
- **Delete** existing images
- **Search** images



To **Upload new image** into the gallery:

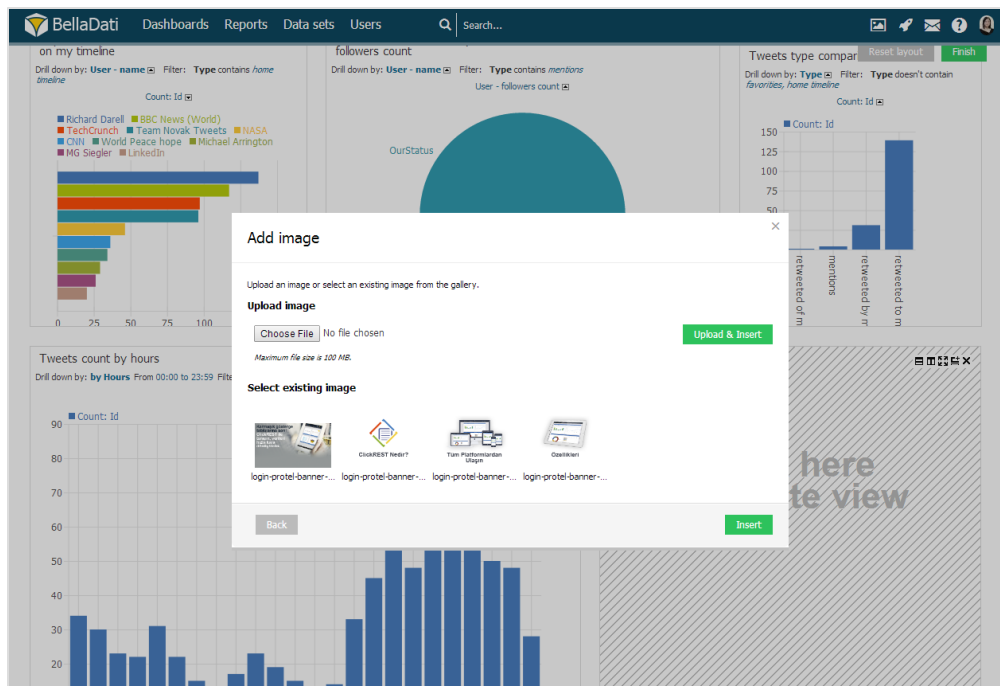
- Click button **Browse** (double click needed if you are using Internet Explorer 8)
- Choose image which should be uploaded
- Click button **Upload** to upload selected image into the gallery

Using Media

To insert images into the report from the **Media Gallery**, hover over empty view ale select Media / Images.

Media browser allows you to:

- **Upload** new images
- **Select** and **Insert** existing images.

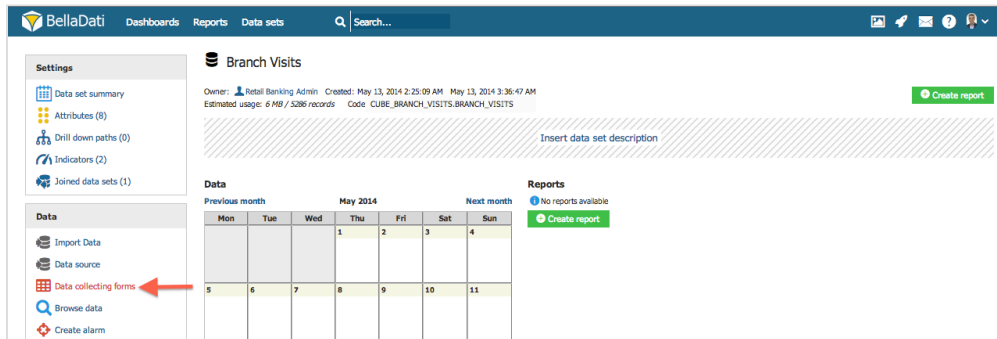


Data Collection Module



You have to have **Data Collection** enabled to be able to create and publish forms.

Data Collection module allows you to create and publish forms connected to BellaDati **Data Sets**.



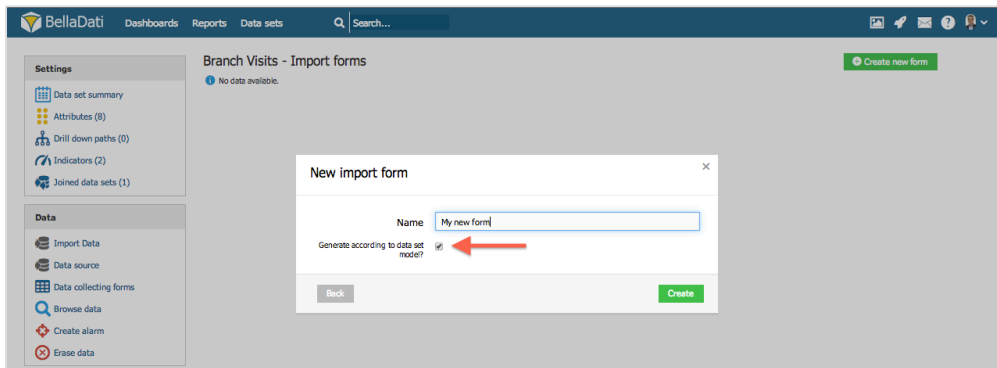
Creating Form

Navigate to **Data Set** for which you want to create the form. Click on **Data Collecting Forms** in left navigation.

BellaDati will list all existing forms. Click on the **Create new form** button.



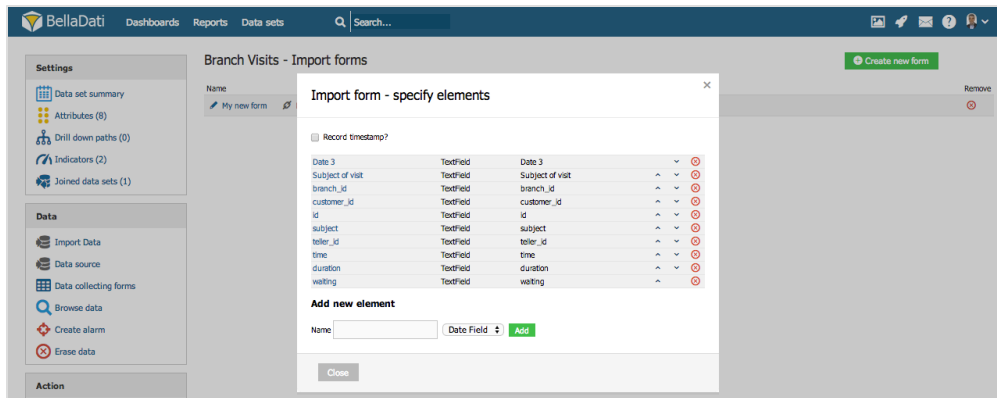
You can generate form based on existing table structure. Check **Generate according data set structure** to have form prepared for you.



Creating Form Elements

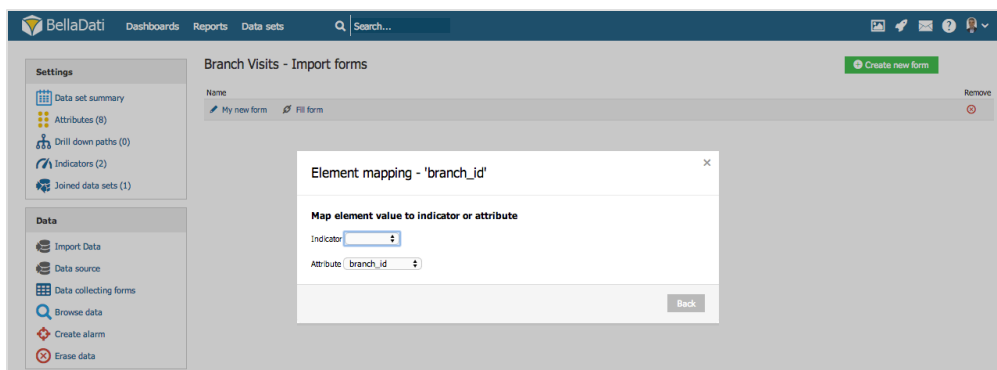
To create form element, provide its name, type and click **Add** button. BellaDati offers following input types:

- **Date field**
- **Text Field**
- **Checkbox**
- **Select**
- **Username**



Mapping Elements to Data Set Columns

In order to map form element to data set column, click on the element name and select one of the **Attributes** or **Indicators**.



Publishing Form

Click on **Fill form** from **Data Collecting Forms** list. BellaDati will open new window.

- Distribute URL of this form to all users responsible for collecting data.

Branch Visits - data collecting form

You are logged-in as **Retail Banking Admin** (logout)

Switch to multi form

Date 3
Subject of visit
branch_id
customer_id
id
subject
teller_id
time
duration
waiting

Submit

- Field with type username is not visible. Username of logged in user will be recorded.

Managing Forms

You can create multiple forms. Click on **Create new form** to add new one.
You can modify form anytime by clicking on its name.



To watch some use cases head to our Video Tutorials [section](#).

Administration

This section describes the basic BellaDati administration which can be performed via application GUI by an user with necessary [permissions and roles](#).



If you are looking for detailed guide for system administration, visit the [Developer network](#) instead.

The following chapters are covered here:

- [Administering Users](#)
- [Importing Users](#)
- [Administering User Groups](#)
- [Sharing console](#)
- [Managing User Profile](#)
- [Managing Configuration](#)
- [Domain Configuration and Administration](#)
- [Domain Backup](#)
- [Usage Monitoring](#)
- [Administering domains](#)
- [Mass mailing](#)
- [Report transferring](#)



You must have the Domain Administrator role to administer your domain.

The screenshot shows the 'Users' management page in the BellaDati application. The top navigation bar includes 'Dashboards', 'Reports', 'Data sets', and 'Users'. A search bar is present on the right. On the left, an 'Action' sidebar contains 'Create User' and 'Import users'. The main area displays a table of users with columns: Name, Login, Last login, Business phone, User groups, and Domain. The table lists three users: 'Mr. Bank', 'Mr. Teller', and 'Retail Banking Admin'. The 'Last login' for 'Retail Banking Admin' is 'Jun 17, 2024 4:41 AM'.

Name	Login	Last login	Business phone	User groups	Domain
Mr. Bank	mr@bank@belladati.com				Retail Banking
Mr. Teller	mr@teller@belladati.com				Retail Banking
Retail Banking Admin	retailbankingdemo@belladati.com	Jun 17, 2024 4:41 AM			Retail Banking

- [Administering Users](#)
- [Importing Users](#)
- [Administering User Groups](#)
- [Sharing console](#)
- [Managing User Profile](#)
- [Managing Configuration](#)
- [Domain Configuration and Administration](#)
- [Domain Backup](#)
- [Usage Monitoring](#)
- [Administering domains](#)
- [Mass mailing](#)
- [Report transferring](#)

Administering Users

Click **Users** in the main menu to display the list of users in the current domain.

This table shows:

- User name (surname, title)
- Login name (usually e-mail)
- Last login date and time
- Phone
- User group each user belongs to
- Authentication source
- Domain (*more domains are relevant for global BellaDati administrator only*)
- Active/Inactive user

Name	Login	Last login	Business phone	User groups	Authentication source	Domains	Active
Mr. User	aaaa@test.com	Jan 12, 2015 11:50 AM		abc	BellaDati	Radek-test	✓
Super User	bbbb@test.com	Jan 12, 2015 11:51 AM		abc	BellaDati	Radek-test	✓
Radek Micka	radek.micka@belladati.com	Feb 3, 2015 3:38 PM		abc	BellaDati	Radek-test	✓

Actions available:

- **Sort** existing users by name, login, last login date and phone.
- **Filter users** by: expression (match within name or login), user group, show deactivated users
- **Search user**
- **Create new user**
- **Bulk user import** using the CSV file
- **Bulk change delete**

Creating user

1. Click **Create User** in the left submenu. The popup appears.
2. Enter the information and set the options below.
3. Click **Create**. New user has been created now in the actual domain.

Enter the following information in the popup (bold are mandatory):

- **Name**: 2 characters minimum
- **Surname**: 2 characters minimum
- **E-mail**
- Phone
- Mobile phone

You can immediately assign these basic [roles](#) to the user:

- Report editor
- Data manager



Check "Send notification" option to let the new user know about his new account in BellaDati via automatic e-mail.

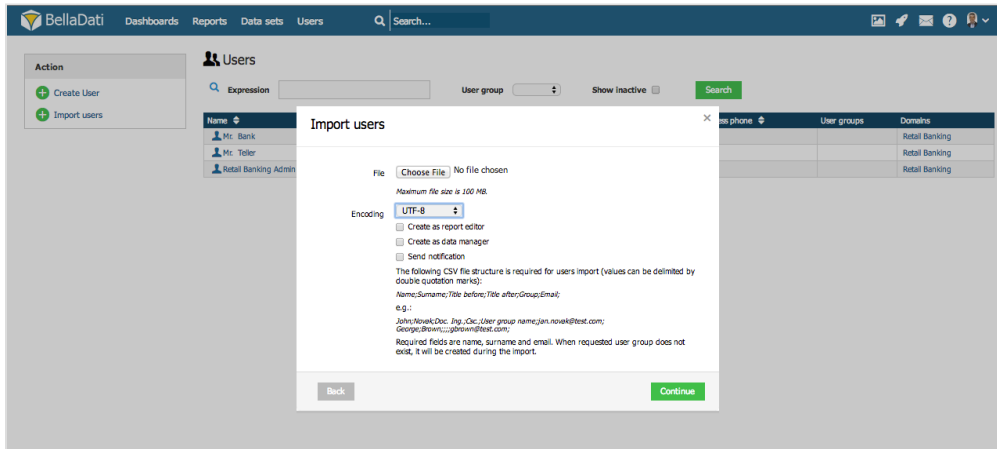



To send email notifications, your administrator must have [configured an email server](#).

Importing Users

BellaDati allows you to **import users** from external systems. To import users:

1. Go to **User Administration**
2. Click **Import Users**



 BellaDati offers bulk import from text (CSV) file. This is a fast way to create new users when you are migrating from an old BellaDati domain or another application and you already have the list of users

1. Select the text file to upload.
2. Set encoding: UTF-8, Win-1250, Win-1252, ISO-8859-1 or Auto
3. Optionally set [roles](#): Report editor, Data manager
4. Check the "Send notification" option will let the new users know about their new account in BellaDati via automatic e-mail.

The file structure:



1. **Name, surname and e-mail** are mandatory.
2. The e-mail will serve also as login name. The password will be generated randomly.
3. When requested user group does not exist, it will be created during the import.
4. The selected roles will be assigned to all users in the list. When no role has been selected, all users will have only the common BellaDati user role.

Administering User Groups

User groups serve for easier and more transparent sharing [data sets](#), [Reports](#), and [Dashboards](#) within domain.

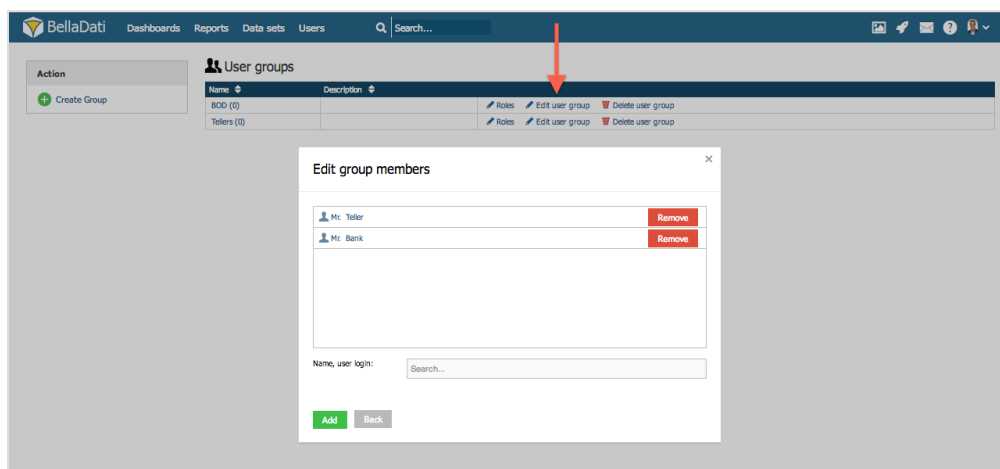


The main advantage of using user groups is the simplification of controlling the user access to underlying BellaDati objects (reports, dashboards, data sets and data).

Click **User groups** in the main menu **Users**. The list of all groups in the current domain will be displayed. You can sort the list by group name and description.


The following actions are available:

- **Add or remove group users** (members): Click on the group name. The popup appears. Use autocomplete for adding the new users.
- **Edit group name and description**
- **Create new group**: Enter the group name and optional description and click "Save".
- **Set roles**: report editor, data manager
- **Delete user group**: You must confirm the delete in the popup.



Sharing console

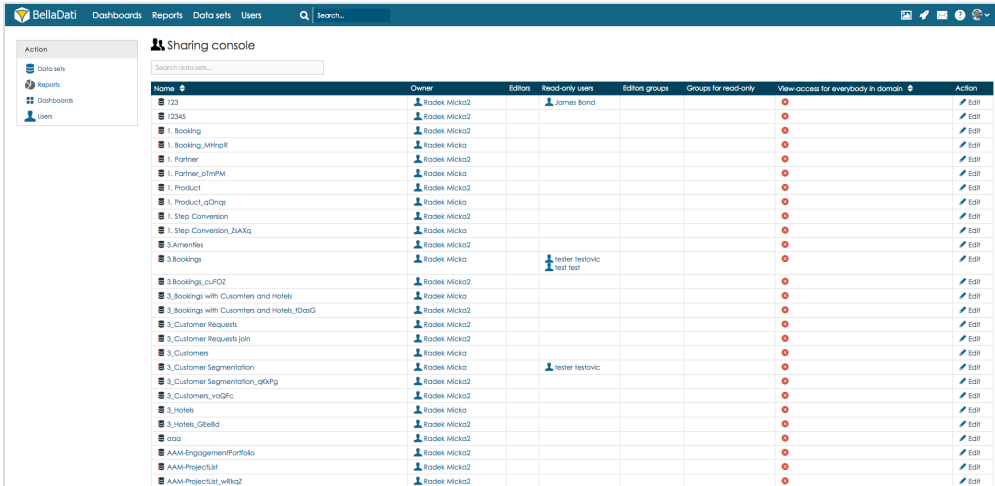
Point to the Users in the top menu and click Sharing console. On this page there you can view and change sharing settings of all the reports, dashboards and data sets or see the list of shared reports, dashboards and data sets with selected user.

 Only Domain administrator is allowed to access the Sharing console.

Data sets

The list contains separate columns for data set owner, editors, read-only users, group editors, read only groups.

To change the sharing settings click button Edit for selected data set.

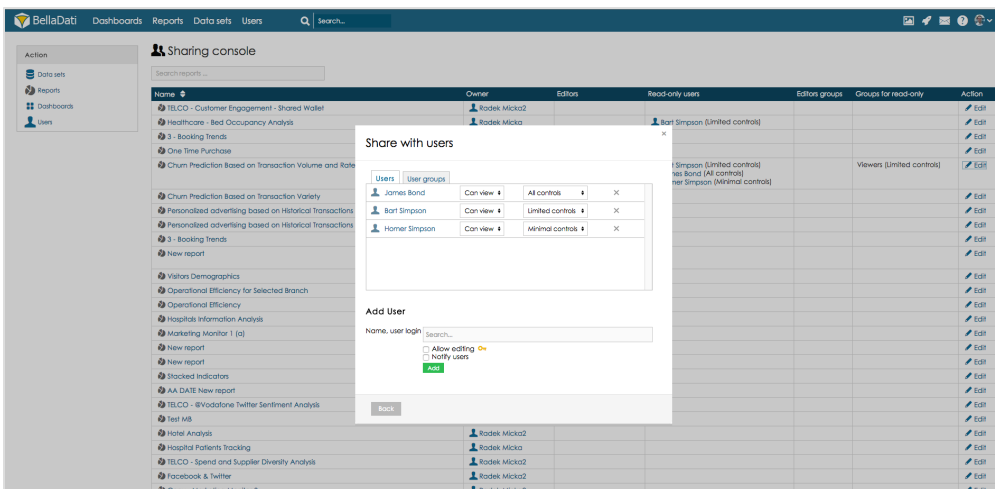


Name	Owner	Editor	Read-only users	Editor groups	Groups for read-only	View access for everybody in domain	Action
123	Radek Micka2		James Bond				Edit
12345	Radek Micka2						Edit
1. Booking	Radek Micka2						Edit
1. Booking_MiniPit	Radek Micka2						Edit
1. Partner	Radek Micka2						Edit
1. Partner_crmPM	Radek Micka2						Edit
1. Product	Radek Micka2						Edit
1. Product_gOnp	Radek Micka2						Edit
1. Step Conversion	Radek Micka2						Edit
1. Step Conversion_JuXq	Radek Micka2						Edit
3.Amenities	Radek Micka2						Edit
3.Bookings	Radek Micka2		tester testovic test test				Edit
3.Bookings_cufQZ	Radek Micka2						Edit
3.Bookings with Customers and Hotels	Radek Micka2						Edit
3.Bookings with Customers and Hotels_IdatG	Radek Micka2						Edit
3.Customer Requests	Radek Micka2						Edit
3.Customer Requests join	Radek Micka2						Edit
3.Customers	Radek Micka2						Edit
3.Customer Segmentation	Radek Micka2		tester testovic				Edit
3.Customer Segmentation_gkXpG	Radek Micka2						Edit
3.Customers_vdGfc	Radek Micka2						Edit
3.hotels	Radek Micka2						Edit
3.hotels_GfEldd	Radek Micka2						Edit
200	Radek Micka2						Edit
AAA-EngagementPortfolio	Radek Micka2						Edit
AAA-ProjectList	Radek Micka2						Edit
AAA-ProjectList_wilka2	Radek Micka2						Edit

Reports

The list contains separate columns for report owner, report editors, read-only users, group editors, read only groups.

To change the sharing settings click button Edit for selected report.



Name	Owner	Editor	Read-only users	Editor groups	Groups for read-only	Action
TELCO - Customer Engagement - Shared Wallet	Radek Micka2					Edit
Healthcare - Bed Occupancy Analysis	Radek Micka2		Bar Simpson (limited controls)			Edit
3 - Booking Trends			Bar Simpson (limited controls)			Edit
One Time Purchase			Bar Simpson (limited controls)			Edit
Churn Prediction Based on Transaction Volume and Rate			Bar Simpson (limited controls)		Viewers (limited controls)	Edit
Churn Prediction Based on Transaction Variety			Bar Simpson (limited controls)			Edit
Personalized advertising based on Historical Transactions			Bar Simpson (limited controls)			Edit
Personalized advertising based on Historical Transactions			Bar Simpson (limited controls)			Edit
3 - Booking Trends			Bar Simpson (limited controls)			Edit
New report			Bar Simpson (limited controls)			Edit
Values Demographics			Bar Simpson (limited controls)			Edit
Operational Efficiency for Selected Branch			Bar Simpson (limited controls)			Edit
Operational Efficiency			Bar Simpson (limited controls)			Edit
Hospital Information Analysis			Bar Simpson (limited controls)			Edit
Marketing Monitor 1 (a)			Bar Simpson (limited controls)			Edit
New report			Bar Simpson (limited controls)			Edit
New report			Bar Simpson (limited controls)			Edit
Stacked Indicators			Bar Simpson (limited controls)			Edit
AA DATE New report			Bar Simpson (limited controls)			Edit
TELCO - Vodafone Twitter Sentiment Analysis			Bar Simpson (limited controls)			Edit
Test M8			Bar Simpson (limited controls)			Edit
Hospital Analysis			Bar Simpson (limited controls)			Edit
Hospital Patients Tracking			Bar Simpson (limited controls)			Edit
TELCO - Spend and Supplier Diversity Analysis			Bar Simpson (limited controls)			Edit
Facebook & Twitter			Bar Simpson (limited controls)			Edit
Copy - Marketing Monitor 3			Bar Simpson (limited controls)			Edit

Dashboards

The list contains all the **Dashboards** with users and user groups who can view the dashboard.

To change the sharing settings click button Edit for selected report.

BellaDati

Dashboards

Reports

Data sets

Users

Search...

Search...

Action

Data sets

Reports

Dashboards

Users

Sharing console

Search dashboards ...

Name	Owner	Read-only users	Groups for read-only	Public access	Alerts	Action
Reference Dashboard	Radek Micka					
Flights Overview	BellaDati BellaDati					
Default dashboard	Radek Micka					
CrossUp Sell Dashboard	Radek Micka2					
Default dashboard	test test					
Retail Banking App	Radek Micka2					
Retail Banking App	BellaDati BellaDati					
Insights into Booking Rates and Process	Radek Micka2					
Sample dashboard - Healthcare	Radek Micka					
MQJ dashboard	release test					
Hospital Analysts Dashboard	Radek Micka					
Retail Banking App	Radek Micka2					
MQJ dashboard	James Bond					
MQJ dashboard	bart Simpson					
MQJ dashboard	Homer Simpson					
Default dashboard	BellaDati BellaDati					
Default dashboard	Radek Micka2					
demo dash	Radek Micka					
MQJ dashboard	tester testovic					
Telco Dashboard	Radek Micka2					
Customer Segmentation insights	Radek Micka2					
Hospital Analysts Dashboard	Radek Micka					
Revenue and Emission for Illinois Region	Radek Micka2					
Retail Banking App	Radek Micka2					
Performance by Marketing Source	Radek Micka2					

Users

Choose user in the input box to display dashboards, reports and data set which are shared or created by selected user.

To change the sharing settings click button Edit for selected report, dashboard or data set.

BellaDati

Dashboards

Reports

Data sets

Users

Search...

Action

Data sets

Reports

Dashboards

Users

Sharing console

Search data sets...

Name	Owner	Editors	Read-only users	Editors groups	Groups for read-only	View-access for everybody in domain	Action
123	Radek Micka2		James Bond				Edit
12345	Radek Micka2						Edit
1. Booking	Radek Micka2						Edit
1. Booking_Mixupit	Radek Micka						Edit
1. Partner	Radek Micka2						Edit
1. Partner_cdmfm	Radek Micka						Edit
1. Product	Radek Micka2						Edit
1. Product_sOnap	Radek Micka						Edit
1. Step Conversion	Radek Micka2						Edit
1. Step Conversion_BAAq	Radek Micka						Edit
3.Amenities	Radek Micka2						Edit
3.Bookings	Radek Micka		tester testovic test test				Edit
3.Bookings_cufQZ	Radek Micka2						Edit
3.Bookings with Customers and Hotels	Radek Micka						Edit
3.Bookings with Customers and Hotels_DarG	Radek Micka2						Edit
3.Customer Requests	Radek Micka2						Edit
3.Customer Requests job	Radek Micka2						Edit
3.Customers	Radek Micka						Edit
3.Customer Segmentation	Radek Micka		tester testovic				Edit
3.Customer Segmentation_qKfPg	Radek Micka2						Edit
3.Customers_vdGfc	Radek Micka2						Edit
3.hotels	Radek Micka						Edit

Managing User Profile

Click on your profile picture in the top right menu to display your **profile**.

Each BellaDati user has the following information associated (bold are mandatory):

- **Login email**
- **Name:** 2 characters minimum
- **Surname:** 2 characters minimum
- Degree before, degree (title)
- Photo
- Work details: office, position
- Job title
- Phone, mobile
- **E-mail**
- Address: street, number, city, district, region, zip code, state



Uploading the photo is recommended since you can easily identify who has created or modified an report, data set, dashlet or imported the new data via tooltips.

The screenshot shows the 'Edit User Profile' interface for a user named 'Retail Banking Admin'. The form is divided into two main sections: 'Basic info' and 'Address'. The 'Basic info' section includes fields for Login (retailbankingdemo@belladati.com), Expire when (MM/yyyy), Name (Retail Banking), Surname (Admin), Email (retailbankingdemo@belladati.com), Photo (with a 'Remove' button and an 'Update photo' section with a 'Choose File' button), Office (BDO), Job title (CTO), Business phone (702023559), and Mobile (702023559). The 'Address' section includes fields for Street (Rest Mile 1), City (Chicago), District (Loop), Region (IL), Zip code (13000), and Country (United States). On the left, there is a sidebar with a user profile picture and a list of actions: Edit User Profile, Change password, User roles, and User groups.

Preferred language

You can select locale, which defines, how BellaDati is displaying labels.

- **Use browser default settings:** BellaDati selects the actual language automatically according your browser's locale. Since BellaDati does not support all locales (see below), English will be the default language in most cases.
- **Particular language** (represented by flag): BellaDati will be always displayed in the selected language regardless the domain or browser locale.

This screenshot shows the 'Configuration' section of the 'Edit User Profile' form. It includes a 'Preferred language' dropdown menu set to 'English', a 'Charts display technology' dropdown set to 'Auto detection', and a 'Time zone' dropdown. A red arrow points to the 'Preferred language' dropdown. The 'Save' button is at the bottom left. The sidebar on the left is identical to the previous screenshot.



1. Currently supported languages in BellaDati are: **English, German and Czech**
BellaDati platform is however very flexible to add new languages - please contact [support](#) to get actual status of language options.
2. *Language detection: German for Germany and Austria, Czech for Czech republic and Slovakia, English otherwise.*

Charts display technology

BellaDati supports two basic technologies to render charts:

- **HTML5**
- **Flash**
- Auto detection: HTML5 is preferred all time when supported browser is detected.

The screenshot shows the 'Users' configuration page in BellaDati. The user profile for 'Retail Banking' is displayed. The 'Configuration' section at the bottom includes a 'Preferred language' dropdown set to 'English', a 'Charts display technology' dropdown set to 'Auto detection' (highlighted with a red arrow), and a 'Time zone' dropdown. A 'Save' button is at the bottom left.



1. **We strongly recommend to use HTML5 technology for charts rendering.** Flash technology is considered obsolete and has worse performance than HTML5 technology. *Therefore flash is usually necessary only for backward compatibility on those corporate PCs with only Internet Explorer 8 or older installed.*
Required browsers with HTML5 support: Microsoft Internet Explorer 9, Mozilla Firefox 5+, Google Chrome
2. Browser's compatibility with chart rendering technologies is always checked during login. A warning message is displayed when a problem has been found.
3. Changing chart render technology may require logout and login to BellaDati again to take effect.


Actions

- **User group edit:** Add groups via auto-complete, remove user from groups.
- **Change password:** Change user's password to specified here.
- **User roles:** You can assign or remove the roles here.
- **User groups:** You can assign or remove the user from user groups.
- **Password reset:** Generate new password and send it to user here.
- **Deactivate / Activate:** Temporarily suspends or activates the user account. That user cannot login to BellaDati however all
- **Delete user:** Removes the user from the domain.

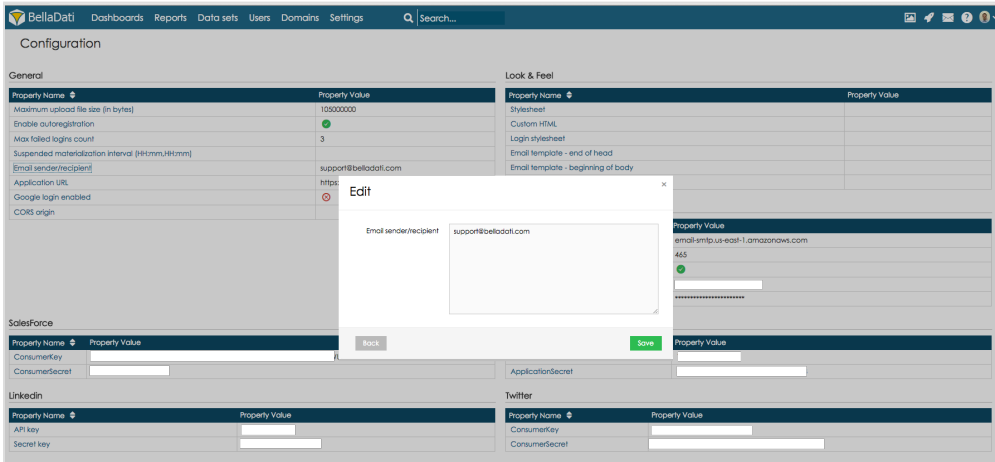


When an user is deleted, all his data sets will be assigned to domain administrator (he becomes its owner). All dashboards and reports owned by the deleted user will be deleted too!

Managing Configuration

 This feature is available for on-premise installations only.

In **BellaDati version 2.7.5 and higher** you can set all relevant configuration parameters directly from the user interface. Configuration page is available for users with **domain administrator** role from the **Settings** top menu.



You can configure the following parameters:

General settings

Application name	Application name shown in header, emails, exports.
Maximum upload file size (in bytes)	Maximum size of uploaded file. Default value: 2000000
Max failed logins count	Maximum failed logins count. Default value: unlimited
Suspended materialization interval (HH:mm,HH:mm)	Defines the interval, when the materialization of joined data set is suspended. Default value: undefined
Email sender/recipient	The email used as the email sender and recipient for contact form submissions. Default value: support@belladati.com

Look & Feel settings

Stylesheet	Edit CSS which will be used for whitelabelling .
Custom HTML	Edit global custom HTML.
Login stylesheet	Edit CSS which will be used in login page.
Email template - end of head	Edit HTML which will be placed at the end of email head.
Email template - beginning of body	Edit HTML which will be used in the beginning of email body.
Email template - end of body	Edit HTML which will be used at the end of email body.

Email server

Address	SMTP server address
---------	---------------------

Port	SMTP server port. Default value: 25
Ssl	SSL enabled. Default value: false
Username	Username if need
Password	Password

SalesForce

ConsumerKey	Consumer key
ConsumerSecret	Consumer secret

Facebook

ApplicationId	Application ID
ApplicationSecret	Application secret

Twitter

ConsumerKey	Consumer key
ConsumerSecret	Consumer secret

Intuit

ConsumerKey	Consumer key
ConsumerSecret	Consumer secret
AppToken	Application token

Setup Active Directory Authentication



You have to be the **domain admin** in order to configure the Active Directory authentication.

To setup the Active Directory authentication, login as domain admin and open Settings -> Active Directory.

To build the Active Directory connection, BellaDati needs following parameters:

Name	Description
Name	The name of the authentication domain. This name will appear on the login page.
URL	The URL of the LDAP service. Search tree should be included. E.g. ldap://hostname:389/OU=ou,DC=ad,DC=belladati,DC=com
BindDN	DN of the user able to search the organization tree.
BindDN password	Password of the BindDN user.
Search attribute	The attribute to be searched. This value is corresponding to the username the user enters on the login screen. E.g. sAMaccountName.
Create not existing accounts	If this option is selected, users those are not created in BellaDati will be created after successful authentication against Active Directory.


If there are more AD accounts in the domain. User can choose which account will be used as default.

AD attribute mapping

Once the connection is setup, you can create mapping where will be set which user fields from Active Directory will be imported to which user field in BellaDati. Fields that can be defined are: Email, Name, Surname, Business Phone, Mobile, Job Title, Office.

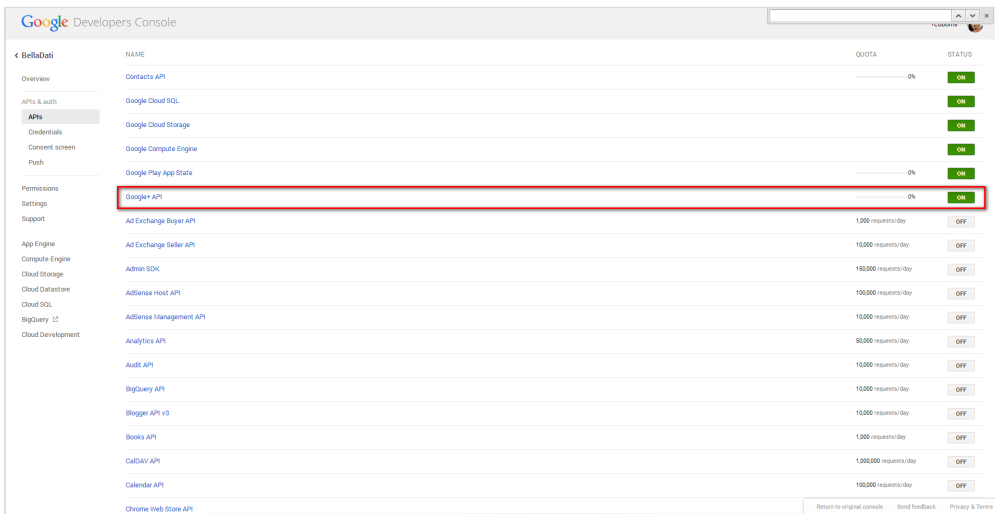
Setup Google login

BellaDati can be configured to support Google OAuth2 users log-in.

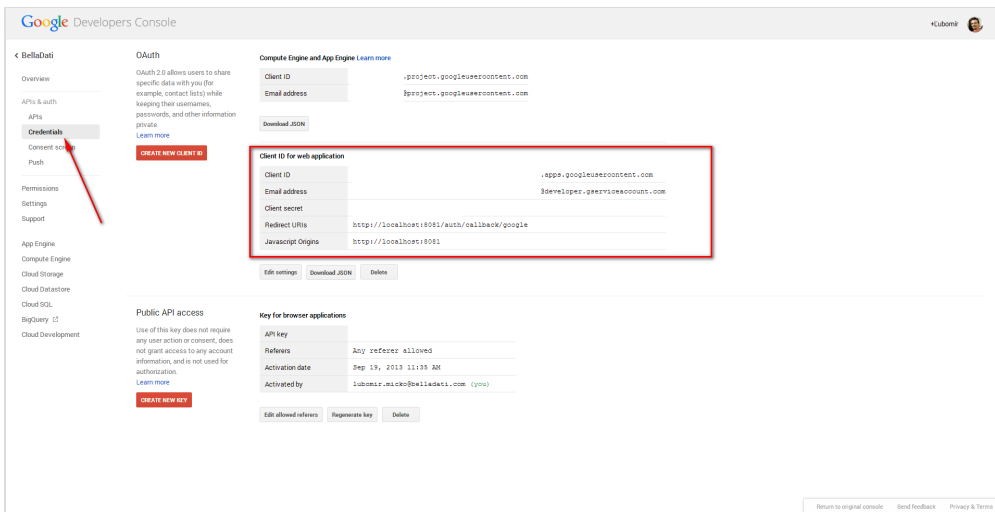
 This procedure is applicable in on-premise installations only.

Prerequisites

1. Existing **Google Developers account** in <https://console.developers.google.com/project> and project created.
2. Enabled access to **Google+ API**



3. Existing client application credentials (**web application**). Context for callback url is **/auth/callback/google**.



4. If your BellaDati instance is running behind the **proxy**, you have to setup the JVM parameters **-Dhttp.proxyHost**, **-Dhttp.proxyPort**, eventually **-Dhttp.proxyUsername** and **-Dhttp.proxyPassword**. BellaDati server must have the access to the following domains: **www.googleapis.com** and **accounts.google.com**.

Enabling Google log-in

In order to enable the Google log-in on BellaDati log-in page, enter the application as **Domain administrator**, open the Application settings page, and do the following:

1. Provide Google **Client ID** and **Client Secret** parameters
2. Enable Google log-in

BellaDati | Dashboards | Reports | Data sets | Users | Domains | Settings | Search...

Configuration

General

Property Name	Property Value
Maximum upload file size (in bytes)	3000000
Enable authentication	<input checked="" type="checkbox"/>
New field login count	
Suspended maintenance interval (in minutes)	
Email sender frequency	support@belladati.com
Application URL	http://localhost:8081
Google login enabled	<input checked="" type="checkbox"/>

Look & Feel

Property Name	Property Value
StyleSheet	
Login stylesheet	

SFTP

Property Name	Property Value
Host	
Port	
SFTPFS	
Username	
Password	

Facebook

Property Name	Property Value
ApplicationID	
ApplicationSecret	

Twitter

Property Name	Property Value
ConsumerKey	
ConsumerSecret	

RAMA

Property Name	Property Value
Host	
Port	
Username	
Password	

Google

Property Name	Property Value
API Key	
Secret Key	

Salesforce

Property Name	Property Value
ConsumerKey	
ConsumerSecret	

LinkedIn

Property Name	Property Value
API Key	
Secret Key	

Intuit

Property Name	Property Value
ConsumerKey	
ConsumerSecret	
AppToken	


Zendesk

Property Name	Property Value
Secret	

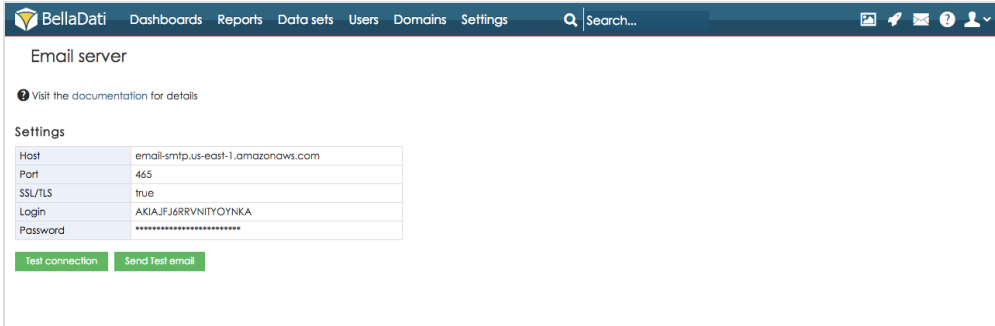
⚠ Make sure, that the **Application URL parameter** matches the URL you have entered in Google Developers Console.

⚠ Internet Explorer users must disable the pop-up blocking feature.

Email server configuration

 This feature is available for on-premise installation only.

In **BellaDati version 2.7.11.4** or higher is available Email server configuration page which allows you to check if the connection parameters are set correctly.



BellaDati Dashboards Reports Data sets Users Domains Settings Search...

Email server

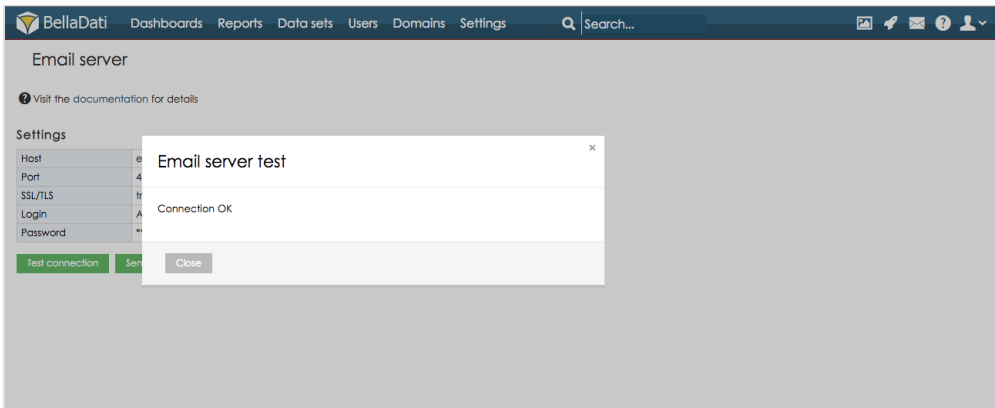
Visit the documentation for details

Settings

Host	email-smtp.us-east-1.amazonaws.com
Port	465
SSL/TLS	true
Login	AKIAJFJ6RRVNTYOYNKA
Password	*****

Test connection Send Test email

Click button *Test connection* to check if the connection parameters are set correctly. New dialogue window will show you result of the test.



BellaDati Dashboards Reports Data sets Users Domains Settings Search...

Email server

Visit the documentation for details

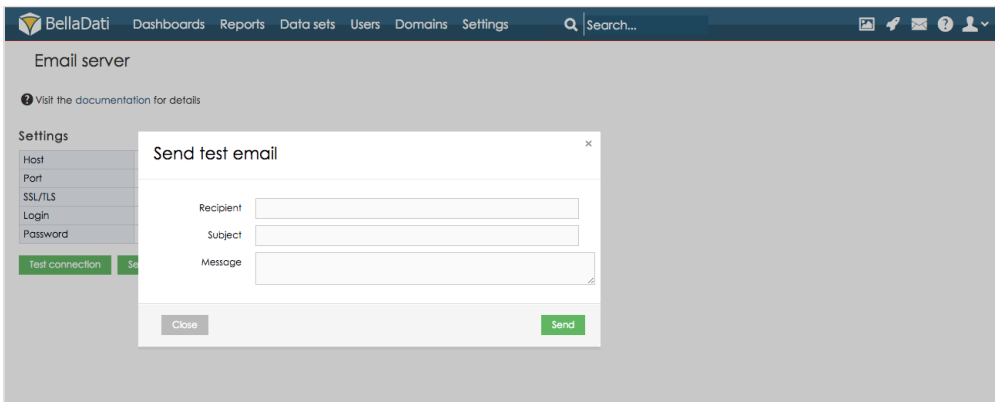
Settings

Host	email-smtp.us-east-1.amazonaws.com
Port	465
SSL/TLS	true
Login	AKIAJFJ6RRVNTYOYNKA
Password	*****

Test connection Send Test email

Email server test
Connection OK
Close

Click button *Send Test email* to open dialogue window where can be set parameters of test email. You are allowed to set Recipient, Subject and Message. Click button *Send* to send the test email.



BellaDati Dashboards Reports Data sets Users Domains Settings Search...

Email server

Visit the documentation for details

Settings

Host	email-smtp.us-east-1.amazonaws.com
Port	465
SSL/TLS	true
Login	AKIAJFJ6RRVNTYOYNKA
Password	*****

Test connection Send Test email

Send test email
Recipient
Subject
Message
Close Send

Twitter Authentication Setup

In order to setup Twitter authentication, you need to setup your own Twitter application at apps.twitter.com.

Create an application

Application Details

Name *

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.


Website *

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens. (If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL

Where should we return after successfully authenticating? [OAuth 1.0a](#) applications should explicitly specify their `oauth_callback` URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

Make sure that in field Callback URL is used the same value as in your [BellaDati Configuration](#) (field Application URL).

 In case you have BellaDati installed on your localhost, set <http://127.0.0.1:8080> as Callback URL.

Once this app is created, select tab Key and Access Tokens and copy values from fields Consumer Key (API key) & Consumer Secret (API Secret) into BellaDati configuration settings.

Application Management

To bring you Twitter, we and our partners use cookies on our and other websites. Cookies help personalize Twitter content, tailor Twitter Ads, measure their performance, and provide you with a better, faster, safer Twitter experience. By using our services, you agree to our [Cookie Use](#).

BellaDati test 2

[Details](#) [Settings](#) [Keys and Access Tokens](#) [Permissions](#)

Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key)	gANNCVLJdgxPYTVNlwwyGfMa
Consumer Secret (API Secret)	OrLiHatSUR6nLLJZmnRbtrz8PxaSjnxEZRI7XqYTzWTkzEU
Access Level	Read and write (modify app permissions)
Owner	Akclm114
Owner ID	2915812619

Application Actions

[Regenerate Consumer Key and Secret](#) [Change App Permissions](#)

BellaDati Documentation, Version 2.7.11.5

Page 358 of 370

Folders Dashboards Reports Data sets Users Settings
search...

General

Property Name	Property Value
Application name	BellaDati
Maximum upload file size (in bytes)	999999999999
Max failed logins count	100
Suspended initialization interval (minutes)	
Email sender/reply email	support@belladati.com
Application URL	http://127.0.0.1:8080
Google login enabled	<input checked="" type="checkbox"/>
CDN origin	

Look & Feel

Property Name	Property Value
Stylesheet	
Custom HTML	
Login stylesheet	
Email template - end of head	
Email template - beginning of body	
Email template - end of body	

SMTP server

Property Name	Property Value
Host	smtp.gmail.com
Port	465
SSL/TLS	<input checked="" type="checkbox"/>
Login	radek.micka@belladati.com
Password	*****

Facebook

Property Name	Property Value
ApplicationId	
ApplicationSecret	

SalesForce

Property Name	Property Value
ConsumerKey	
ConsumerSecret	

Twitter

Property Name	Property Value
ConsumerKey	6ANNCHVIdgwpPTTYNwvyQfMa
ConsumerSecret	OrUkafSU8enLUzmrbnB8PkoasJvKEZ807kqYTDwtkoGJ

LinkedIn

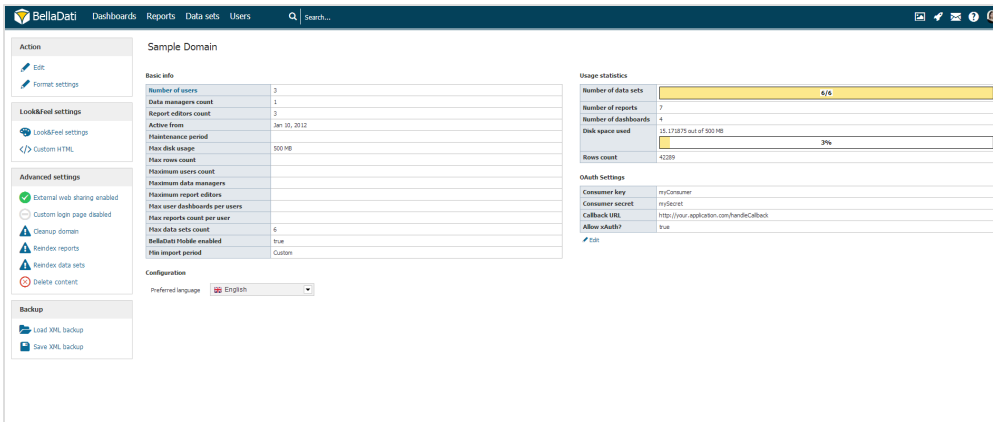
Property Name	Property Value
App key	
Secret key	

Application URL

Twitter configuration

Domain Configuration and Administration

Click on the domain name at top right corner of the main menu.



Available actions:

- **Edit:** Timezone change - affects the date and time displayed in BellaDati, however **data itself are not affected!** You can also change the name of the domain or set blank dashboard as default for new users.
- Add **Color schemes**.
- **Delete content:** All data sets, reports and dashboards will be deleted after confirming this action in the popup.
- **Disable web sharing:** Reports and dashboards will not be shareable via public links.
- **Re-index data sets and reports:** Performs the content re-indexing for full-text search
- **Cleanup domain:** Performs database level cleanup (deletes unused tables, performs vendor-specific cleanup procedure, e.g. vacuuming in PostgreSQL) and empties the caches.
- **Preferred language:**
 - **Use browser default settings:** BellaDati selects the actual language automatically according your browser's locale. Since BellaDati does not support all locales (see below), English will be the default language in most cases.
 - **Particular language** (represented by flag): BellaDati will be always displayed in the selected language regardless the domain or browser locale.



1. Currently supported languages in BellaDati are: **English, German, French, Chinese, Korean, Czech**
BellaDati platform is however very flexible to add new languages - please contact [support](#) to get actual status of language options.
2. *Language detection: German for Germany and Austria, Czech for Czech republic and Slovakia, English otherwise.*
3. Language selected in the [user's profile](#) has priority over this setting.



Please note, the screenshot above is only illustrative. Domain admin and data manager has access only to fewer information about the domain.



Information about occupied space may not be exact and actual. These information are updated asynchronously. Run cleanup domain to get most recent information.



Domain limits are set automatically according registered tariff or On-Premise license.

Application Settings and Configuration



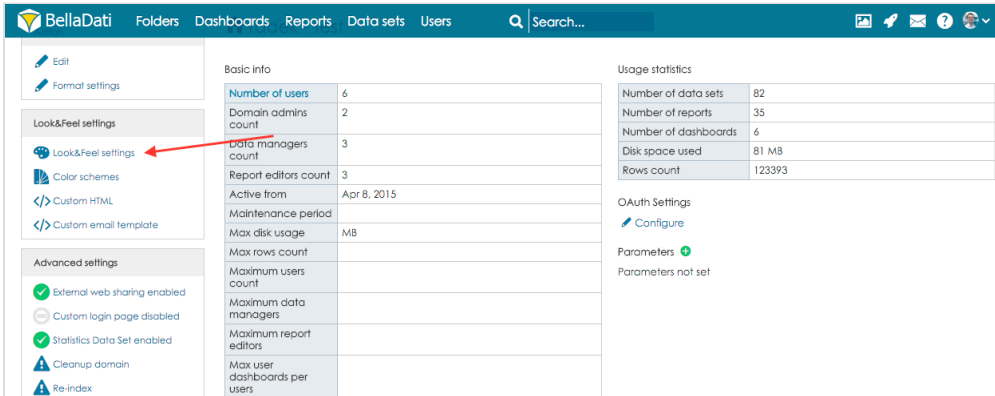
These settings are available in BellaDati On-Premise only.

Go to menu "Settings" - "Configuration". You have following parameters:

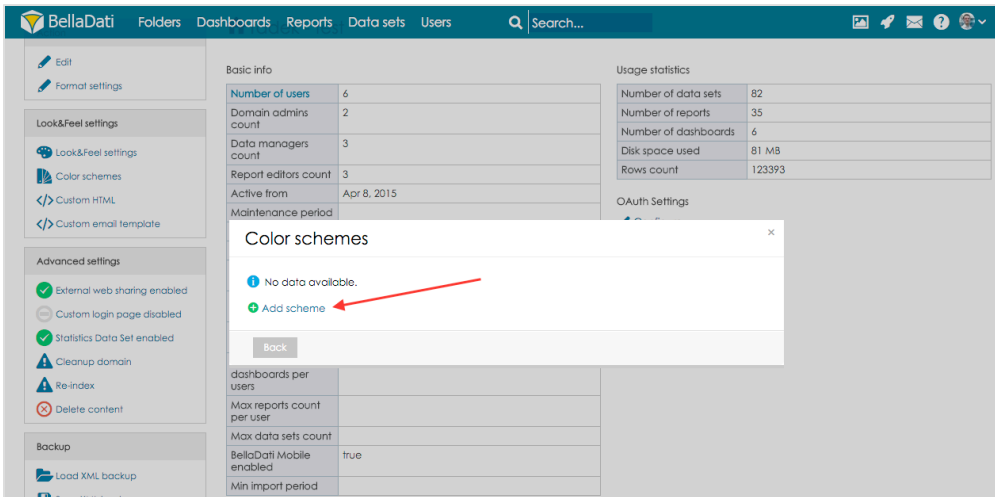
- **Maximum upload file size (in bytes):** default 20MB (21000000)
- **Max failed logins count:** default 3

Color schemes

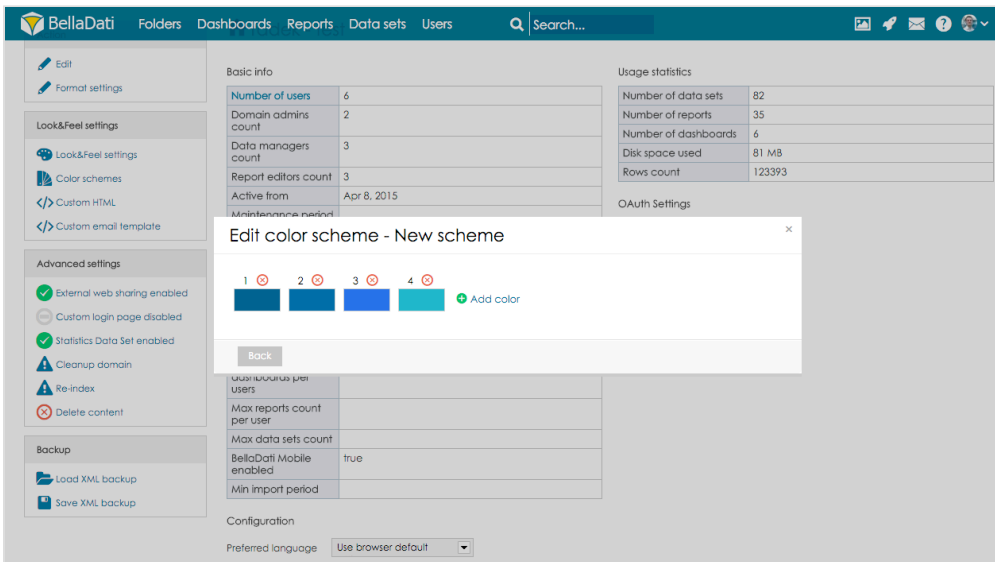
BellaDati allows domain administrators to set color schemes which can be selected in the reports. To create color scheme, select option Color schemes in section **Look&Feel settings**.



In newly displayed dialogue window hit a click on button **Add scheme**



Give a name to the new scheme and hit a click on button **Add scheme**. Now hit a click on button **Add color** to add new color to the color scheme. You can add as many colors to the color scheme as you want.



Domain Banners (Cover Pictures)

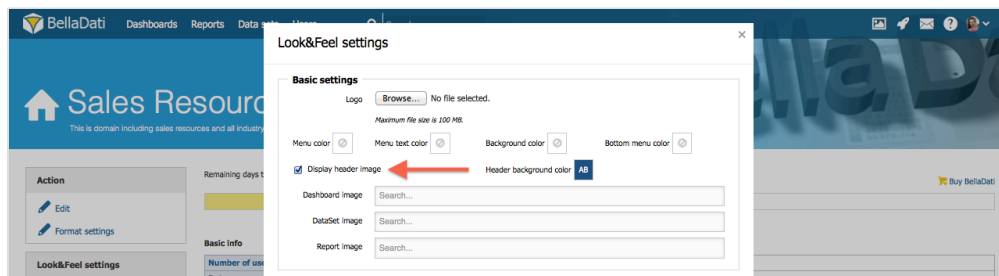
BellaDati enables you to enrich its interface with large **banner pictures**.




Banner pictures are turned off by default. You need to have domain administrator role assigned to be able to modify cover pictures.

To set up banner picture,

- Go to **Domain Settings**
- Navigate to **Look & Feel**
- Check **Display header image**
- Choose **Header background color**
- Specify **images** for Dashboard, Data Set and Report pages

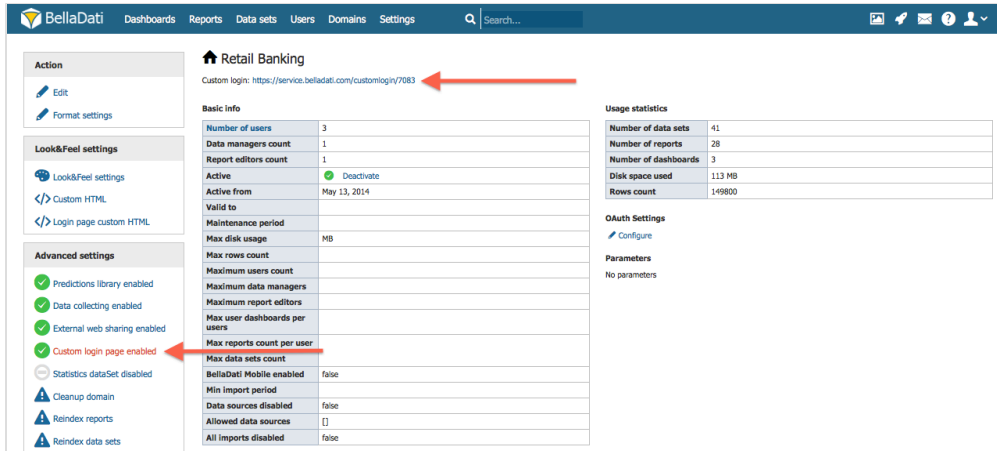


External Custom Login Page

 You have to be domain administrator to create and edit custom login page.

BellaDati allows you to create, edit and leverage **custom login page** for your domain. To enable **custom login page**, click on **Enable custom login** in the **Domain settings**.

BellaDati will generate **unique URL** for your custom login page.



BellaDati Dashboards Reports Data sets Users Domains Settings Search...

Action

- Edit
- Format settings

Look&Feel settings

- Look&Feel settings
- Custom HTML
- Login page custom HTML

Advanced settings

- Predictions library enabled
- Data collecting enabled
- External web sharing enabled
- Custom login page enabled**
- Statistics dataSet disabled
- Cleanup domain
- Reindex reports
- Reindex data sets

Retail Banking

Custom login: <https://service.belladati.com/customlogin/7083>

Basic info

Number of users	3
Data managers count	1
Report editors count	1
Active	Deactivate
Active from	May 13, 2014
Valid to	
Maintenance period	
Max disk usage	MB
Max rows count	
Maximum users count	
Maximum data managers	
Maximum report editors	
Max user dashboards per users	
Max reports count per user	
Max data sets count	
BellaDati Mobile enabled	false
Min Import period	
Data sources disabled	false
Allowed data sources	
All imports disabled	false

Usage statistics

Number of data sets	41
Number of reports	28
Number of dashboards	3
Disk space used	113 MB
Rows count	149800

OAuth Settings

- Configure

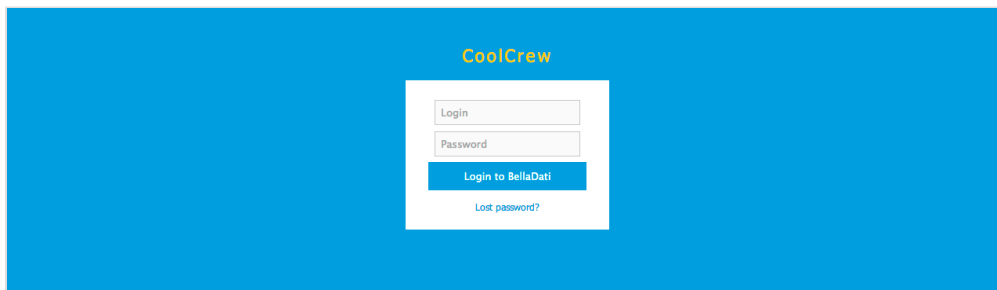
Parameters

No parameters


Editing HTML

You can edit and add your own HTML & CSS style sheets in *Custom HTML* dialog after clicking on **Login page custom HTML**.


Custom Login Example



Domain Backup

 You need to have **Domain Admin** role assigned to perform the backup.

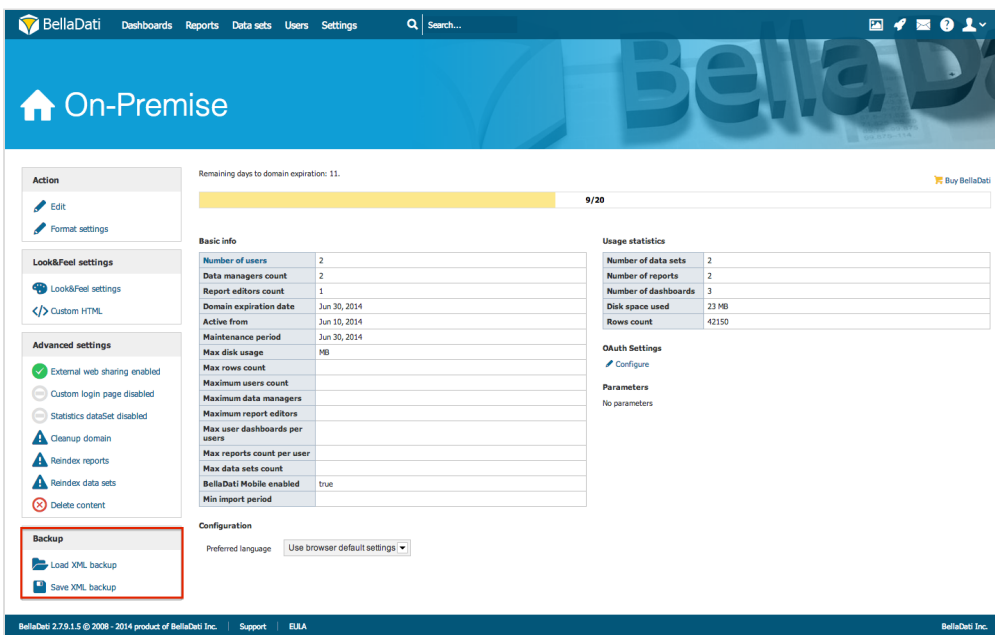
Domain backup allows you to store all domain information in XML format.

 Backup doesn't store data. You can [export](#) them manually from the Data Set or use one of the procedures described in [BellaDati Backup and Restore](#).

Saving XML Backup

To perform **Domain Backup**:

1. Navigate to **Domain** link in upper right corner of BellaDati window
2. Click **Save XML Backup** in left menu bar
3. **XML Backup** of your domain will be downloaded



The screenshot shows the BellaDati On-Premise settings interface. The left sidebar contains a 'Backup' section with 'Load XML backup' and 'Save XML backup' options. The main content area displays system statistics and configuration options.

Basic info	
Number of users	2
Data managers count	2
Report editors count	1
Domain expiration date	Jun 30, 2014
Active from	Jun 10, 2014
Maintenance period	Jun 30, 2014
Max disk usage	MB
Max rows count	
Maximum users count	
Maximum data managers	
Maximum report editors	
Max user dashboards per users	
Max reports count per user	
Max data sets count	
BellaDati Mobile enabled	true
Min import period	

Usage statistics	
Number of data sets	2
Number of reports	2
Number of dashboards	3
Disk space used	23 MB
Rows count	42150

Configuration

Preferred language: Use browser default settings

Loading XML Backup

To load **Domain Backup**

1. Navigate to **Domain** link in upper right corner of BellaDati window
2. Click **Load XML Backup** in left menu bar
3. Navigate to your **XML Backup**
4. Follow the wizard to restore your backup

BellaDati

Dashboards

Reports

Data sets

Users

Q

Search...

Edit

Format settings

Look&Feel settings

Look&Feel settings

Custom HTML

Login page custom HTML

Advanced settings

External web sharing enabled

Custom login page enabled

Statistics dataSet enabled

Cleanup domain

Reindex reports

Reindex data sets

Delete content

Backup

Load XML backup

Save XML backup

Basic info

Number of users	3
Data managers count	1
Report editors count	1
Active from	May 13, 2014
Maintenance period	
Max disk usage	MB
Max rows count	
Maximum users count	
Maximum data managers	
Maximum report editors	
Max user dashboards per users	
Max reports count per user	
Max data sets count	
BellaDati Mobile enabled	false
Min import period	

Configuration

Preferred language

Use browser default settings

Usage statistics

Number of data sets	42
Number of reports	28
Number of dashboards	3
Disk space used	114 MB
Rows count	149702

OAuth Settings

Configure

Parameters

No parameters

BellaDati Documentation, Version 2.7.11.5

Page 365 of 370

Usage Monitoring

BellaDati allows you to monitor usage of **reports and dashboards**. Statistics is available in **Usage monitoring** dataset. In order to turn on usage monitoring:

1. Go to **Domain Settings**
2. Click on **Statistics Dataset enabled**

Retail Banking
Custom login: <https://service.belladati.com/en/customlogin/7063>

Action
Edit
Format settings

Look&Feel settings
Look&Feel settings
Custom HTML
Login page custom HTML

Advanced settings
External web sharing enabled
Custom login page enabled
Statistics dataSet enabled
Cleanup domain
Reindex reports
Reindex data sets
Delete content

Basic info

Number of users	3
Data managers count	1
Report editors count	1
Active from	May 13, 2014
Maintenance period	
Max disk usage	MB
Max rows count	
Maximum users count	
Maximum data managers	
Maximum report editors	
Max user dashboards per users	
Max reports count per user	
Max data sets count	
BellaDati Mobile enabled	false
Min import period	

Usage statistics

Number of data sets	42
Number of reports	28
Number of dashboards	3
Disk space used	114 MB
Rows count	149701

OAuth Settings
Configure

Parameters
No parameters

Configuration
Preferred language: Use browser default settings

BellaDati will generate **Usage monitoring** table in **Datasets**.

Usage monitoring - Browse data

Add record Filter Delete selected data Order by

	Date	Time	DashboardId	DashboardName	ReportId	ReportName
✓	6/20/2014	1:09:47 AM	15764	Default dashboard		
✓	6/20/2014	1:10:15 AM			35027	Cross-Sel Opportunities Analysis

This table contains following attributes:

- Date
- Time
- DashboardId
- DashobardName
- Host (user's IP address)
- ReportId
- ReportName
- Type (Open report, Open dahboard, Login, Logout)
- UserID
- Source (Web, Mobile)

Administering domains



Domains administrations is available for multi-domain license type only. **SuperAdmin** role is required.

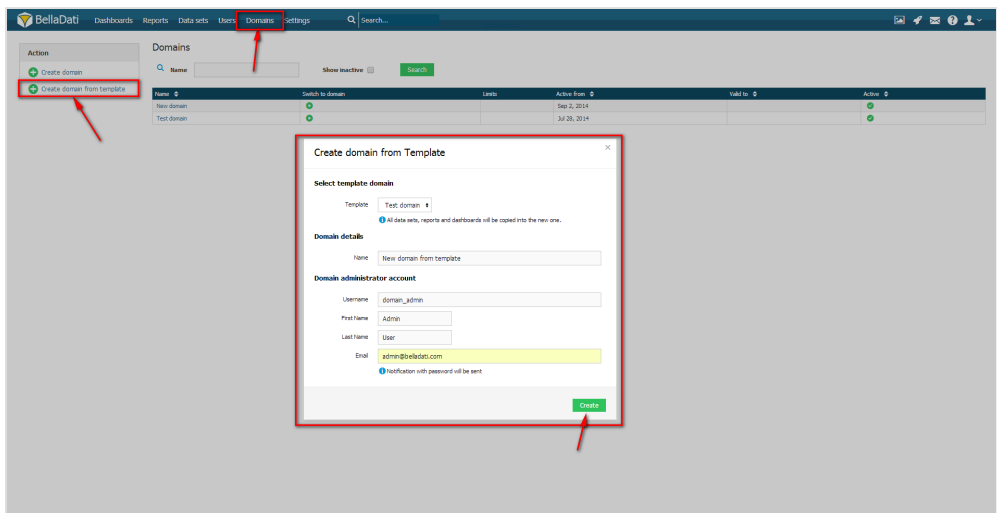
Creating domain from template

In multi-domain installations, administrators are able to create new domains based on the selected template domain.



This can speedup and simplify the process of preparing standardized domains for new customers.


New domain will contain all data sets, data sources, reports and dashboards like the selected template domain. Ownership of all of these objects will be transferred to the newly created domain administrator.

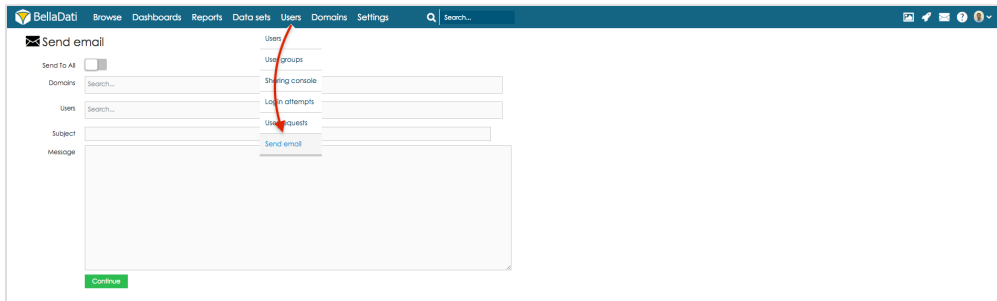


Mass mailing

Mass mailing serves allows you to send emails to all users in the domain (or across domains in multitenant environment).

To access mass mailing page, hover your mouse over menu item *Users* and choose option *Send email*.

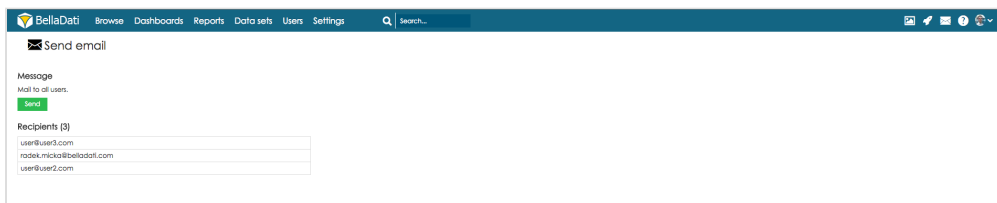
 Only domain administrator or global administrator have permissions to use mass mailing.




This page allows you to:

- Select checkbox **Send to all** to send email to all users in the domain (or all users across domains)
- Select **Domains** to which email will be sent
- Select **Users** to who the email will be sent
- Add **Subject** of the email
- Add **Message** for recipients

Click button **Continue** to see email overview.



On the email overview page, there you can see email message and all the recipients to who this email will be sent. Press button **Send** to send this email.

 Email is customised according to domain settings.

Report transferring

Sometimes it is required to transfer reports from one BellaDati installation to another, from test environment to production etc. There are several ways how to do this in BellaDati:

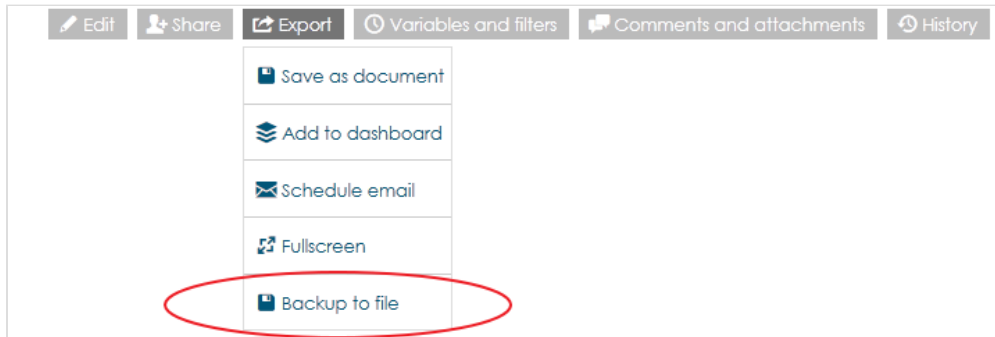
1. Report XML backup
2. Data Set backup
3. Domain backup
4. BellaApps

On this page:

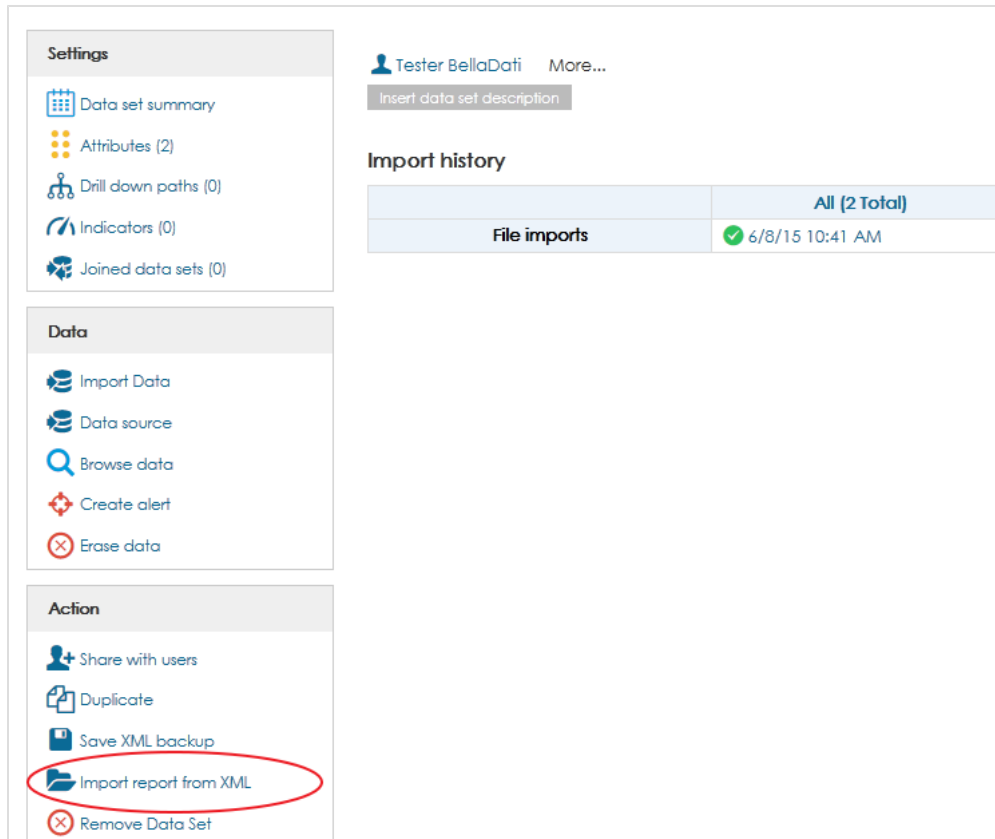
- [Report XML backup](#)
- [Data Set backup](#)
- [Domain backup](#)
- [BellaApps](#)

Report XML backup

Report XML backup is used for exporting the structure of the report. It contains the whole report **without data**. You can find the link **Backup to file** in top right menu under Export.



Click *Backup to file* to create report XML backup. XML file will be promptly downloaded to your hard drive. After that you can restore this report on your data set by clicking on **Import report from XML** in left menu on data set page.



Data Set backup

The following structures are included in this backup:

- **Data sets:** attributes, indicators, indicator groups and their settings (appearance, format, formulas), data set [owners](#), sharing settings
- Data source settings
- Alarms
- Joins
- **Reports:** Views (tables structure, chart types and their structure), report indicators, report [owners](#), sharing settings.

Data are not saved in this backup

Continue by [Structure Backup](#)

Domain backup

Domain backup allows you to store all domain information in XML format, except for data.

Continue by [Domain Backup](#)

BellaApps

BellaApp is a package of selected dashboards and reports with related data connectors (data sets), which can be easily shared with other users and uploaded into BellaDati. It also contains data, but only **first 10000 rows** from data set.

Continue by [BellaApps](#)