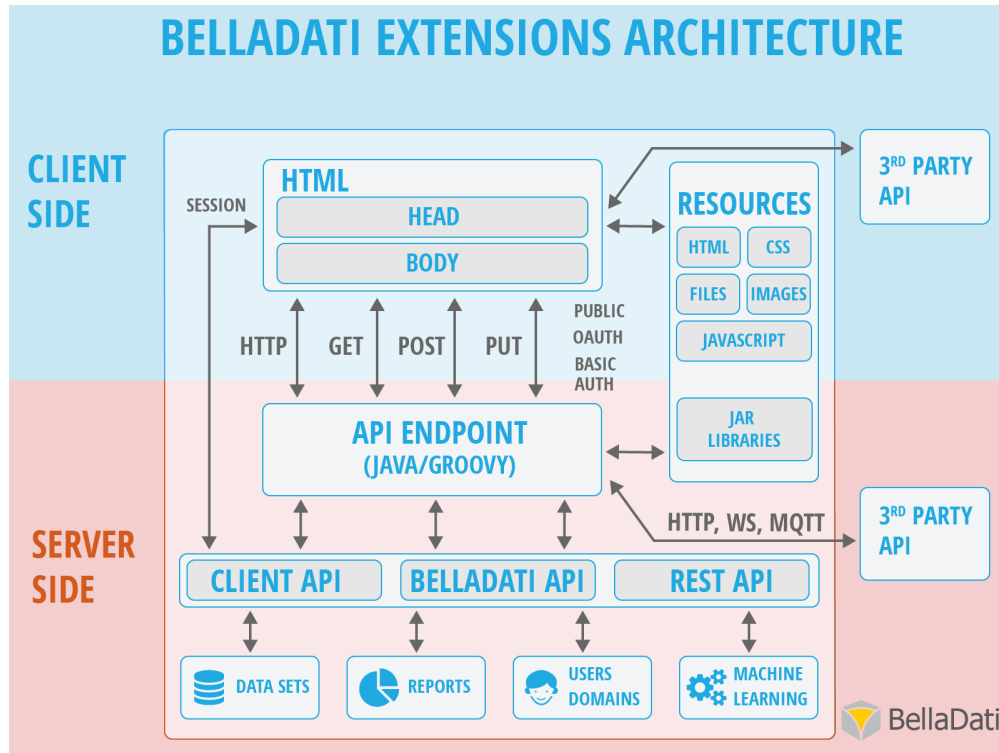


Extensions API (on server)

Architecture

From runtime perspective, each extension is composed of 3 major building blocks:

1. **API endpoints** - running on server (server-side), having access to BellaDati API, REST API and providing methods to connect to 3rd party systems. Programming language is Groovy/Java. Endpoints are accessible over HTTP using HTTP GET, POST, PUT, DELETE methods and have optional support for authentication (oAuth, Basic auth) and session.
2. **HTML content** - provided by server, but running on client (client-side). Consists of HTML page (head and body) and web resources (Javascript, CSS, images, files, ...)
3. **Resources** - referencable resources for server-side (JAR libraries) and client-side (Javascript, CSS, images, files)



Setup of BellaDati-Extension-API dev environment

1. In your development environment setup following Maven dependency. You can find all available BellaDati-extension-api at <https://search.maven.org/artifact/com.belladati/extensions-api>

```
<dependency>
<groupId>com.belladati</groupId>
<artifactId>extensions-api</artifactId>
<version>1.1</version>
</dependency>
```

and follow Usage from GitHub repository <https://github.com/BellaDati/belladati-extensions-api>.

2. Get familiar with [uploading Extensions zip files](#) or creating extensions in BellaDati GUI

Javadoc for BellaDati-Extension

<http://api.belladati.com/extensions-api/latest/>

BellaDati Extension types

There are 4 types of extensions, each targeting different functionality and goal:

- **New page** - generic extension, which gets own URL and can optionally provide API endpoints.

 Use it when building java web application that is utilizing BellaDati features and expands them with custom business logic

- **Custom menu** - custom navigation HTML inside regular BellaDati UI
- **Table renderer** - Javascript table renderer with custom CSS
- **Chart renderer** - Javascript chart renderer with custom CSS

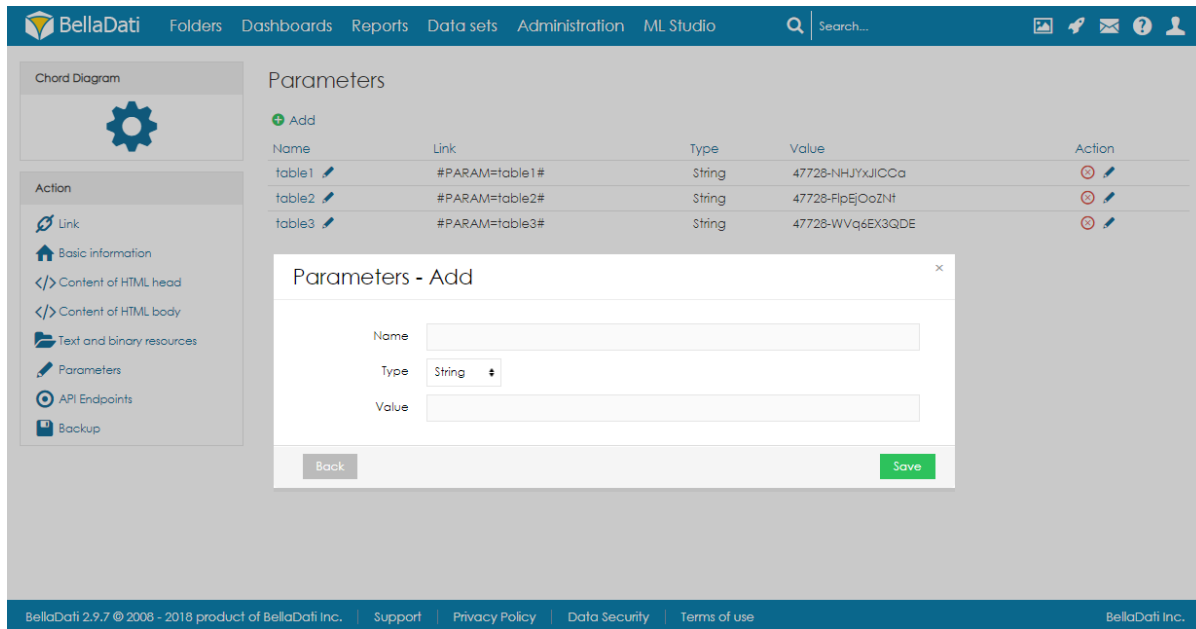
Parameters

Each extension can have one or more parameters. They can be used for customization or parametrization of the extension. By using parameters, it is possible to deploy the same extensions in multiple environments without the need to edit the code of the extension itself. In a multi-domain environment, parameters can be also changed by domain administrators.







Examples of parameters usage:

- loading of API keys of external services
- configuring usernames, data set IDs etc.
- enabling or disabling functions of extension

Parameters are created in the configuration of the extension. Each parameter has a name, value and a link. The link is used in the code of the extension to load the value of the parameter. It can be used in the HTML code, in the Endpoint or in the Resource (e.g. Javascript file).



The screenshot displays the BellaDati web interface for managing parameters. The top navigation bar includes 'BellaDati', 'Folders', 'Dashboards', 'Reports', 'Data sets', 'Administration', and 'ML Studio'. The main content area is titled 'Parameters' and features a table with the following data:

Name	Link	Type	Value	Action
table1	#PARAM=table1#	String	47728-NHJYxJICCa	 
table2	#PARAM=table2#	String	47728-FlpEJOoZNI	 
table3	#PARAM=table3#	String	47728-WVq6EX3QDE	 

Below the table is a 'Parameters - Add' modal form with the following fields:

- Name:
- Type: String (dropdown menu)
- Value:

The modal includes 'Back' and 'Save' buttons.

Example of parameter usage:

JavaScript

HTML

Tutorials

This section includes tutorials how BellaDati Extensions can be created and used. Each tutorial contains:

- steps how to achieve some goal
- partial source codes
- final extension package which can be imported into your own instance of BellaDati

Please see following child pages for various tutorials:

- [Creating API Endpoint](#)

- [Creating custom table renderer](#)
- [Creating custom chart renderer](#)
- [Creating advanced custom table](#)
- [Exporting public Report to PDF](#)
- [Exporting public Report to XLS](#)
- [Extensibility of BellaDati](#)