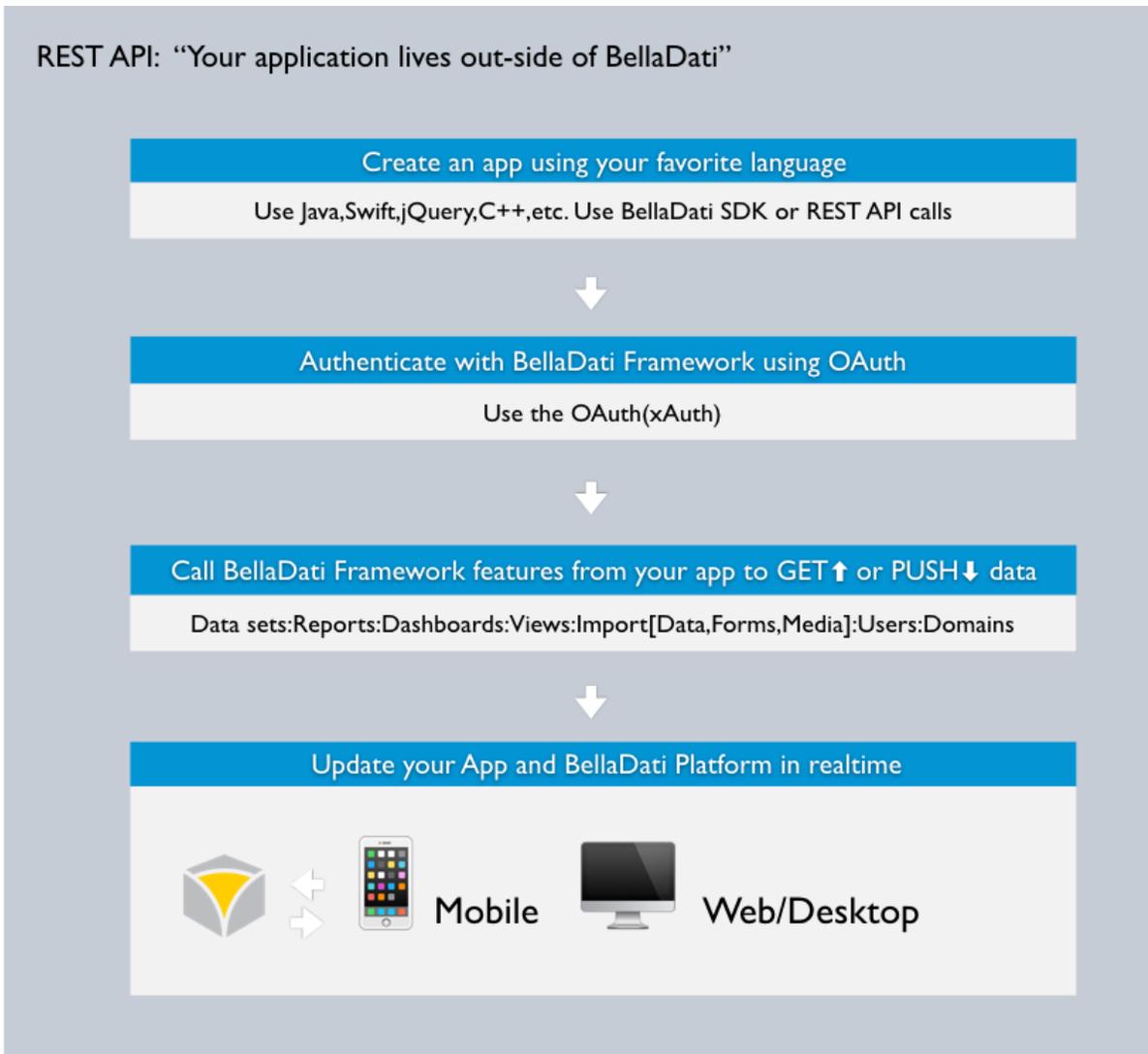# REST API

The REST API allows other applications to directly access data provided by BellaDati and is the underlying interface for our mobile Business Intelligence BellaDati Mobile. Through the API, users can retrieve reports, charts and even entire dashboards to integrate with their own custom client application.


General Notes

Enable REST API

Before you can access the REST API, you need to enable it in your domain configuration.

1. Open your domain configuration page. To reach this page, move the mouse over your name in the upper right corner and click on the domain link.
2. Under **OAuth Settings**, click **Configure**.
3. Enter a **Consumer Key** and a **Consumer Secret**. You can ignore the other settings for now.
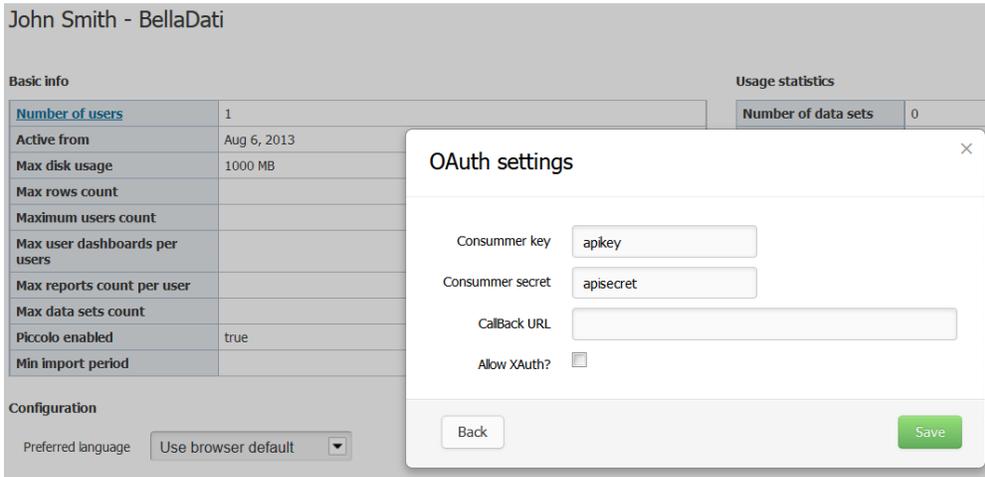
Base URL

The base address to access the REST API is `https://service.belladati.com/api/`.

If you are using an on-premise deployment, it is `https://your-server/belladati/api/`.

SSL Only

All API requests to BellaDati cloud service must be sent over SSL.

With an on-premise deployment, although not mandatory, we strongly recommend using SSL for security reasons.

UTF-8 Encoding

Every string passed to and from the BellaDati REST API needs to be UTF-8 encoded. For maximum compatibility, normalize to Unicode Normalization Form C (NFC) before UTF-8 encoding.

Locale

BellDati REST API uses the locale parameter to specify language settings of content responses. If you want to retrieve data in a language other than English, insert the appropriate IETF language tag. When a supported language is specified, BellaDati will return translated content where applicable.

Response Format

The BellaDati REST API uses the JSON format for any responses to API calls. And additionaly some responses can return HTML and PNG format for pictures.

Error Handling

Errors are returned using standard HTTP error code syntax. Any additional info is included in the body of the return call in JSON format. Error codes not listed here are described in the respective REST API method.

Standard API HTTP errors are:

Code	Description
400	Bad input parameter. Refer to the error message to find out which one and why.
401	Bad or expired token. This can happen if the access token has expired or is otherwise invalid. To fix this, re-authenticate the user.
403	Bad OAuth request (incorrect consumer key, bad nonce, expired time stamp...). Refer to the error message for details.
404	File or folder not found at the specified path. Check if the URL you're trying to access is correct.

405	Unexpected request method. The request method should be GET or POST depending on the request you're trying to make.
415	Unsupported Media Type. The server refused to accept the request because the payload format is in an unsupported format.
503	Your app is making too many requests and is being rate limited. 503s can trigger on a per-app or per-user basis.
5xx	Server error. Refer to the error message for details.

Application runtime errors are:

Code	Description
request_token_unauthorized	Passed request token has been unauthorized and cannot be used. Create new request token.
request_token_rejected	Request token didn't pass the validation. It may be in wrong format or corrupted.
access_token_already_issued	Access token was already issued for the passed request token.
invalid_consumer	Consumer key was not found or not present in the request. Check the domain settings and logs files.
request_token_not_found	Request token was not present or not found while trying to exchange it for access token.
token_not_found	Access token was not present or not found in the request. This may mean, that other prerequisites like request integrity and consumer key are fullfiled.
token_unauthorized	Passed access token has been unauthorized and cannot be used. Create new request token.
parameter_absent	Mandatory parameter is absent in the request.
user_not_found	User passed in xAuth request (x_auth_username) was not found.
too_many_login_failures	Too many login failures for user/password combination while using xAuth.
user_login_failed	User/password combination is not valid while using xAuth.
user_not_active	Passed user is not active while using xAuth.
user_expired	Passed user account is not active while using xAuth.
no_domain	User is not associated with a domain.
domain_expired	User domain validitiy expired while using xAuth.
api_disabled	REST API is not enabled in the license key.

- Access Control
- REST API Resources
- Types and enumerations