

Vendor Integration REST API Specification

Overview

This document defines a set of REST APIs that enables PayPal to notify the creation of new localization projects and to update their statuses. The API based communication is intended to improve the current communication which is based on emails, phone calls, or other disorganized means, and to reduce the amount of manual work associated with those communications.

The APIs are implemented by each vendor. PayPal calls the APIs when a project is created for a vendor and its statuses are changed. The vendor can then store the project information to plan and initiate work, keep track of the progress, generate reports, and so on.

As the APIs are implemented by vendors, this specification doesn't define their URLs or resource identifiers. Vendors are free to choose any endpoints and API paths. However, the API protocol must be HTTPS as the information sent via APIs contain sensitive information. Vendors will need to register their API URLs with PayPal (typically by email) so that PayPal's localization automation platform knows what to call.

APIs

Authentication

PayPal InfoSec requires us to use TLS 1.2 mutual authentication for API calls with external servers. TLS mutual authentication is also known as two way auth. The usual HTTPS call is one way auth. It only proves the identity of the server to the client using X.509 certificate. Two way auth authenticates the client to the server as well by requiring the client to send a client side X.509 certificate to the server.

Our implementation should work as below.

- Server (vendor side) is configured so that the TLS handshake is two way authentication for all API calls.
- In the TLS handshake, server sends a X.509 certificate to the client (PayPal) that is signed by a well-known CA (Certificate Authority) such as Symantec, Comodo, etc. The client validates the server cert against the CA certs available in the OS or in the client framework.
- Client (PayPal) sends a client-side X.509 certificate that is signed by a private CA. PayPal provides the private CA cert to vendors. Vendor installs the cert in the server for authenticating client. Server must authenticate client by using only this CA so that only PayPal client can be authenticated.

With TLS mutual authentication, the client authentication is taken care of by the TLS protocol. The application level authentication, such as logging in by user credential, is not necessary.

Here are some links that might help understand TLS mutual auth:

- <http://www.robinhowlett.com/blog/2016/01/05/everything-you-ever-wanted-to-know-about-ssl-but-were-afraid-to-ask/>

- <https://www.codeproject.com/Articles/326574/An-Introduction-to-Mutual-SSL-Authentication>

³ Notifying Project Creation

When a new localization project is created and it is assigned to a vendor, this API provided by the said vendor is called.

- **HTTP Method** - POST
- **Request Body** - JSON object containing the followings:
 - Project Id - unique project id
 - Project Name - may not be unique
 - Source Language - in BCP 47, e.g. en-US
 - Target Language - in BCP 47, e.g. ja-JP
 - Word Count
 - Creation Date - project creation date in UTC in ISO 8601 format
 - Due Date - project due date in UTC in ISO 8601 format
 - URL to the project
 - Project Type - possible values are as follows:
 - translation - translation project
 - linguisticReview - linguistic review project
 - Status - status of the project. Possible values are as follows:
 - inProgress - initial status
 - canceled
 - completed
 - Example:

```
{
  "projectId" : "2704937",
  "projectName" : "6686570_P2P_Core_P2P_Core_p2pnodeweb_develop_el_GR",
  "sourceLanguage": "en-US",
  "targetLanguage": "el-GR",
  "wordCount": {
    "ICE": 198,
    "exact": 3,
    "99-80%": 12,
    "79-70%": 24,
    "69-0%": 2085,
    "repetition": 346
  },
  "creationDate": "2017-10-14T14:17:27.396Z",
  "dueDate": "2017-10-18T16:00:00.000Z",
  "projectUrl": "https://www.matecat.com/translate/translate.rtf/en-US-el-GR/97",
  "projectType": "translation",
  "status": "inProgress"
}
```

- **Response Body** - JSON object containing Google account email addresses of people who are authorized by the vendor to work on the project. Only the people who are listed here can

access the project. The list can be different project by project. Vendors are responsible for maintain the trusted linguists.

- Example:

```
{
  "authorizedUsers": [
    "johndoe@gmail.com",
    "janedoe@gmail.com",
    "richarddoe@gmail.com"
  ]
}
```

- **Error** - Any responses with the HTTP response code other than 2xx are treated as error.

³ Notifying Project Update

This API is called when the project status is changed or due date is updated.

- **HTTP Method** - PUT
- **Request Body** - JSON object containing the followings:
 - Project Id - project id
 - Due Date - updated due date. This field is present only when the due date is updated.
 - Status - updated status. This field is present when the project is canceled or completed.
 - Completion Date - project completion or cancel date in UTC in ISO 8601 format
 - Example:

```
{
  "projectId" : "2704937",
  "completionDate": "2017-10-18T14:28:03.837Z",
  "status": "completed"
}
```

- **Error** - Any responses with the HTTP response code other than 2xx are treated as error.