REST Job.Submit.Translate

Purpose

Submits a translation job for processing in real-time or batch mode.

Call

HTTP Method	POST
Content Type	multipart/form-data

Parameters

requesttype	String	The type of call being made (case insensitive). This should be set to "**Job.Submit.Translate**".
apikey	String	Your API Key. See Authentication: Setting Up a Language Studio API User and API Key for more details.
domaincode	Integer	The domain code of the custom engine to be used for this translation
datatoprocess	String	The data that will be processed. If DataType = 1 (File) A list of file paths delimited by a | (pipe). **Note:** This is only needed in REST if you are submitting files that already exist in the Work Project folder. Use the ***file*** parameter If DataType = 2 (Text) A string to process
projectno	Integer	(Optional. Default = First project in the account.) The number of Work Project in account to run the job in.
advancedsettingsxml	String	(Optional. Default = Null) Contains an XML structure with additional settings.
file	File	(Optional. Default = Null) Contains 1 or more files that are being uploaded to a Work Project folder. Multiple files Only used to upload files from local storage to a Work Project folder. Use the ***datatoprocess*** parameter to specify files that are already in the Work Project folder. The ***datatoprocess*** parameter will be ignored if files are uploaded.

Advanced Settings XML Structure

```
<AdvancedSettings-Translate>
  <BatchName></BatchName>
  <ClientMetaData></ClientMetaData>
  <ClientRef></ClientRef>
  <ConfidenceScore>0</ConfidenceScore>
  <DataType></DataType>
  <DomainVersionNo></DomainVersionNo>
  <GenerateTMX>0</GenerateTMX>
  <GenerateXLIFF>0</GenerateXLIFF>
  <JobDetail></JobDetail>
  <JobMode>1</JobMode>
  <JobXML><![CDATA[]]></JobXML>
  <KeepUnknownTags>0</KeepUnknownTags>
  <LogJobSteps>0</LogJobSteps>
  <NotifyEmail></NotifyEmail>
  <NotifyURL></NotifyURL>
  <Priority></Priority>
  <ResultDetail></ResultDetail>
  <SentenceSegment>0</SentenceSegment>
  <TextDataTypeFileExtension>txt</TextDataTypeFileExtension>
  <OutputFiles>
    <File source="" target=""/>
  </OutputFiles>
</AdvancedSettings-Translate>
```

All parameters are optional.

Name	Description
BatchName	The name of batch. This is a user specified label that can be used to group files together. It can be very useful in reports. Default = Null
ClientMetaData	Client meta data that can be used by other applications and carried with the translations. The same data will be applied to each file. Default = Null. Only valid for JobMode=1
ClientRef	Client reference number or ID that can be used to associated data with external applications. Default = Null. Only valid for JobMode=1
ConfidenceScore	If True will return confidence scores. Default = False
DataType	Determines the type of data contained in DataToProcess. Valid values: 1 - File 2 - Text string Default = 1

DomainVersionNo	The version number of the custom engine to use when translating. Default = The version number specified in the custom engine domain settings. Only valid for JobMode=1
GenerateTMX	If True, a separate TMX file will be created when non-TMX documents are submitted for translation. Default = False. Only valid for JobMode=1
GenerateXLIFF	If True, a separate XLIFF file will be created when non-XLIFF documents are submitted for translation. Default = False
JobDetail	Determines the level of detail for the job when JobMode=2 (Real-Time) 1. Standard 2. Enhanced Default = 1. Only valid for JobMode=2
JobMode	The mode to run the job in. Valid values: 1. Batch - Jobs are submitted to the job queue and processed progressively. 2. Real-time - Jobs are processed and a response is immediate. **See notes below. Default = 1 Notes: • Diagnostic Review jobs can only be run from the Web Interface. • Real-time jobs require infrastructure deployed 24/7. This option is only available if you have made prior arrangements for real-time infrastructure. • Jobs will error if this DomainCode is not configured for real-time access on the server and real-time is specified.
JobXML	Additional custom advanced configuration settings. Talk to your Omniscien Technologies account manager for specific non-standard settings. Defaul = Null
LogJobSteps	If True then an additional log file will be produced with detailed step by step information that is useful for debugging. If JobMode = 1 (Batch) A detailed log file is produced. If JobMode = 2 (Real-Time) A summary is produced in the JSON format. Default = False
NotifyEmail	When specified will send an email on job completion or error. 3 values will be included in the email subject: • JobID • JobType • Status Default = Null. Only valid for JobMode=1

NotifyURL	When specified will connect to the specified URL on job completion or error. 3 values will be included on the QueryString: • JobID • JobType • Status Default = Null. Only valid for JobMode=1
Priority	The value of priority of this job (Requires permissions) Standard jobs run at priority 99. Jobs with a smaller number will take priority. Jobs with a higher number will be lower priority. Default = 99. Only valid for JobMode=1
ResultDetail	Determines the level of detail for results when JobMode=2 (Real-Time). 1. Translation text only 2. Translation text + Segment text (source and target) 3. Translation text + Segment text (source and target) + Word Alignment Default = 1. Only valid for JobMode=2
OutputFiles	Determines the file names of the output files when written to the project folder. A list of files with their paired source and target name. If not defined, then the file will be saved in the project folder with the naming convention: <filename>.out.<fileextension> Default = Null. Only valid for JobMode=1
KeepUnknownTags	If True, XML tags <unknown>word</unknown> will be kept in the output and not removed. Default = False - remove unknown tags
SentenceSegment	(Optional) Set "true" = Run segment the sentence before translate (Default value is "true")
TextDataTypeFileExtension	If DataType = 2 (Text String), this value will be used to determine how the text will be processed. Default = txt

Returns

String	JSON Response with job submission information.

Remarks

Translation jobs can be submitted as either batch-mode (default) or real-time.

Batch-Mode:

- One or more jobs can be submitted in a single request.
- Jobs will be queued and translated as soon as resources are available.
- Multiple concurrent jobs can translate at the same time.
- This is the most common option and is suitable for all kinds of translation except those that require a real-time response (i.e. chat).
- To check job status use the REST API Job.Status.
- To downloaded a complete job use the REST API Job.Download.
- Batch-mode may have a slight delay on the first document as a result of the time needed to load an engine on demand.

Real-Time:

- Only one request for real-time translation can be submitted per request.
- A request can be multiple sentences, but in general should not be a large number of sentences as this feature is designed for use cases like chat rather than documents.
- A response will be returned immediately.
- This option requires a real-time server to be allocated and always be on so that it can respond immediately without loading an engine.

Example

Request Examples

Example 1 - Submission of a single job in Batch mode

CURL

```
curl
-H "Content-Type: multipart/form-data"
-X POST
-F "requesttype=Job.Submit.Translate"
-F "apikey=Your_API_KEY"
-F "domaincode=1234"
-F "datatoprocess="
-F "file=@/path/to/file/TH-EN-1.txt"
-F "file=@/path/to/file/TH-EN-2.txt"
 https://<SERVER URL>
```

Query String

```
Not Supported
```

Java

```
private void JobSubmitTranslate(String APIKey, String DomainCode, String
DataToProcess) {
        org.apache.http.client.HttpClient client = new org.apache.http.
impl.client.DefaultHttpClient();
        org.apache.http.client.methods.HttpPost post = new org.apache.http.
client.methods.HttpPost(https://<SERVER URL>);
        org.apache.http.entity.mime.MultipartEntity entity = new org.
apache.http.entity.mime.MultipartEntity();
        java.nio.charset.Charset chars = java.nio.charset.Charset.forName
("UTF-8");
        // Set normal parameter
        org.apache.http.entity.mime.content.StringBody param1 = new org.
apache.http.entity.mime.content.StringBody("Job.Submit.Translate", chars);
        entity.addPart("requesttype", param1);

        org.apache.http.entity.mime.content.StringBody param2 = new org.
apache.http.entity.mime.content.StringBody(APIKey, chars);
        entity.addPart("apikey", param2);

        org.apache.http.entity.mime.content.StringBody param3 = new org.
apache.http.entity.mime.content.StringBody(DomainCode, chars);
        entity.addPart("domaincode", param3);

        // Set file to upload
        File oFile = new File(DataToProcess);
        if (oFile.exists()) {
                entity.addPart(“file”, new org.apache.http.entity.mime.
content.FileBody(oFile));
        }

        //Set entity
        post.setEntity(entity);

        // Post parameters to a server
        org.apache.http.HttpResponse response = client.execute(post);

        // Get response text
        org.apache.http.HttpEntity entityRes = response.getEntity();
        String returnValue = org.apache.http.util.EntityUtils.toString
(entityRes, "UTF-8");
        System.out.println(“Output=” + returnValue);
}
```

Example 2 - Submission of a multiple jobs in Batch mode

CURL

```
curl
-H "Content-Type: multipart/form-data"
-X POST
-F "requesttype=Job.Submit.Translate"
-F "apikey=Your_API_KEY"
-F "domaincode=1234"
-F "datatoprocess="
-F "file=@/path/to/file/TH-EN-1.txt"
-F "file=@/path/to/file/TH-EN-2.txt"
https://<SERVER URL>
```

Query String

```
Not supported
```

Java

```
private void JobSubmitTranslate(String APIKey, String DomainCode, String
DataToProcess) {
        org.apache.http.client.HttpClient client = new org.apache.http.
impl.client.DefaultHttpClient();
        org.apache.http.client.methods.HttpPost post = new org.apache.http.
client.methods.HttpPost(https://<SERVER URL>);
        org.apache.http.entity.mime.MultipartEntity entity = new org.
apache.http.entity.mime.MultipartEntity();
        java.nio.charset.Charset chars = java.nio.charset.Charset.forName
("UTF-8");

        // Set normal parameter
        org.apache.http.entity.mime.content.StringBody param1 = new org.
apache.http.entity.mime.content.StringBody("Job.Submit.Translate", chars);
        entity.addPart("requesttype", param1);

        org.apache.http.entity.mime.content.StringBody param2 = new org.
apache.http.entity.mime.content.StringBody(APIKey, chars);
        entity.addPart("apikey", param2);

        org.apache.http.entity.mime.content.StringBody param3 = new org.
apache.http.entity.mime.content.StringBody(DomainCode, chars);
        entity.addPart("domaincode", param3);

        // Set file to upload
        String[] Files = DataToProcess.split("[|]");
        for (String sFilePath : Files) {
                File oFile = new File(sFilePath);
                if (oFile.exists()) {
                        entity.addPart("file", new org.apache.http.entity.
mime.content.FileBody(oFile));
                }
        }

        // Set entity
        post.setEntity(entity);

        // Post parameters to a server
        org.apache.http.HttpResponse response = client.execute(post);

        // Get response text
        org.apache.http.HttpEntity entityRes = response.getEntity();
        String returnValue = org.apache.http.util.EntityUtils.toString
(entityRes, "UTF-8");
        System.out.println(“Output=” + returnValue);
}
```

Example 3 - Submission of a job in Real-Time mode

CURL

```
curl
-H "Content-Type: application/x-www-form-urlencoded"
-X POST
-d "requesttype=Job.Submit.Translate"
-d "apikey=Your_API_KEY"
-d "domaincode=1234"
-d "datatoprocess=This is a test. This is the second sentence."
-d "advancedsettingsxml=<AdvancedSettings-Translate><DataType>2<
/DataType><JobMode>2</JobMode></AdvancedSettings-Translate>"
https://<SERVER URL>
```

Query String

```
GET https://<SERVER URL>?requesttype=Job.Submit.
Translate&apikey=YOUR_API_KEY&domaincode=1234&datatoprocess=This+is+a+test.
+This+is+the+second+sentence.&advancedsettingsxml=%3CAdvancedSettings-
Translate%3E%3CDataType%3E1%3C%2FDataType%3E%3CJobMode%3E1%3C%2FJobMode%3E%
3C%2FAdvancedSettings-Translate%3E
```

Java

```
private void JobSubmitTranslate (String APIKey, String DomainCode, String
DataToProcess, String AdvancedSettingsXML)  {
        // set parameters
        String sParameters = "requesttype=Job.Submit.Translate&apikey=” +
URLEncoder.encode(APIKey, "UTF-8") + “&domaincode=” + URLEncoder.encode
(DomainCode, "UTF-8") + “&datatoprocess=” + URLEncoder.encode
(DataToProcess, "UTF-8") + “&advancedsettingsxml=” + URLEncoder.encode
(AdvancedSettingsXML, "UTF-8");
        //call RestAPI and receive output
        String returnValue = HttpPost(“https://<SERVER URL>”, sParameters);
    System.out.println(“Output=” + returnValue);
}

private String HttpPost(String webUrl, String parameters) throws Exception
{
        StringBuilder output = new StringBuilder();
        OutputStreamWriter writer = null;
        BufferedReader reader = null;
        InputStreamReader isReader = null;
        try {
                // Send data
                URL url = new URL(webUrl);
                URLConnection conn = url.openConnection();
                conn.setDoOutput(true);
                writer = new OutputStreamWriter(conn.getOutputStream());
                writer.write(parameters);
                writer.flush();
                // Get the response
                isReader = new InputStreamReader(conn.getInputStream(),
"UTF-8");
                reader = new BufferedReader(isReader);
                String line;
                while ((line = reader.readLine()) != null) {
                        output.append(line);
                }
                if (reader != null) { reader.close(); }
                if (isReader != null) { isReader.close();}
                if (writer != null) { writer.close();}
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
                if (reader != null) { reader.close(); }
                if (isReader != null) { isReader.close();}
                if (writer != null) { writer.close();}
        }
    return output.toString();
}
```

Example 4 - Submission of a job in Real-Time mode with ResultDetail = 2 and JobDetail=2

CURL

```
curl
-H "Content-Type: application/x-www-form-urlencoded"
-X POST
-d "requesttype=Job.Submit.Translate"
-d "apikey=Your_API_KEY"
-d "domaincode=1234"
-d "datatoprocess=This is a test. This is the second sentence."
-d "advancedsettingsxml=<AdvancedSettings-Translate><DataType>2<
/DataType><JobMode>2</JobMode><JobDetail>2</JobDetail><ResultDetail>2<
/ResultDetail></AdvancedSettings-Translate>"
https://<SERVER URL>
```

Query String

```
POST https://<SERVER URL>?requesttype=Job.Submit.
Translate&apikey=YOUR_API_KEY&domaincode=1234&datatoprocess=This+is+a+test.
+This+is+the+second+sentence.&advancedsettingsxml=%3CAdvancedSettings-
Translate%3E%3CDataType%3E1%3C%2FDataType%3E%3CJobMode%3E1%3C%2FJobMode%3E%
3CJobDetail%3E2%3C%2FJobDetail%3E %3CResultDetail%3E2%3C%2FResultDetail%3E
%3C%2FAdvancedSettings-Translate%3E
```

Java

```
private void JobSubmitTranslate (String APIKey, String DomainCode, String
DataToProcess, String AdvancedSettingsXML)  {
        // set parameters
        String sParameters = "requesttype=Job.Submit.Translate&apikey=” +
URLEncoder.encode(APIKey, "UTF-8")  + “&domaincode=” + URLEncoder.encode
(DomainCode, "UTF-8") + “&datatoprocess=” + URLEncoder.encode
(DataToProcess, "UTF-8") + “&advancedsettingsxml =” + URLEncoder.encode
(AdvancedSettingsXML, "UTF-8");
        //call RestAPI and receive output
        String returnValue = HttpPost(“https://<SERVER URL>”, sParameters);
    System.out.println(“Output=” + returnValue);
}

private String HttpPost(String webUrl, String parameters) throws Exception
{
        StringBuilder output = new StringBuilder();
        OutputStreamWriter writer = null;
        BufferedReader reader = null;
        InputStreamReader isReader = null;
        try {
                // Send data
                URL url = new URL(webUrl);
                URLConnection conn = url.openConnection();
                conn.setDoOutput(true);
                writer = new OutputStreamWriter(conn.getOutputStream());
                writer.write(parameters);
                writer.flush();
                // Get the response
                isReader = new InputStreamReader(conn.getInputStream(),
"UTF-8");
                reader = new BufferedReader(isReader);
                String line;
                while ((line = reader.readLine()) != null) {
                        output.append(line);
                }
                if (reader != null) { reader.close(); }
                if (isReader != null) { isReader.close();}
                if (writer != null) { writer.close();}
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
                if (reader != null) { reader.close(); }
                if (isReader != null) { isReader.close();}
                if (writer != null) { writer.close();}
        }
    return output.toString();
}
```

Response Examples

Example 1 - Submission of a single job in Batch mode

```
{
  "result": {
    "jobs": [
      {
        "job": {
          "file":"TH-EN-1.txt",
          "status": "1",
          "description": "Job successfully added to queue.",
          "jobid": "191260"
        }
      }
    ]
  },
  "duration": "134",
  "startdate": "20171221 08:37:33:934",
  "errortext": "",
  "errorno": "",
  "enddate": "20171221 08:37:34:068",
  "requestid": "20171221-083733.378-105.172.000001"
}
```


Example 2 - Submission of a multiple jobs in Batch mode

```
{
  "result": {
    "jobs": [
      {
        "job": {
          "file": "TH-EN-1.txt",
          "status": "1",
          "description": "Job successfully added to queue.",
          "jobid": "191260"
        }
      },
      {
        "job": {
          "file": "TH-EN-2.txt",
          "status": "0",
          "description": "Duplicate job in queue.",
          "jobid": ""
        }
      }
    ]
  },
  "duration": "134",
  "startdate": "20171221 08:37:33:934",
  "errortext": "",
  "errorno": "",
  "enddate": "20171221 08:37:34:068",
  "requestid": "20171221-083733.378-105.172.000001"
}
```


Example 3 - Submission of a job in Real-Time mode

```
{
  "result": {
    "jobs": [
      {
        "job": {
          "resulttext": "This is a test. This is the second sentence.",
          "description": "",
          "status": "1",
          "jobid": "312453"
        }
      }
    ]
  },
  "duration": "134",
  "startdate": "20171221 08:37:33:934",
  "errortext": "",
  "errorno": "",
  "enddate": "20171221 08:37:34:068",
  "requestid": "20171221-083733.378-105.172.000001"
}
```


Example 4 - Submission of a job in Real-Time mode with ResultDetail = 2 and JobDetail=2

```
{
  "result": {
    "jobs": [
      {
        "job": {
          "resulttext": "This is a test. This is the second sentence.",
          "resultsegments": [
            {
              "segment": {
                "source": "This is a test",
                "target": "Esto es una prueba",
                "confidencescore": "100.00",
                "status": ""
              }
            },
            {
              "segment": {
                "source": "This is the second sentence.",
                "target": "Esta es la segunda oración.",
                "confidencescore": "92.31",
                "status": ""
              }
            }
          ],
          "description": "",
          "status": "1",
          "jobid": "312453"
        }
      }
    ]
  },
  "JobDetail": "TODO - PUT JOB DETAIL OUTPUT HERE",
  "requestid": "20170314-092048.406-1.1.000001",
  "duration": "2000",
  "startdate": "20170314 13:48:05.591",
  "errortext": "",
  "errorno": "",
  "enddate": "20170314 13:49:03.591"
}
```


See Also

- LSTools.Job.Cancel
- LSTools.Job.Download
- LSTools.Job.Status
- LSTools.Job.Submit.JSJob
- LSTools.Job.Submit.Translate
- REST Job.Cancel
- REST Job.Download
- REST Job.Status
- REST Job.Submit.JSJob