



## **AT-RTS Ver 5.3.0**

AT Remote Translation Service

API Interfaz REST

Responsable de PFD  
Departamento IT

06 - 07 - 2020

## Índice

<b>1.OBJETIVO</b>	<b>3</b>
<b>2.¿QUÉ ES AT-RTS?</b>	<b>3</b>
<b>3.ALTA DEL SERVICIO</b>	<b>4</b>
<b>4.INTEROPERABILIDAD REST</b>	<b>5</b>
4.1 Comunicaciones HTTP	5
4.1.1 API endpoint	5
4.1.2 Uso HTTP	5
4.1.3 Gestión de errores	5
4.1.4 Formato de fechas	6
4.2 Autenticación	6
4.2.1 BÁSICA	6
4.2.2 API Key	7
4.3 Métodos REST	8
4.3.1 Control de vida:	8
4.3.2 Traducción síncrona	8
4.3.3 Traducción asíncrona	9
4.3.4 Solicitar estado de traducción por token	11
4.3.5 Seguimiento de estado y consumo	12
<b>ANEXO 1. CÓDIGOS DE RETORNO STATUS HTTP</b>	<b>14</b>

## 1. Objetivo

El presente documento tiene como objetivo detallar la documentación técnica de la API asociada al servicio de traducción remota que AT-LanguageSolutions habilita a sus clientes, tras la contratación del servicio web en modo rest.

## 2. ¿Qué es AT-RTS?

AT- RTS (Remote Translation Service) es la API web que permite la integración de nuestros servicios de traducción con los sistemas de nuestros clientes, permitiendo gestionar desde estos el control de todo el proceso de traducción de contenidos.

La API web, basada en servicio web SOAP, dispone interoperabilidad REST desde la versión 5.3.0.

Las funcionalidades operacionales son:

- **Control de vida:** Control servicio activo.
- **Traducción síncrona:** Peticiones síncronas de ordenes de traducción automática (no humana).
- **Traducción asíncrona:** Peticiones asíncronas de ordenes de traducción profesional especializada.

Cada petición retorna un **token**, asociado a la petición de traducción, cuya finalidad es identificar de forma única dicha petición de trabajo para poder ser solicitada su traducción correspondiente en un futuro una vez finalizada la revisión.

Cabe recalcar que las peticiones de traducción asíncronas disponen de la posibilidad de indicar una URL de retorno **callback** que implican activar un servicio de notificación de traducción finalizada para que nuestros clientes puedan automatizar el proceso de descarga de peticiones finalizadas.

En resumen, en este caso AT entrega las peticiones directamente a los sistemas de nuestros clientes para que estos los ubiquen en sus subsistemas.

Conceptualmente, en AT-LanguageSolutions garantizamos el **retorno de las traducciones** de peticiones de órdenes asíncronas a las infraestructuras de nuestros clientes, teniendo en cuenta que, si el sistema de comunicación falla **repetidamente durante un cierto número de ocasiones** por cualquier motivo, se enviaría un correo electrónico de aviso a la cuenta de correo que el cliente haya especificado durante la solicitud de la petición.

- **Petición de retorno de traducción:** Petición para obtener una orden de traducción profesional a partir de un **token** generado por la petición de traducción asíncrona. Si la orden no está finalizada, retorna la información del estado del documento.
- **Listado de peticiones:** Petición de estado y consumo de órdenes de traducción a partir de fecha de corte, detallando la siguiente información:
  - projectid: Identificador de proyecto AT contratado
  - token: *Token* de petición
  - sourcelang: Idioma origen
  - targetlang: Idioma destino

- filename: Nombre del archivo asociado a petición
- wordcount: Número de palabras
- status: Estado de la traducción
  - 10: En revisión
  - 20: Listo y pendiente de Entregar
  - 30: Entregado
  - 40: Eliminado
  - 100: Error de procesamiento
- urlcallback: (opcional) URL de retorno de traducción finalizada
- errnotifiersendto (opcional): email de notificación de error en el retorno de la traducción
- dateinsert: Fecha de alta
- type: Tipo de traducción, SINCRONO O ASINCRONO.

### 3. Alta del servicio

Una vez contratado el servicio, tras consensuar el modo de autenticación, el equipo IT de AT-LanguageSolutions provisionará y activará el servicio facilitando:

- URL de subdominio de acceso HTTPS.
- Credenciales de autenticación en base a tipología contratada.
  - API KEY: Licencia asociada a identificador de proyecto de traducción o bien,
  - Básica:
    - Usuario
    - Clave de acceso
    - Lista identificadores de proyecto disponibles, para usuario y claves indicadas.

A partir de este momento, cualquier petición ***deberá incorporar autenticación (Ver Comunicaciones HTTP)***, aun tratándose de un servicio web sobre protocolo HTTPS (TLS - Transport Layer Security) con la finalidad que las peticiones sean totalmente seguras.

## 4. Interoperabilidad Rest

### 4.1 Comunicaciones HTTP

#### 4.1.1 API endpoint

Solo se puede acceder a la API a través de HTTPS, la URL base se provisiona durante el proceso de alta de servicio.

#### 4.1.2 Uso HTTP

AT-RTS API v530 intenta utilizar el verbo HTTP apropiado para acceder a cada punto final de acuerdo con la especificación **REST** cuando sea posible:

Nombre	Descripción
GET	Control de vida. Seguimiento de estado y consumo. Solicitud de una orden de traducción por token.
POST	Control de vida. Seguimiento de estado y consumo. Solicitud de ordenes de traducción no humana. Solicitud de ordenes de traducción profesional. Solicitud de una orden de traducción por token.

Todas las peticiones HTTP a API permiten operaciones vía **POST**, aunque algunas de ellas se pueden peticionar vía **GET**.

En las peticiones **POST** los parámetros se deben enviar codificados incluyendo la siguiente cabecera.

***content-type: application/x-www-form-urlencoded***

Las respuestas de cualquier petición http al servicio tendrá el siguiente formato:

***content-type application/json charset=utf-8***

#### 4.1.3 Gestión de errores

En caso de no disponer cabecera ***application/x-www-form-urlencoded*** para ***peticiones POST*** o enviar parámetros incorrectos, la petición HTTP siempre retornara el estado:

***400 Bad Request***

En caso de no disponer de una autenticación válida, el estado de cualquier respuesta será:

***401 Unauthorized***

En caso de llamar a un endpoint no existente o bien realizar un GET sobre un ***endpoint*** que sólo admite una petición POST, el estado de cualquier respuesta será:

***405 Method Not Allowed***

En caso de que el servicio no esté disponible o bien haya ocurrido algún error grave en los procesos internos, el estado de cualquier respuesta será:

### **500 Internal server error**

Un ejemplo de retorno petición errónea podría ser la siguiente

```

Response
  Header
    Status: 401 Unauthorized
    Content-type: application/json

  Body
  {
    "error": {
      "errorCode": 401,
      "errorDescription": "API Key no válida."
    }
  }
    
```

La clave **errorCode** indica el código de error, mientras que la clave **errorDescription** muestra la descripción del error.

Para más información acerca de los códigos de error consulte [Anexo 1. Códigos de retorno HTTP](#).

Si el proceso se ha ejecutado con éxito el sistema devuelve el siguiente estado HTTP en la cabecera:

### **200 OK**

y en el cuerpo un esquema JSON asociado a la petición cuya estructura puede consultar en la definición de los endpoints correspondientes.

#### **4.1.4 Formato de fechas**

Las horas y fechas se devuelven y se esperan en formato de fecha [ISO 8601 \(Coordinated Universal Time\)](#):

**YYYY-MM-DDTHH:MM:SSZ**

## **4.2 Autenticación**

Las peticiones hacia el servicio web son transacciones HTTPS, habilitándose tal y como se comenta en apartado 3:

- Credenciales user|pass y lista de identificadores de proyecto asociados a par de lenguas para autenticación básica, o bien
- Clave de licencia asociada a un proyecto de trabajo por pares de lenguas (API-Key).

### **4.2.1 BÁSICA**

Consiste en enviar las credenciales de autenticación al servidor utilizando la cabecera de Autorización, que se construye de la siguiente forma:

- Nombre de usuario y contraseña, combinado en una cadena "usuario:contraseña".

- La cadena resultante se codifica utilizando la variante [RFC2045-MIME de Base64](#), sin tener la limitación de 76 caracteres por línea.
- Antes de la cadena codificada, se debe añadir el método de autorización y un espacio, es decir, “Basic “.

Ejemplo utilizando el nombre de usuario “username” y la contraseña “password”:

**Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=**

Si se usa este tipo de autenticación en las peticiones al servicio web, **se debe incluir el identificador del proyecto (projectid)** en el cuerpo de las peticiones siempre.

Ejemplo de petición:

```
POST URL_base/translateSynchronous HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Authorization: Basic your_credentials_base64encoded
projectid=your_projectid&sourcelang=spa&targetlang=cat&filename=your_filename&base64=yourcontentfile_base64encoded
```

#### 4.2.2 API Key

Las peticiones en las que se incluye una API Key autentican al usuario y además identifican el proyecto de trabajo. Por lo tanto, en las peticiones que se especifique mediante uso de API Key **no se debe añadir el identificador del proyecto**.

La API Key se debe incluir en todas las peticiones http a excepción del método ping:

- Cabecera de la petición
- Parámetro en url para peticiones GET.
- Parámetro BODY para peticiones POST.

Es decir, se debe aplicar alguna de las metodologías anteriores.

Ejemplos:

- En la cabecera de la petición:

```
GET URL_base/getlist HTTP/1.1
X-ATRTS-API-Key: your_api_key
?sourcelang=spa&targetlang=cat&filename=your_filename&base64=your_contentfile_base64encoded
```

- GET: Parámetro URL:

```
GET URL_base/getlist HTTP/1.1
?sourcelang=spa&targetlang=cat&datecutoff=your_datecutoff_ZULU_format_YYYY-MM-DDTHH:MI:SSZ&X-ATRTS-API-Key=your_api_key
```

- POST: Parámetro Body:

```
POST URL_base/getlist HTTP/1.1
Content-Type: application/x-www-form-urlencoded
sourcelang=spa&targetlang=cat&datecutoff=your_datecutoff_ZULU_format_YYYY-MM-DDTHH:MI:SSZ&X-ATRTS-API-Key=your_api_key
```

## 4.3 Métodos REST

### 4.3.1 Control de vida:

#### Descripción:

Chequeo del buen funcionamiento del servicio contratado.

Este método **NO REQUIERE AUTENTICACIÓN**.

#### Petición:

GET|POST /ping

#### Ejemplo de solicitud HTTP de control de vida:

```
GET URL_base/ping http 1.1
Response
Header
  Status:200 OK
Body
{
  "error": {
    "errorCode": 0,
    "errorDescription": null
  },
  "version": "Webservice SOAP|REST AT-RTS 5.3.0"
}
```

### 4.3.2 Traducción síncrona

#### Descripción:

Petición de orden de traducción síncrona (traducción automática no humana).

Tras la petición, el servicio retorna el archivo traducido al idioma destino especificado.

#### Petición:

POST /translateSynchronous

Parámetros de entrada:

Nombre	Tipo	Obligatorio	Descripción
projectid	Integer	Sí, si aut. básica. No, si. API Key.	Identificador de proyecto de traducción facilitado por AT.
sourcelang	String	Sí ( <a href="#">ISO-639-3</a> )	Código de idioma origen.
targetlang	String	Sí ( <a href="#">ISO-639-3</a> )	Código de idioma destino.
filename	String	Sí	Nombre del archivo.
base64	String	Sí	Contenido del archivo a traducir, codificado en base64.

encoding	String	No	Codificación del archivo. Por defecto se aplicará UTF-8.
options	String	No	JSON opciones auxiliares.

Ejemplo de petición usando autenticación mediante autenticación básica:

```

POST URL_base/translateSynchronous HTTP/1.1
Headers
Content-Type: application/x-www-form-urlencoded
Authorization: Basic your_credentials_base64encoded
projectid=your_projectid&sourcelang=spa&targetlang=cat&filename=your_filename&base64=your_contentfile_base64encoded
    
```

### Respuesta:

El servicio retornará una estructura de datos *JSON* como la siguiente en caso de petición ejecutada correctamente:

```

{
  "error": {
    "errorCode": 0,
    "errorDescription": null
  },
  "status": job_status,
  "filename": "your_filename",
  "base64": "your_contentfile_base64encoded_translated",
  "wordcount": your_file_count_words
}
    
```

En la respuesta la clave **status** retornará valor 30 (Entregado) o bien 100 en caso de error, la clave **filename** el nombre de archivo tramitado, la clave **base64** el contenido del fichero traducido al idioma destino solicitado codificado en Base64 y el campo **wordcount** el número de palabras del fichero traducidas, es decir el consumo.

### 4.3.3 Traducción asíncrona

#### Descripción:

Se genera una petición de traducción asíncrona, es decir, una revisión de una traducción llevada a cabo por un profesional de la división de lingüistas de AT-LanguageSolutions.

#### Petición:

**POST** /translateAsynchronous

Parámetros de entrada:

Nombre	Tipo	Obligatorio	Descripción
projectid	Integer	Sí, si aut. básica. No, si aut. mediante API Key.	Identificador de proyecto.
sourcelang	String	Sí ( <a href="#">ISO-639-3</a> ).	Código de idioma origen.

targetlang	String	Sí ( <a href="#">ISO-639-3</a> ).	Código de idioma destino.
filename	String	Sí	Nombre del archivo.
base64	String	Sí	Contenido del archivo a traducir, codificado en base64.
encoding	String	No	Codificación del archivo. Por defecto se aplicará UTF-8.
callbackurl	String	No	URL de <i>callback</i> para automatizar el retorno de la traducción. A esta URL se le añadirá el TOKEN de la petición al final.
errnotifiersendto	String	No	Correo electrónico para mandar una notificación si no se puede entregar la traducción mediante la URL de <i>callback</i> .
options	String	No	JSON opciones auxiliares.

Ejemplo de petición usando autenticación básica:

```

POST URL_base/translateAsynchronous HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Authorization: Basic your_credentials_base64encoded
projectid=your_projectid&sourcelang=spa&targetlang=cat&filename=your_filename&base64=your_contentfile_base64encoded&callbackurl=https%3A%2F%2Fwww.example.com%2Freturn%2Fcode%3D&errnotifiersendto=youremail_tonotify_error
    
```

### Respuesta:

El servicio retornará una estructura de datos *JSON* como la siguiente:

```

{
  "error": {
    "errorCode": 0,
    "errorDescription": null
  },
  "filename": "your_filename",
  "token": "1qxi56bgMWzbBi+6a1QS/p19bdjE2FkPwqJLrjIWdZyYq66lbaGayMV4deqbj/cW"
}
    
```

La dirección de callback resultante en este ejemplo será <https://www.example.com/return/code=1qxi56bgMWzbBi+6a1QS/p19bdjE2FkPwqJLrjIWdZyYq66lbaGayMV4deqbj/cW>

La clave **filename** indica el nombre del archivo, y la clave **token** una cadena de caracteres alfanuméricos generado mediante encriptación desde el sistema de AT-LanguageSolutions, que actúa como identificador único del archivo asociado a la petición de traducción asíncrona.

### 4.3.4 Solicitar estado de traducción por token

#### Descripción

Método concebido exclusivamente para la solicitud del resultado de las peticiones de traducciones asíncronas a través de un token único (generado en la petición de traducción asíncrona). Para que el funcionamiento de este método sea correcto en caso de realizar una petición de tipo **GET**, el token debe ir codificado en formato URL.

#### Petición:

**GET|POST /getFileByToken**

Parámetros de entrada:

Nombre	Tipo	Obligatorio	Descripción
projectid	Integer	Sí, si aut. básica. No, si aut. mediante API Key.	Identificador de proyecto.
token	String	Sí	Token resultado de la petición de traducción asíncrona.
options	String	No	JSON opciones auxiliares.

Ejemplo de petición usando autenticación mediante API Key:

```
POST URL_base/getFileByToken HTTP/1.1
Content-Type: application/x-www-form-urlencoded
X-ATRTS-API-Key: your_api_key

token=1qxi56bgMWzbBi+6a1QS/p19bdjE2FkPwqJLrjIWdZyYq66lbaGayMV4
deqbj/cW
```

#### Respuesta:

El servicio retornará una estructura de datos *JSON* como la siguiente:

```
{
  "error": {
    "errorCode":0,
    "errorDescription": null
  },
  "status": job_status,
  "filename": "your_filename",
  "base64": "your_contentfile_base64encoded_translated",
  "wordcount": your_file_count_words
}
```

La clave **status** indica el estado de la traducción. Los posibles valores para traducciones asíncronas son:

- 10: En revisión
- 20: Listo y pendiente de Entregar
- 30: Entregado
- 40: Eliminado
- 100: Error de procesamiento

La clave **filename** indica el nombre del fichero, la clave **base64** indica el contenido del fichero en base64 si la solicitud se encuentra en estado 30 (entregado) y la clave **wordcount** indica el número de palabras del fichero traducidas.

### 4.3.5 Seguimiento de estado y consumo

#### Descripción:

Petición concebida para la consulta del estado de todas las peticiones de traducción solicitadas, ya sean síncronas o asíncronas. Se pueden filtrar los resultados a partir de la fecha de envío, en formato zulu, de las peticiones de ordenes de traducción.

#### Petición:

**GET /getList**

Parámetros de entrada:

Nombre	Tipo	Obligatorio	Descripción
projectid	Integer	Sí, si aut. básica. <b>No</b> , si aut. mediante API Key.	Identificador de proyecto.
sourcelang	String	Sí ( <a href="#">ISO-639-3</a> ).	Código de idioma origen
targetlang	String	Sí ( <a href="#">ISO-639-3</a> ).	Código de idioma destino.
datecutoff	String	No( <a href="#">ISO 8601</a> ).	Fecha de corte, en el formato siguiente: YYYY-MM-DDTHH:MI:SSZ (UTC u hora Zulu)
options	String	No	JSON opciones auxiliares.

Ejemplo de petición usando autenticación básica:

```
GET URL_base/getlist HTTP/1.1
Authorization: Basic your_credentials_base64encoded
?projectid=your_projectid&sourcelang=spa&targetlang=cat&datecutoff=
your_datecutoff_ZULU_format_YYYY-MM-DDTHH:MI:SSZ
```

#### Respuesta:

El servicio retornará una estructura de datos *JSON* como la siguiente:

```
{
  "error":
    {
      "errorCode":0,
      "errorDescription": null
    },
  "files":
    [
      {"projectid":projectid_file_1,"token":token_file_1,"sourcelang":sourcelang_file_1,"targetlang":targetlang_file_1,"filename":your_filename_file_1,"wordcount":num_words_file_1,"urlcallback":URL_provided_1+token_1,"errnotifiersendto":email_notified,"status":status_file_1,"dateinsert":dateinsert_file_1,"type":type_file_1},
      {"projectid":projectid_file_2,"token":token_file_2,"sourcelang":sourcelang_file_2,"targetlang":targetlang_file_2,"filename":your_filename_file_2,"wordcount":num_words_file_2,"urlcallback":URL_provided_2+token_2,"errnotifiersendto":email_notified,"status":status_file_2,"dateinsert":dateinsert_file_2,"type":type_file_2}
    ]
}
```

Donde la clave *files* contiene un array con toda la información referente a los archivos asociados a peticiones de traducción de un proyecto y par de lenguas solicitadas posteriormente a la fecha indicada por parámetro *datecutoff*.

El contenido del array de la clave *files* detalla la siguiente información:

- Identificador del proyecto (*projectid*)
- Token, sólo para las peticiones asíncronas (*token*).
- Idioma de origen (*sourcelang*).
- Idioma de destino (*targetlang*).
- Nombre del fichero (*filename*).
- Número de palabras (*wordcount*).
- Estado:
  - 10: En revisión
  - 20: Listo y pendiente de Entregar
  - 30: Entregado
  - 40: Eliminado
  - 100: Error de procesamiento
- URL de retorno de la traducción (*urlcallback*), opcional.
- Email de notificación de error en el retorno de la traducción (*errnotifiersendto*), opcional.
- Fecha de petición (*dateinsert*) en formato UTC o "Zulú".
- Tipo de petición: síncrona (10) o asíncrona (20) (*type*)

## Anexo 1. Códigos de retorno STATUS HTTP

Todas las peticiones correctas devuelven un código de estado HTTP 200 OK.

Para las peticiones incorrectas, se pueden devolver los siguientes códigos de estado HTTP:

- 400 Bad Request – Faltan parámetros obligatorios en la petición, algún parámetro es incorrecto.
- 401 Unauthorized – Las credenciales de autenticación no son correctas.
- 405 Method Not Allowed – La petición no utiliza el método adecuado (GET o POST) o bien el método es incorrecto.
- 500 Internal Server Error – Error interno de la aplicación.