



Publications Office

TEDAPI Service

Technical Specifications Document

Subject	Technical Specifications Document
Version	1.04
Release Date	25/01/2017
Filename	TEDAPI-TSP-Technical_Specifications_Document-v1.04
Document Reference	TEDAPI-TSP

Table of Contents

1	Introduction.....	4
1.1	Purpose of the Document	4
1.2	Intended Audience	4
1.3	Structure of the Document.....	4
2	REST API description.....	5
2.1	Accessing the API.....	5
2.2	Security	5
2.2.1	Authentication	5
2.3	Operations	5
2.3.1	Get Notice Information	5
2.3.2	Import Notice.....	8
2.4	Objects.....	13
2.4.1	notice_information	13
2.4.2	extracted_data.....	13

List of Tables

Table 1: Reference Documents	Error! Bookmark not defined.
Table 2: Abbreviations and Acronyms	3
Table 3: Definitions	3

Abbreviations and Acronyms

ABBREVIATIONS AND ACRONYMS	
Abbreviation	Meaning
API	Application programming interface
ESPD	European Single Procurement Document
HTTPS	HyperText Transfer Protocol Secure
JSON	JavaScript Object Notation
OP	Office des Publications Officielles des Communautés Européennes
REST	Representational State Transfer
URL	Uniform Resource Locator

Table 1: Abbreviations and Acronyms

DEFINITIONS	
Term	Meaning
CURL	A command line tool for getting or sending files using URL syntax.

Table 2: Definitions

1 INTRODUCTION

1.1 PURPOSE OF THE DOCUMENT

The purpose of this document is to provide the required information to operate the TEDAPI system.

1.2 INTENDED AUDIENCE

The present document is intended to be read by the following people:

- OP IT Project Manager;
- OP Infrastructure Team.

1.3 STRUCTURE OF THE DOCUMENT

The document is organized as follows:

- Chapter 1 - Introduction: summarises the purpose and scope of this document;
- Chapter 2 - REST API description: describes the REST API of the TEDAPI application.

2 REST API DESCRIPTION

The TEDAPI application provides a REST API based on JSON to other applications in order to expose its services.

The TEDAPI application will exposes the following REST operations:

- Get Notice Information;
- Import Notice

2.1 ACCESSING THE API

The REST API is accessible at the following URL: <https://esentool.ted.europa.eu/api/espd/latest/>.

2.2 SECURITY

An authentication is required to access TEDAPI REST API due to data access restriction.

The credentials to access to the TEDAPI REST API are defined on a per user bases.

The TEDAPI API user/password can be changed in the TEDAPI application.

2.2.1 AUTHENTICATION

To access to the TEDAPI REST API clients must be authenticated by using standard HTTP basic authentication over HTTPS.

2.2.1.1 Simple curl example

```
$ curl 'https://esentool.ted.europa.eu/api/espd/latest/notice/15-123456-001' -i -u 'api:password' -H 'Content-Type: application/json'
```

2.2.1.2 Supplying Basic Auth headers

1. Build a string of the form username:password
2. The created string should be Base64 encoded
3. Supply an "Authorization" header with content "Basic " followed by the encoded string. For example, the string "api:password" encodes to "YXBpOnBhc3N3b3Jk" in base64, so you would make the request as follows:

```
$ curl 'https://esentool.ted.europa.eu/api/espd/latest/notice/15-123456-001' -i -H "Authorization: Basic YXBpOnBhc3N3b3Jk" -H "Content-Type: application/json"
```

2.3 OPERATIONS

2.3.1 GET NOTICE INFORMATION

2.3.1.1 URI

GET /notice/{reception_id}

2.3.1.1.1 Request

Parameter	Parameter Type	Data Type	Description
-----------	----------------	-----------	-------------

reception_id	Path	string	The reception ID of the notice
--------------	------	--------	--------------------------------

2.3.1.2 Response

2.3.1.2.1 200

In case of successful response, a **notice_information** object is returned.

Example:

```
{
  "info": {
    "EN": {
      "officialName": "Vestibulum in odio et neque.",
      "title": "Suspendisse venenatis pulvinar diam eu.",
      "shortDescription": "<P>Morbi pretium tempor sapien a aliquet.</P><P>Nulla rutrum nibh dignissim mollis pulvinar.</P><P>Suspendisse sed justo in neque malesuada sollicitudin sed eleifend purus.</P><P>Fusce vitae urna arcu.</P><P>Nam ultrices hendrerit ligula.</P><P>Vivamus id fincidunt lacus.</P><P>Nulla metus libero, congue eget odio sit amet, iaculis suscipit eros.</P><P>Vivamus quis nisi vel dui rhoncus auctor vel commodo neque.</P><P>Mauris porttitor ut dolor id semper.</P><P>Donec ultricies placerat erat luctus egestas.</P><P>Ut eu massa et libero tempor efficitur eu ut sapien.</P><P>Interdum et malesuada fames ac ante ipsum primis in faucibus.</P><P>Ut cursus, eros sit amet scelerisque pellentesque, elit nulla commodo elit, vel sodales quam eros tempor turpis.</P><P>Aliquam neque neque, consequat et aliquet a, varius quis dolor.</P>",
      "referenceNumber": "123456",
      "tedUrl": "http://ted.europa.eu/udl?uri=TED:NOTICE:123456-2015:TEXT:EN:HTML"
    }
  },
  "noDocOjs": "2015/S 125-123456"
}
```

2.3.1.2.2 400

In case the request was correctly understood but could not be processed correctly.

```
{
  "timestamp": "2015-12-31T15:11:10Z",
  "status": 400,
  "error": "Bad Request",
  "exception": "Error when processing the notice",
  "message": "Request could not be processed correctly",
  "path": "/latest/notice/15-003823-999"
}
```

2.3.1.2.3 401

In case the request was correctly understood but user does not have access to the resource.

```
{
  "timestamp": "2015-12-31T15:11:10Z",
  "status": 401,
  "error": "Unauthorized",
  "message": "Full authentication is required to access this resource",
}
```

```
"path": "/latest/notice/15-003823-999"  
}
```

2.3.1.2.4 403

In case the request was correctly understood but access has been denied

```
{  
  "timestamp": "2015-12-31T15:11:10Z",  
  "status": 403,  
  "error": "Forbidden",  
  "message": "Access is denied",  
  "path": "/latest/notice/15-003823-999"  
}
```

2.3.1.2.5 404

In case the related notice could not be found.

Example:

```
{  
  "timestamp": "2015-12-31T14:42:34Z",  
  "status": 404,  
  "error": "Not Found",  
  "exception": "The notice could not be found",  
  "message": "Notice not found",  
  "path": "/latest/notice/15-003823-999"  
}
```

2.3.1.2.6 415

In case the provided content type is not correct.

Example:

```
{  
  "timestamp": "2015-12-31T14:42:34Z",  
  "status": 415,  
  "error": "Unsupported Media Type",  
  "exception": "Content type 'xx' not supported",  
  "message": "Content type 'xx' not supported",  
  "path": "/latest/notice/15-003823-999"  
}
```

2.3.1.2.7 422

In case an error occurred when retrieving the notice information.

```
{  
  "timestamp": "2015-12-31T15:17:29Z",  
  "status": 422,  
  "error": "Unprocessable Entity",
```

```

"exception": "Could not extract metadata from notice with reception id 15-003823-001",
"message": "Request could not be processed correctly",
"path": "/latest/notice/15-003823-001"
}

```

2.3.1.2.8 500

In case an unexpected error occurred. The field `error_id` can be used for further communication.

```

{
  "timestamp": "2015-12-31T15:23:18Z",
  "status": 500,
  "error": "Internal Server Error",
  "exception": "Something bad happened",
  "message": "Something bad happened",
  "path": "/latest/notice/15-003823-001",
  "error_id": "37accc09-2a60-4665-9e72-5e260c7710f1"
}

```

2.3.1.2.9 503

When an external service needed to perform the request is unavailable. The field `error_id` can be used for further communication.

```

{
  "timestamp": "2015-12-31T15:32:34Z",
  "status": 503,
  "error": "Service Unavailable",
  "exception": "Could not contact tmon-viewer",
  "message": "Viewer not available",
  "path": "/latest/notice/15-003823-001",
  "error_id": "5992be52-aa55-4afd-9ef8-af5a0298e577"
}

```

2.3.2 IMPORT NOTICE

2.3.2.1 URI

POST /enotices/import

2.3.2.1.1 Request

Parameter	Parameter Type	Data Type	Description
notice	formData	string	XML file64 base encoded

2.3.2.2 Response

2.3.2.2.1 200

In case of successful response.

Example:

```

{

```



```
"timestamp": "2015-12-31T15:11:10Z",  
"status": 200,  
"message": "The notice has been imported",  
"path": "/latest/enotices/import "  
}
```

In case of successful response but the notice contains old NUTS code

Example:

```
{  
"timestamp": "2015-12-31T15:11:10Z",  
"status": 200,  
"message": "The notice has been imported but old NUTS have been updated to NUTS 2016",  
"path": "/latest/enotices/import "  
}
```

2.3.2.2.2 400

In case the request was correctly understood but could not be processed correctly.

Example:

```
{  
"timestamp": "2015-12-31T15:11:10Z",  
"status": 400,  
"error": "Bad Request",  
"exception": "Error when retrieving notice",  
"message": "Request could not be processed correctly",  
"path": "/latest/enotices/import "  
}
```

In case the request was correctly understood but there is a parsing error with the imported notice.

Example:

```
{  
"timestamp": "2016-07-05T16:11:07.768Z",  
"status": 400,  
"error": "Bad Request",  
"exception": "Error when processing notice",  
"message": "The notice is an invalid XML file, it cannot be parsed",  
"error_id": "a4aa9352-7258-4842-aef2-8557fad5d640",  
"path": "/ latest /enotices/import"  
}
```

In case the request was correctly understood but the imported notice contains an incorrect code NUTS/CPV/country/language/...

Example:

```
{
  "timestamp": "2016-07-05T16:11:07.768Z",
  "status": 400,
  "error": "Bad Request",
  "exception": "Error when processing notice",
  "message": "The notice contains at least one invalid code (NUTS, CPV, Language,...)",
  "path": "/ latest /enotices/import"
}
```

In case the request was correctly understood but there is at least one invalid character regarding UTF-8 range.

Example:

```
{
  "timestamp": "2016-07-05T16:15:21Z",
  "status": 400,
  "error": "Bad Request",
  "exception": "Error when processing notice",
  "message": "The notice contains at least one invalid character regarding UTF-8 range",
  "path": "/ latest /enotices/import"
}
```

In case the request was correctly understood but the type of form for this notice is not supported.

Example:

```
{
  "timestamp": "2016-07-05T16:24:44Z",
  "status": 400,
  "error": "Bad Request",
  "exception": "Error when processing notice",
  "message": "The type of form for this notice is not supported",
  "path": "/ latest /enotices/import"
}
```

In case the request was correctly understood but the user is unknown in eNotices.

Example:

```
{
  "timestamp": "2016-07-05T16:24:44Z",
  "status": 400,
  "error": "Bad Request",
  "exception": "Error when processing notice",
  "message": "User not found",
  "path": "/ latest /enotices/import"
}
```

```
}
```

2.3.2.2.3 401

In case the request was correctly understood but user does not have access to the resource.

Example:

```
{  
  "timestamp": "2015-12-31T15:11:10Z",  
  "status": 401,  
  "error": "Unauthorized",  
  "message": "Full authentication is required to access this resource",  
  "path": "/latest/enotices/import "  
}
```

2.3.2.2.4 403

In case the request was correctly understood but access has been denied.

Example:

```
{  
  "timestamp": "2015-12-31T15:11:10Z",  
  "status": 403,  
  "error": "Forbidden",  
  "message": "Access is denied",  
  "path": "/latest/enotices/import "  
}
```

2.3.2.2.5 404

In case the related service could not be found.

Example:

```
{  
  "timestamp": "2015-12-31T14:42:34Z",  
  "status": 404,  
  "error": "Not Found",  
  "exception": "The service could not be found",  
  "message": "Service not found",  
  "path": "/latest/enotices/import "  
}
```

2.3.2.2.6 415

In case the provided content type is not correct.

Example:

```
{  
  "timestamp": "2015-12-31T14:42:34Z",  
  "status": 415,  
  "error": "Unsupported Media Type",  
}
```

```
"exception": "Content type 'xx' not supported",  
"message": "Content type 'xx' not supported",  
"path": "/latest/enotices/import"  
}
```

2.3.2.2.7 422

In case an error occurred when retrieving the notice information.

Example:

```
{  
  "timestamp": "2015-12-31T15:17:29Z",  
  "status": 422,  
  "error": "Unprocessable Entity",  
  "exception": "Import notice is not in valid Base64 scheme",  
  "message": "Request could not be processed correctly",  
  "path": "/latest/enotices/import"  
}
```

2.3.2.2.8 500

In case an unexpected error occurred. The field `error_id` can be used for further communication.

```
{  
  "timestamp": "2015-12-31T15:23:18Z",  
  "status": 500,  
  "error": "Internal Server Error",  
  "exception": "Something bad happened",  
  "message": "Something bad happened",  
  "path": "/latest/notice/15-003823-001",  
  "error_id": "37accc09-2a60-4665-9e72-5e260c7710f1"  
}
```

2.3.2.2.9 503

When an external service needed to perform the request is unavailable. The field `error_id` can be used for further communication.

```
{  
  "timestamp": "2015-12-31T15:32:34Z",  
  "status": 503,  
  "error": "Service Unavailable",  
  "exception": "Could not contact eNotices services",  
  "message": "Service not available",  
  "path": "/latest/notice/15-003823-001",  
  "error_id": "5992be52-aa55-4afd-9ef8-af5a0298e577"  
}
```

2.4 OBJECTS

2.4.1 NOTICE_INFORMATION

Property	Data Type	Description
info	string	A map containing the notice language ISO code as key and the associated extracted_data object as value
noDocOjs	string	The publication number of the notice ([]/[]-[]-[]-[]-[]) if available

2.4.2 EXTRACTED_DATA

Property	Data Type	Description
officialName	string	The content of section I.1 Official Name of the notice
title	string	The content of section II.1.1 Title of the notice
shortDescription	string	The content of section II.1.4 Short Description of the notice
referenceNumber	string	The content of section II.1.1 Reference number of the notice. This value is optional.
tedUrl	string	The TED website URL of the notice if already published