



Publications Office

TEDApps - ESENTOOL

Technical Specifications Document

Subject	Technical Specifications Document
Version	1.14
Release Date	01/03/2018
Filename	ESENTOOL-TSP- Technical_Specifications_Document.docx
Document Reference	ESENTOOL-TSP

Document History

DOCUMENT HISTORY			
Version	Release Date	Author	Description
0.01	19/05/2015	Nicolas Buisson	First draft version
0.02	17/06/2015	Yannick Decoux	Submitted for Review
1.00	09/10/2015	Gauthier Leruitte	Update Rest API
1.01	30/10/2015	Gauthier Leruitte	Add statuses and reason codes list Update publication_info object
1.02	20/11/2015	Yannick Decoux	Add missing status
1.03	02/12/2015	Yannick Decoux	Update date format in /notice/search operation
1.04	12/04/2016	Yannick Decoux	Add Notice lifecycle
1.05	09/06/2016	Philippe Rock	Updated according to eSentool Release 2.1.0
1.06	19/07/2016	Philippe Rock	Minor correction
1.07	17/08/2016	Gauthier Leruitte	Minor correction
1.08	19/01/2017	Gauthier Leruitte	Version delivered with release 2.2.0
1.09	19/05/2017	Gauthier Leruitte	Version aligned with release 2.4.0: <ul style="list-style-type: none"> - Add operations: <ul style="list-style-type: none"> o Stop Publication Process o Change Status o Email Recipients
1.10	12/09/2017	Grégory Petit	Updated with OP comments from ESENTOOL-615 and ESENTOOL-772
1.11	28/09/2017	Grégory Petit	Updated with OP comments from ESENTOOL-910
1.12	13/10/2017	Grégory Petit	Updated with OP comments from ESENTOOL-917
1.13	28/11/2017	Dylan Havenith / Grégory Petit	Version aligned with release 2.5.0: <ul style="list-style-type: none"> - Update section : <ul style="list-style-type: none"> o 2.3.5.2.3 : 403 - Add section: <ul style="list-style-type: none"> o Recover Submit Updated with OP comments from ESENTOOL-921
1.14	01/03/2018	Grégory Petit / Dylan	Update with OP comments from ESENTOOL-

		Havenith	<p>946</p> <p>Add the "Simulation" column in chapter 2.4.1.</p> <p>Version aligned with release 2.6.0:</p> <ul style="list-style-type: none"> - Update section: <ul style="list-style-type: none"> o 2.3.3 Render Notice o 2.3.5 Stop publication process o 2.3 Operations: update all HTTP 400 error
--	--	----------	--

REVIEWS			
Version	Release Date	Reviewer	Position
0.02	17/06/2015	Nicolas Buisson	Quality reviewer
1.00	09/10/2015	Yannick Decoux	Quality reviewer
1.01	30/10/2015	Nicolas Buisson	Quality reviewer
1.02	20/11/2015	Nicolas Buisson	Quality reviewer
1.04	12/04/2016	Nicolas Buisson	Quality reviewer
1.05	09/06/2016	Nicolas Waeckerlé	Quality reviewer
1.06	26/07/2016	Nicolas Waeckerlé	Quality reviewer
1.07	17/08/2016	Victor Andrianjafintrimo	Quality reviewer
1.09	19/05/2017	Grégory Petit	Quality reviewer
1.10	12/09/2017	Dylan Havenith	Quality reviewer
1.11	28/09/2017	Dylan Havenith	Quality reviewer
1.12	13/10/2017	Dylan Havenith	Quality reviewer
1.13	28/11/2017	Dylan Havenith / Grégory Petit	Quality reviewer
1.14	01/03/2018	Grégory Petit / Dylan Havenith	Quality reviewer

Table of Contents

1	Introduction.....	8
1.1	Purpose of the Document	8
1.2	Intended Audience	8
1.3	Structure of the Document.....	8
2	REST API description.....	9
2.1	Accessing the API.....	9
2.2	Security	9
2.2.1	Paths	9
2.2.2	Authentication	10
2.3	Operations	11
2.3.1	Submit Notice	11
2.3.2	Get Notice Information.....	13
2.3.3	Render Notice	15
2.3.4	Search Notice	18
2.3.5	Stop Publication Process.....	21
2.3.6	Change Status for Simulation.....	24
2.4	Objects.....	25
2.4.1	notice_information	25
2.4.2	publication_info	28
2.4.3	ted_links.....	28
2.4.4	validation_report	28
2.4.5	validation_report_items	28
2.4.6	simple_result	28
2.4.7	page_result	29
2.4.8	stop_publication_info.....	29
2.4.9	notice_status	29
2.4.10	simulation_notice_status.....	30
2.4.11	simulation_reason_code.....	31
2.4.12	simulate_error	32
2.5	Notice life cycle.....	33
2.6	Java REST client.....	34
2.6.1	Create an eSentool REST Client instance	34

2.6.2 Call one of the REST operation 34

List of Tables

Table 1: Reference Documents	Error! Bookmark not defined.
Table 2: Applicable Documents.....	Error! Bookmark not defined.
Table 3: Abbreviations and Acronyms	7
Table 4: Definitions	7
Table 5: API credentials examples.....	10

List of Figures

Figure 1: Notice life cycle	33
-----------------------------------	----

Abbreviations and Acronyms

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

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 eSentool 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 eSentool application.

2 REST API DESCRIPTION

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

The eSentool application will expose the following REST operations:

- Submit Notice;
- Get Notice Information;
- Render Notice;
- Search Notice;
- Stop Publication Process;
- Change Status for Simulation

2.1 ACCESSING THE API

Three environments are available:

- Qualification: This environment should be used for test purpose and in order to get qualified. The qualification REST API is accessible at the following URL: <https://esentool.ted.europa.eu/api/qualification/latest/>.
- Production: This environment should be used to send notices for publication on TED. The production REST API is accessible at the following URL: <https://esentool.ted.europa.eu/api/production/latest/>.
- Simulation: This environment should be used for simulation purpose. It allows simulating the notices publication acceptance/refusal on TED. The simulation REST API is accessible at the following URL: <https://esentool.ted.europa.eu/api/simulation/latest/>.

If a resource from a different environment than the previously stated ones is requested, it will be the web application, and not the REST API, that will handle the request, thus generating a HTML response.

2.2 SECURITY

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

2.2.1 PATHS

The application is protected by several authentications following the accessed path and the business related.

2.2.1.1 Notices

The notices part is dedicated to the eSenders and is mapped to `/api/[environment]/[version]/notices/*`.

An eSender can access only to the notice information of its own notices, and notices of its customers. However, an eSender cannot render the notices of one of its customer. A customer can only access to its own notices.

The credentials to access to this part of the eSentool REST API are the <eSender login>:<eSender web service password>. For the customers the credentials are <eSenderlogin><customer login>:<eSender web service password>.

Examples:

Company	Credentials
eSender TED123	TED123:TED123password
Customer customer1 of eSender TED123	TED123customer1:TED123password
Customer customer1 of eSender TED456	TED456customer1:TED456password

Table 3: API credentials examples

The eSender web service password could be edited in the eSentoool web application.

2.2.1.2 eSenders

The eSenders part is dedicated to external users and is mapped to `/api/[environment]/[version]/esenders/*`.

The credentials to access to this part of the eSentoool REST API are defined in the related endpoint application properties.

2.2.2 AUTHENTICATION

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

2.2.2.1 Simple curl example

```
curl -u TED123:password -X GET -H "Content-Type: application/json"  
https://esentoool.ted.europa.eu/api/qualification/latest/notice/Q20150101-0001
```

2.2.2.2 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 `"TED123:password"` encodes to `"VEVEMTlzOnBhc3N3b3Jk"` in base64, so you would make the request as follows:

```
curl -X GET -H "Authorization: Basic VEVEMTlzOnBhc3N3b3Jk" -H "Content-Type: application/json"  
https://esentoool.ted.europa.eu/api/qualification/latest/notice/Q20150101-0001
```

2.3 OPERATIONS

2.3.1 SUBMIT NOTICE

2.3.1.1 URI

POST /notice/submit

2.3.1.1.1 Request

Parameter	Parameter Type	Data Type	Description
notice	formData	string	XML file base64 encoded

2.3.1.2 Response

2.3.1.2.1 200

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

Example:

```
{
  "submission_id": "Q20150617-5001",
  "received_at": "2015-06-17T08:10:04Z",
  "status": "RECEIVED",
  "reason_code": null,
  "status_updated_at": "2015-06-17T08:10:04Z",
  "no_doc_ext": null,
  "form": null,
  "languages": [],
  "publication_info": null,
  "technical_validation_report": null,
  "validation_rules_report": null,
  "quality_control_report": null
}
```

2.3.1.2.2 400

In case the received notice is not valid Base64 scheme.

Example:

```
{
  "timestamp": "2015-06-17T08:12:45Z",
  "status": 400,
  "error": "Bad Request",
  "exception": "The input is not in valid Base64 scheme",
  "message": "Invalid argument"
}
```

In case wrong parameter name provided.

Example:

```
{
  "timestamp" : "2018-01-05T08:51:38Z",
  "status" : 400,
  "error" : "Bad Request",
  "message" : "Request parameters unknown",
  "path" : "/latest/notice/submit",
  "exception" : "Request parameter 'notices' is not recognized, allowed parameters are: notice"
}
```

In case for other generic error.

Example:

```
{
  "timestamp": "2016-05-25T18:24:55Z",
  "status": 400,
  "error": "Bad Request",
  "message": "This error might be caused by a miss use of the API. Please check parameters and API usage according to technical specification",
  "path": "/api/production//v1.0/notice/submit",
  "error_id": "651dfcdc-efef-47fc-9452-4ad16afb9d03"
}
```

2.3.1.2.3 406

In case the Accept header is provided and set with a value different than 'application/json'.

Example:

```
{
  "timestamp": "2017-01-10T10:03:30Z",
  "status": "406",
  "error": "Not Acceptable",
  "message": "Not acceptable value for 'Accept' header. Only 'application/json' format is supported"
}
```

2.3.1.2.4 422

In case the limit of sent notices has been reached on production, qualification or simulation endpoint for this day.

Example:

```
{
  "timestamp" : "2017-12-05T07:57:29Z",
  "status" : 422,
  "error" : "Unprocessable Entity",
  "message" : "Limit of sent notices per day reached",
  "path" : "/api/production/v1.0/notice/submit",
  "exception" : "The limit of 5000 sent notice(s) in production today has been reached for eSender TED12345"
```

}

2.3.1.3 Availability

This operation is available on all endpoints: qualification, simulation and production.

2.3.2 GET NOTICE INFORMATION

2.3.2.1 URI

GET /notice/{submission_id}

GET /notice/{publication_number}

Info: in order to search a notice by “No Doc Ext”: please refer to [Search Notice](#) operation.

2.3.2.1.1 Request

Parameter	Parameter Type	Data Type	Description
submission_id	Path	string	The submission ID of the notice
publication_number	Path	string	The publication number of the notice. Accepted formats are: <ul style="list-style-type: none"> • xxxxxx-yyyy • yyyyxxxxxx

2.3.2.2 Response

2.3.2.2.1 200

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

Example:

```
{
  "submission_id": "Q20150616-5075",
  "received_at": "2015-06-16T10:47:05Z",
  "status": "VALIDATION_ACCEPTED",
  "reason_code": null,
  "status_updated_at": "2015-06-16T10:47:05Z",
  "no_doc_ext": "2014-206001",
  "form": "F01",
  "languages": ["EN"],
  "publication_info": null,
  "technical_validation_report": {
    "type": "TECH",
    "items": [
      {
        "name": "T001",
        "valid": true,
        "severity": null,
        "message": "Xml is not valid against XSD",
      }
    ]
  }
}
```

```
        "details": null
      }
    ]
  },
  "validation_rules_report": {
    "type": "VALIDATION_RULES",
    "items": [
      {
        "name": "R006",
        "valid": true,
        "severity": null,
        "message": "Check that the XML file/notice contains only utf-8 characters",
        "details": null
      }
    ]
  },
  "quality_control_report" : null,
  "ref_submission_id" : null,
  "ref_no_doc_ojs" : null }

```

2.3.2.2.2 400

In case wrong parameter name provided.

Example:

```
{
  "timestamp" : "2018-01-05T09:00:48Z",
  "status" : 400,
  "error" : "Bad Request",
  "message" : "Request parameters unknown",
  "path" : "/latest/notice/20170504-000008/",
  "exception" : "Request parameter 'notice' is not recognized, no parameter expected"
}
```

In case the operation is performed with a publication number on qualification or simulation endpoint.

Example:

```
{
  "timestamp" : "2018-01-19T15:08:43Z",
  "status" : 403,
  "error" : "Forbidden",
  "message" : "Request is forbidden",
  "path" : "/latest/notice/2013123456",
  "exception" : "Cannot search notice by publication number on simulation endpoint"
}
```

In case for other generic error

Example:

```
{
  "timestamp": "2016-05-25T18:24:55Z",
  "status": 400,
  "error": "Bad Request",
  "message": "This error might be caused by a miss use of the API. Please check parameters and API usage according to technical specification",
  "path": "/api/production/v1.0/notice/submit",
  "error_id": "651dfcdc-efef-47fc-9452-4ad16afb9d03"
}
```

2.3.2.2.3 404

In case the related notice could not be found.

Example:

```
{
  "timestamp": "2015-06-17T08:16:14Z",
  "status": 404,
  "error": "Not Found",
  "message": "Notice not found"
}
```

2.3.2.2.4 406

In case the Accept header is provided and set with a value different than 'application/json'.

Example:

```
{
  "timestamp": "2017-01-10T10:03:30Z",
  "status": "406",
  "error": "Not Acceptable",
  "message": "Not acceptable value for 'Accept' header. Only 'application/json' format is supported"
}
```

2.3.2.3 Availability

This operation is available on all endpoints when submission ID is provided: qualification, simulation and production. However, the operation is only available on production endpoint when publication number is provided.

2.3.3 RENDER NOTICE

This operation allows rendering a notice in different formats: PDF, HTML, or regulation. PDF and HTML render the notice as displayed on TED website, while REGULATION render the notice as filled on eNotices.

The notice could be provided with the *notice* parameter, or by specifying the *submission_id* of an existing notice in eSentool.

Additionally the *language* of the rendering could be specified.

2.3.3.1 URI

POST /notice/render

2.3.3.1.1 Request

Parameter	Parameter Type	Data Type	Description
notice	formData	string	XML file base64 encoded
submission_id	formData	string	The submission ID of the notice to be rendered
format	formData	string	Indicated the format of the rendering: PDF, HTML, REGULATION
language	formData	string	Optionally, the language of the rendering. The language must be specified in ISO2. For examples: FR, DE, EN ...
coverPage	formData	boolean	Indicate if cover page must be generated. Default value: true.

2.3.3.2 Response

2.3.3.2.1 200

In case of successful response a **simple_result** containing the rendered notice base64 encoded.

Example:

```
{
  "result": "JVBERi0xLjQKJa.....KNzc0OTYKJSVFT0YK"
}
```

2.3.3.2.2 400

In case the received *notice* parameter is not valid Base64 scheme.

In case the *notice* parameter or the *submission_id* parameters are not specified.

In case the rendering *format* is not valid.

In case the *language* parameter is not valid.

Example:

```
{
  "timestamp": "2015-06-17T08:20:30Z",
  "status": 400,
  "error": "Bad Request",
  "exception": "The input is not in valid Base64 scheme",
  "message": "Invalid argument"
}
```

In case wrong parameter name provided.

Example:

```
{
  "timestamp" : "2018-01-05T09:01:59Z",
  "status" : 400,
  "error" : "Bad Request",
  "message" : "Request parameters unknown",
  "path" : "/latest/notice/render",
  "exception" : "Request parameter 'format' is not recognized, allowed parameters are: submission_id, format,
coverPage, language, notice"
}
```

In case for other generic error**Example:**

```
{
  "timestamp": "2016-05-25T18:24:55Z",
  "status": 400,
  "error": "Bad Request",
  "message": "This error might be caused by a miss use of the API. Please check parameters and API usage according
to technical specification",
  "path": "/api/production/v1.0/notice/submit",
  "error_id": "651dfcdc-efef-47fc-9452-4ad16afb9d03"
}
```

2.3.3.2.3 404

In case the related notice could not be found when the *submission_id* parameter is specified.

Example:

```
{
  "timestamp": "2015-06-17T08:21:31Z",
  "status": 404,
  "error": "Not Found",
  "message": "Notice not found"
}
```

In case the related notice could be found but not its content

Example:

```
{
  "timestamp": "2015-06-17T08:21:31Z",
  "status": 404,
  "error": "Not Found",
  "message": "Content not found"
}
```

2.3.3.2.4 406

In case the Accept header is provided and set with a value different than 'application/json'.

Example:

```
{
  "timestamp": "2017-01-10T10:03:30Z",
  "status": "406",
  "error": "Not Acceptable",
  "message": "Not acceptable value for 'Accept' header. Only 'application/json' format is supported"
}
```

2.3.3.2.5 410

In case the related notice could be found but is archived

Example:

```
{
  "timestamp": "2015-06-17T08:21:31Z",
  "status": 410,
  "error": "Gone",
  "message": "The requested file has been archived"
}
```

2.3.3.2.6 422

In case the *notice* parameter is not valid against the corresponding XSD.

Example:

```
{
  "timestamp": "2015-06-17T08:22:20Z",
  "status": 422,
  "error": "Unprocessable Entity",
  "exception": "Line:1;Column:835;Error:cvc-complex-type.2.4.a: Invalid content was found starting with element 'NO_DOC_EXTX'. One of '{NO_DOC_EXT}' is expected.",
  "message": "Notice not valid"
}
```

2.3.3.2.7 503

In case the rendering service is not available.

2.3.3.3 Availability

This operation is available on all endpoints: qualification, simulation and production.

2.3.4 SEARCH NOTICE

2.3.4.1 URI

GET /notice/search

2.3.4.1.1 Request

Parameter	Parameter Type	Data Type	Description
status	queryString	notice_status	The status of the notice
receivedFrom	queryString	string	The start date of the range of notice based on received_at in yyyy/MM/dd date format
receivedTo	queryString	string	The end date of the range of notice based on received_at in yyyy/MM/dd date format
noDocExt	queryString	string	The No Doc Ext of the notice
pageSize	queryString	int	Indicates the size of the paginated result list
page	queryString	int	Indicates the page number of the paginated result list
sort	queryString	string	Indicates the field and the order of the sorting of the paginated result list. For example: submission_id,ASC to sort of the submission_id field in ascending order.

2.3.4.2 Response

2.3.4.2.1 200

In case of successful response a **page_result** objects is returned.

Example:

```
{
  "content": [
    {
      "submission_id": "Q20150612-5009",
      "received_at": "2015-06-12T09:55:09Z",
      "status": "VALIDATION_ACCEPTED",
      "reason_code": null,
      "status_updated_at": "2015-06-12T09:55:09Z",
      "no_doc_ext": "2013-518427",
      "form": "F14",
      "languages": ["FR","NL"],
      "publication_info": null,
      "technical_validation_report": {
        "type": "TECH",
        "items": [{"name": "T001","valid": true, "severity": null, "message": "Xml is not valid against XSD","details": null
      }
    },
    "validation_rules_report": {
      "type": "VALIDATION_RULES",
      "items": [{"name": "R001","valid": true, "severity": null, "message": "The email subject and the attached ZIP file
must have the same name","details": null} ] },
    "quality_control_report": {
      "type": "QUALITY_CONTROL",
```

```
    "items": [{"name": "QualityControl","valid": false, "severity": null, "message": "QualityControlMessage","details":
null}}
  }
}
],
"total_elements": 66,
"last": false,
"total_pages": 7,
"size": 10,
"number": 0,
"sort": null,
"number_of_elements": 10,
"first": true
}
```

2.3.4.2.2 400

In case wrong parameter name provided.

Example:

```
{
  "timestamp" : "2018-01-05T09:03:30Z",
  "status" : 400,
  "error" : "Bad Request",
  "message" : "Request parameters unknown",
  "path" : "/latest/notice/search/",
  "exception" : "Request parameter 'received_from' is not recognized, allowed parameters are: noDoctExt,
receivedTo, receivedFrom, pageSize, page, sort, status"
}
```

In case for other generic error

Example:

```
{
  "timestamp": "2016-05-25T18:24:55Z",
  "status": 400,
  "error": "Bad Request",
  "message": "This error might be caused by a miss use of the API. Please check parameters and API usage according
to technical specification",
  "path": "/api/production/v1.0/notice/submit",
  "error_id": "651dfcdc-efef-47fc-9452-4ad16afb9d03"
}
```

2.3.4.2.3 406

In case the Accept header is provided and set with a value different than 'application/json'.

Example:

```
{
  "timestamp": "2017-01-10T10:03:30Z",
  "status": "406",
  "error": "Not Acceptable",
  "message": "Not acceptable value for 'Accept' header. Only 'application/json' format is supported"
}
```

2.3.4.3 Availability

This operation is available on all endpoints: qualification, simulation and production.

2.3.5 STOP PUBLICATION PROCESS

2.3.5.1 URI

POST /notice/{submission_id}/stop-publication

2.3.5.1.1 Request

Parameter	Parameter Type	Data Type	Description
submission_id	Path	string	The submission ID of the notice
publicationInfo	JSON body	stop_publication_info	Complementary information about the process
simulateError	QueryString	simulate_error	<i>Only available on simulation endpoint.</i> Can be set to "CONFLICT" or "SERVICE_UNAVAILABLE" in order to simulate error code.

2.3.5.2 Response

2.3.5.2.1 200

In case of successful response, no body is returned.

2.3.5.2.2 400

In case wrong parameter name provided.

Example:

```
{
  "timestamp" : "2018-01-05T09:04:54Z",
  "status" : 400,
  "error" : "Bad Request",
  "message" : "Request parameters unknown",
  "path" : "/latest/notice/20170323-000010/stop-publication",
  "exception" : "Request parameter 'param' is not recognized, allowed parameters are: simulateError"
}
```

In case for other generic error

Example:

```
{
  "timestamp": "2016-05-25T18:24:55Z",
  "status": 400,
  "error": "Bad Request",
  "message": "This error might be caused by a miss use of the API. Please check parameters and API usage according to technical specification",
  "path": "/api/production/v1.0/notice/20150101-123456/stop-publication",
  "error_id": "651dfcdc-efef-47fc-9452-4ad16afb9d03"
}
```

2.3.5.2.3 403

In case the related notice cannot be stopped to publication due to his state.

Example:

```
{
  "timestamp": "2015-06-17T08:16:14Z",
  "status": 403,
  "error": "Forbidden",
  "message": "Http error occurred when executing 'stop publication' request. TED-Monitor message: The notice with submission id 20150101-123456 cannot be stopped because it is either in a wrong status or it is already suspended ",
  "path": "/api/production/v1.0/notice/20150101-123456/stop-publication",
}
```

2.3.5.2.4 404

In case the related notice could not be found

Example:

```
{
  "timestamp": "2015-06-17T08:16:14Z",
  "status": 404,
  "error": "Not found",
  "message": "Notice not found",
  "path": "/api/production/v1.0/notice/20150101-123456/stop-publication",
  "exception": "Impossible to find notice in production with submissionId: 20150101-123456"
}
```

2.3.5.2.5 406

In case the Accept header is provided and set with a value different than 'application/json'.

Example:

```
{
  "timestamp": "2017-01-10T10:03:30Z",
  "status": "406",
  "error": "Not Acceptable",
  "message": "Not acceptable value for 'Accept' header. Only 'application/json' format is supported",
  "path": "/api/production/v1.0/notice/20150101-123456/stop-publication"
}
```

```
}
```

2.3.5.2.6 409

In case the notice is locked in TED-Monitor.

Example:

```
{  
  "timestamp": "2017-01-10T10:03:30Z",  
  "status": "409",  
  "error": "Conflict",  
  "message": "The notice is currently locked in TED-Monitor. Please retry later.",  
  "path": "/api/production/v1.0/notice/20150101-123456/stop-publication"  
}
```

2.3.5.2.7 415

In case a `stop_publication_process` object is provided and the Content-Type header is provided and set with a value different than `'application/json'`.

Example:

```
{  
  "timestamp": "2017-01-10T10:03:30Z",  
  "status": "415",  
  "error": "Unsupported Media Type",  
  "message": "Unsupported Media Type",  
  "path": "/api/production/v1.0/notice/20150101-123456/stop-publication",  
  "exception": "Content Type 'XXX' not supported"  
}
```

2.3.5.2.8 503

In case TED-Monitor is not available.

Example:

```
{  
  "timestamp": "2017-01-10T10:03:30Z",  
  "status": "503",  
  "error": "Service unavailable",  
  "message": "Impossible to access TED-Monitor",  
  "path": "/api/production/v1.0/notice/20150101-123456/stop-publication"  
}
```

2.3.5.3 Availability

This operation is only available on the production and simulation endpoint.

2.3.6 CHANGE STATUS FOR SIMULATION

2.3.6.1 URI

POST /notice/{submission_id}/status

2.3.6.1.1 Request

Parameter	Parameter Type	Data Type	Description
submission_id	Path	string	The submission ID of the notice
noticeStatus	formData	simulation_notice_status	The status of the notice
reasonCode	formData	simulation_reason_code	The reason code of the notice

2.3.6.2 Response

2.3.6.2.1 200

In case of successful response, no body is returned.

2.3.6.2.2 400

In case wrong parameter name provided.

Example:

```
{
  "timestamp" : "2018-01-05T09:05:51Z",
  "status" : 400,
  "error" : "Bad Request",
  "message" : "Request parameters unknown",
  "path" : "/latest/notice/20170413-000012/status",
  "exception" : "Request parameter 'status' is not recognized, allowed parameters are: noticeStatus, reasonCode"
}
```

In case for other generic error

Example:

```
{
  "timestamp": "2016-05-25T18:24:55Z",
  "status": 400,
  "error": "Bad Request",
  "message": "This error might be caused by a miss use of the API. Please check parameters and API usage according to technical specification",
  "path": "/api/simulation/v1.0/notice/20150101-123456/status ",
  "error_id": "651dfcdc-efef-47fc-9452-4ad16afb9d03"
}
```

2.3.6.2.3 404

In case the related notice could not be found.

Example:


```
{
  "timestamp": "2015-06-17T08:16:14Z",
  "status": 404,
  "error": "Not found",
  "message": "Notice not found",
  "path": "/api/simulation/v1.0/notice/20150101-123456/status",
  "exception": "Impossible to find notice with submissionId: 20150101-123456"
}
```

2.3.6.2.4 406

In case the Accept header is provided and set with a value different than 'application/json'.

Example:

```
{
  "timestamp": "2017-01-10T10:03:30Z",
  "status": "406",
  "error": "Not Acceptable",
  "message": "Not acceptable value for 'Accept' header. Only 'application/json' format is supported",
  "path": "/api/simulation/v1.0/notice/20150101-123456/status"
}
```

2.3.6.3 Availability

This operation is only available on the simulation endpoint.

2.4 OBJECTS

2.4.1 NOTICE_INFORMATION

Property	Data Type	Description				
submission_id	string	The submission ID generated when the notice is submitted				
received_at	date	Reception date of the notice in eSentool in ISO8601 date-time format				
status	string	Status	Description	Qualification	Production	Simulation
		RECEIVED	Acknowledgement of receipt. The notice is being checked in the reception system.	X	X	X
		RECEPTION_ERROR	In case the notice could not be validated, and the publication		X	X

			process could not be started			
		QUALIFICATION_ERROR	In case the notice could not be validated for qualification	X		
		VALIDATION_ACCEPTED	In case the notice has been validated for qualification	X		
		QUALITY_ACCEPTED	In case the notice has been checked for qualification	X		
		QUALITY_SKIPPED	In case the notice has been skipped for qualification	X		
		IN_PROGRESS	Integration of the notice succeeded and the publication process has started.		X	X
		PUBLISHED	Notice has been published on the TED website.		X	X
		NOT_PUBLISHED	Notice has not been published on the TED website.		X	X
		WAITING_FOR_INFORMATION	In case some more information are required to publish the notice.		X	X
reason_code	string	Status	Reason code	Description		
		RECEPTION_ERROR	BV	Business validation		
			SV	Security Validation		
			XMLV	Xml validation		
		NOT_PUBLISHED	CP	Cancel Publication		
			CPV	Wrong CPV		

			DU	Duplicate
			HR	Heading Authorization Refused
			ILD	Illegible Demfax
			IN	Incomplete document
			MD	Modification
			NP	Not for Publication
			NA	No Answer to Demfax
			OD	Other Department
			OT	Other Reason
			PNP	Prepared Not Published
			WFN	Wrong Form
			WFI	Wrong From Awarding authority information
			WL	Wrong Language
			NDX	NoDocExt Already Exist
status_update_d_at	date	Date of the update of the current status in ISO8601 date-time format		
no_doc_ext	string	Unique identifier of the notice in the eSender system. A string matching the pattern: YYYY-nnnnnn. Examples: 2015-123456, 2016-000001, or 2017-999999		
form	string	The type of form of the notice: F01, F02, ...		
languages	string array	The list of languages of the notice		
publication_info	publication_info	The information of publication of the notice		
technical_validation_report	validation_report	The validation report of the notice generated during the technical checking list		
validation_rules_report	validation_report	The validation report of the notice generated during the validation rules checks		
quality_control_report	validation_report	The validation report of the notice generated during the quality control checks		
ref_submission	string	The Submission Id of the referenced notice, if any.		

_id		
ref_no_docs	string	The publication number of the referenced notice, if published.

2.4.2 PUBLICATION_INFO

Property	Data Type	Description
ojs_number	string	The OJS number of the notice as published on TED website. Examples: 001, 128, 999
no_doc_ojs	string	The NoDocOjs of the notice as published on TED website. Examples: 2014/S 006-000712, 2016/S 128-999999
publication_date	date	The publication date of the notice on TED website.
ted_links	ted_links array	The list of TED website links for each language of the notice

2.4.3 TED_LINKS

Property	Data Type	Description
key	string	The language code of the TED website link
value	string	The TED website link

2.4.4 VALIDATION_REPORT

Property	Data Type	Description
type	string	The type of validation report, can be one of TECH, VALIDATION_RULES, QUALITY_CONTROL
items	validation_report_items array	A list of validation_report_items

2.4.5 VALIDATION_REPORT_ITEMS

Property	Data Type	Description
name	string	The name of the validation rule
valid	boolean	Indicates if the validation rule successfully passed
severity	String	In case the validation rule failed, indicated the severity of the rule: INFO, WARNING, ERROR, CRITICAL
message	string	A message to briefly describe the error
details	string	A complete error message

2.4.6 SIMPLE_RESULT

Property	Data Type	Description
result	string	The result of the operation, this can be a message or a

	base64 payload such as a PDF.
--	-------------------------------

2.4.7 PAGE_RESULT

Property	Data Type	Description
content	array of notice_information	Contains a page of search results
total_elements	int	The total number of elements in this search results
last	boolean	Whether this page is the last page
total_pages	int	Total number of pages
size	int	The page size
number	int	The number of the current page, starting from 0
sort	string	The actual sorting of the elements
number_of_elements	int	Number of elements in this page
first	boolean	Whether this page is the first page

2.4.8 STOP_PUBLICATION_INFO

Stop publication info is a JSON request body structured as follow:

```
{
  "attachment":{
    "filename": "stopPublicationDocument.xml",
    "data": "RGF0YSBpbBiYXNINjQ=..."
  }
}
```

Property	Data Type	Description
filename	string	Filename of the attachment
data	string	Base64 encoded file

2.4.9 NOTICE_STATUS

Property	Data Type	Description				
		Status	Description	Qualification	Production	Simulation
status	string	RECEIVED	Acknowledgement of receipt. The notice is being checked in the reception system.	X	X	X
		RECEPTION_ERROR	In case the		X	X

			notice could not be validated, and the publication process could not be started			
		QUALIFICATION_ERROR	In case the notice could not be validated for qualification	X		
		VALIDATION_ACCEPTED	In case the notice has been validated for qualification	X		
		QUALITY_ACCEPTED	In case the notice has been checked for qualification	X		
		QUALITY_SKIPPED	In case the notice has been skipped for qualification	X		
		IN_PROGRESS	Integration of the notice succeeded and the publication process has started.		X	X
		PUBLISHED	Notice has been published on the TED website.		X	X
		NOT_PUBLISHED	Notice has not been published on the TED website.		X	X
		WAITING_FOR_INFORMATION	In case some more information are required to publish the notice.		X	X

2.4.10 SIMULATION_NOTICE_STATUS

Property	Data Type	Description
----------	-----------	-------------

noticeStatus	string	Status	Description
		PUBLISHED	Notice has been published on the TED website (simulation mode).
		NOT_PUBLISHED	Notice has not been published on the TED website.
		WAITING_FOR_INFORMATION	In case some more information are required to publish the notice.

2.4.11 SIMULATION_REASON_CODE

Property	Data Type	Description	
reasonCode	string	Reason Code	Description
		NO_ANSWER	NA - No Answer to Demfax
		OTHER_DEPARTMENT	OD - Other Department
		PREPARED_NOT_PUBLISHED	PNP - Prepared Not Published
		OTHER_REASON	OT - Other Reason
		ILLEGIBLE_DEMFAX	ILD - Illegible Demfax
		WRONG_LANGUAGE	WL - Wrong Language
		WRONG_FORM_AAI	WFI - Wrong Form Awarding authority information
		WRONG_FORM_NAN	WFN - Wrong Form
		CANCEL_PUBLICATION	CP - Cancel Publication
		NOT_FOR_PUBLICATION	NP - Not for Publication
		INCOMPLETE_DOCUMENT	IN - Incomplete Document
		HEADING_REFUSAL	HR - Heading Authorization Refused
		DUPLICATE	DU - Duplicate
		MODIFICATION	MD - Modification
NODOCEXT_DUPLICATE	NDX - NoDocExt Already Exist		
WRONG_CPV	CPV - Wrong CPV		

2.4.12 SIMULATE_ERROR

Property	Data Type	Code	Description
simulateError	string	CONFLICT	Simulate conflict (two processes trying to update same entity) on the notice.
		SERVICE_UNAVAILABLE	Simulate an underlying service which is unavailable.

2.5 NOTICE LIFE CYCLE

Below is a diagram showing the life cycle of a Notice.

eSentool – Notice life cycle

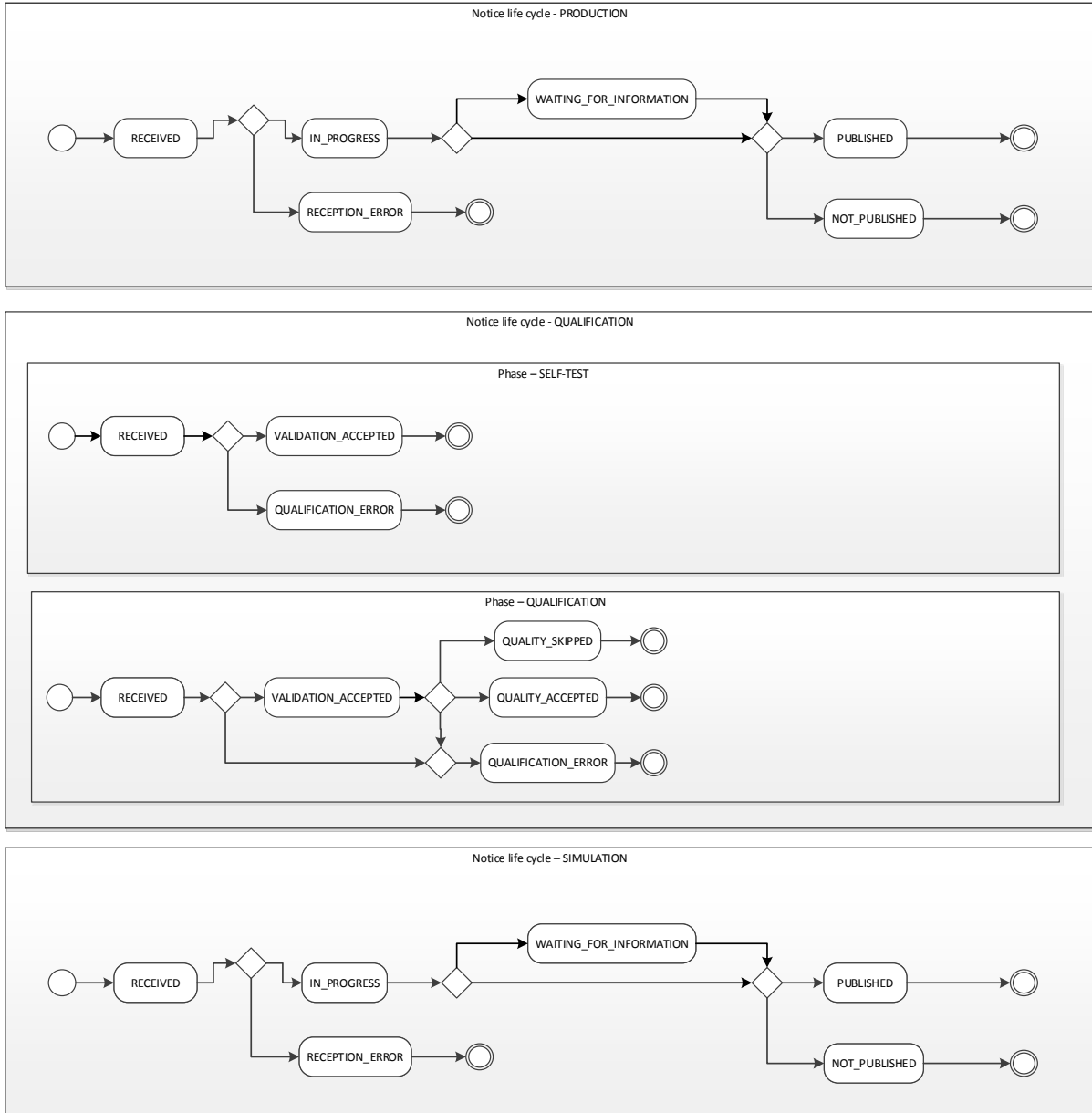


Figure 1: Notice life cycle

2.6 JAVA REST CLIENT

A JAVA client could be used in order to ease the request to the eSentool REST API. This library has external dependencies with `org.apache.httpcomponents:httpclient:4.4.1` and `com.fasterxml.jackson.core:jackson-databind:2.5.3`.

2.6.1 CREATE AN ESENTOOL REST CLIENT INSTANCE

```
new EsentoolRestClientBuilder("https://hostname/qualification/rest/v1.0/", "TED123", "password").build();
```

2.6.2 CALL ONE OF THE REST OPERATION

```
Page<NoticeInformation> result = client.searchNotices(new  
SearchNoticeParams().withPage(0).withPageSize(15).withSort("submission_id").withSortDirection(SearchNoticeP  
arams.SortDirection.ASC));
```