

**Technical document**

Version 0.0.7 – 10 June 2021

# **INTERFACE AGREEMENT API DOCUMENTS**

La Poste – Colissimo



**LA POSTE**



I.	Introduction.....	5
I.1	Document purpose .....	5
I.2	Overview.....	5
II.	General points .....	6
II.1	Coding of formats.....	6
II.2	Coding of Conditions .....	6
II.3	Warning on JSON and XML sample files .....	6
II.4	General presentation of resources .....	7
II.4.1	Encoding of special characters .....	7
II.4.2	API Keyword Languages .....	7
II.4.3	Date format .....	7
II.4.4	HTTP return codes .....	7
II.4.4.1	: Success 2XX .....	7
II.4.4.2	Customer Errors: 4XX .....	7
II.4.4.3	Server Errors: 5XX.....	8
II.4.5	Managing errors/status.....	9
II.4.5.1	Modelling.....	9
II.4.5.2	Error/status block .....	9
II.4.5.3	Error block.....	9
III.	API Documents Technical Integration Guide .....	10
III.1	Access to the API Documents.....	10
III.2	Credential block .....	10
III.3	REST Resources .....	11
III.3.1	The resource /documents .....	11
III.3.1.1	Description .....	11
III.3.1.2	Access URL .....	11
III.3.1.3	HTTP method .....	11
III.3.1.4	HTTP return code/message if successful .....	11
III.3.1.5	HTTP return code/message if Error .....	11
III.3.1.6	Input parameters .....	11
III.3.1.7	Response.....	12
III.3.1.8	Document block .....	12
III.3.1.9	Example .....	12
III.3.2	The resource /document .....	14
III.3.2.1	Description .....	14
III.3.2.2	Access URL .....	14
III.3.2.3	HTTP method .....	14
III.3.2.4	HTTP return code/message if successful .....	14
III.3.2.5	HTTP return code/message if error .....	14
III.3.2.6	Input parameters .....	14
III.3.2.7	Response.....	15
III.3.2.8	Example .....	15
III.3.3	The resource /storedocument.....	16
III.3.3.1	Description .....	16
III.3.3.2	Access URL.....	16
III.3.3.3	HTTP method .....	16
III.3.3.4	HTTP return code/message if successful .....	16

III.3.3.5	HTTP return code/message if Error .....	16
III.3.3.6	Input parameters .....	17
III.3.3.7	Response.....	17
III.3.3.8	Example .....	17
III.3.4	The resource /updatedocument .....	19
III.3.4.1	Description .....	19
III.3.4.2	Access URL .....	19
III.3.4.3	HTTP method .....	19
III.3.4.4	HTTP return code/message if successful .....	19
III.3.4.5	HTTP return code/message if Error .....	19
III.3.4.6	Input parameters .....	20
III.3.4.7	Response.....	20
III.3.4.8	Example .....	20
III.4	Appendices .....	22
III.4.1	Appendix 1: Global status codes .....	22
III.4.2	Appendix 2: Error Codes.....	22
III.4.3	Appendix 3: Language codes.....	24
III.4.4	Appendix 4: Functional nature of documents .....	25

## **I. Introduction**

### **I.1 Document purpose**

---

This document describes access and the particular uses of the API Documents.

This API allows you to attach the documents required for customs clearance of parcels bound for overseas France and parcels with the FTD/DDP (Franc de Taxes et de Droits and Delivery Duty Paid Fedex) option

It also allows you to consult the documents linked with a parcel whatever its destination. The documents are either provided by the customer (those mentioned just before) or by Colissimo (proof of delivery, certificate of delivery, compensation statement)

### **I.2 Overview**

---

The API has been designed in REST in order to allow a better integration in different programming languages (Go, Java, Python, etc.).

## II. General points

### II.1 Coding of formats

Code	Description
<b>A</b>	Alphabetical characters (no spaces or punctuation)
<b>AN</b>	Any printable character
<b>N</b>	Numerical decimal characters from 0 to 9
<b>D</b>	Date
<b>T</b>	Time
<b>DT</b>	Date and time
<b>SUB</b>	Block: list of sub-fields
<b>L</b>	Choose from a list of values
<b>B</b>	Boolean (true/false; 1/0)

The numbers after these codes in the data descriptions indicate the length of the field. A single number indicates a set length field (e.g. "N6" for 6-character numeric), two numbers separated by two full stops indicating the minimum and maximum length (e.g. "AN0..35" for alphanumeric that can be blank and up to a maximum of 35 characters).

### II.2 Coding of Conditions

Code	Description
<b>O</b>	Obligatory data. The information carried by this data must always be present.
<b>F</b>	Optional data This information can be provided if present in the issuing information system.
<b>C</b>	Conditional presence based on the content of another field in the same record.

### II.3 Warning on JSON and XML sample files

For presentation purposes, XML and JSON files are enriched with presentation elements (tabs, line breaks, etc.). These elements are here only to facilitate the reading of the document and are not necessarily included in the product files. This document only commits to the structure of these files, not to their presentation. Likewise, the order of the elements is not guaranteed and no hypothesis should be retained by the clients as to this order.

## II.4 General presentation of resources

---

### II.4.1 Encoding of special characters

---

Characters must be UTF-8 encoded.

### II.4.2 API Keyword Languages

---

The API data is divided into two parts:

- Technical data: described in English
- Functional data: described in French

### II.4.3 Date format

---

All API dates are in DateTime format. Accuracy is to the millisecond (3 digits after the second).

Example:

```
2019-04-11T12:47:25.123Z
```

### II.4.4 HTTP return codes

---

#### II.4.4.1 : Success 2XX

---

Status	Code	Description
<b>200</b>	OK	Classic code for success.

#### II.4.4.2 Customer Errors: 4XX

---

These codes represent errors due to customer actions that the API rejects. Most of these statuses are solvable by the customer (except for rights), often because they are the consequence of usage problems.

Status	Code	Description
<b>400</b>	Bad Request	Usually used for call errors, if no other status matches. This error indicator is used when checking method input data.
<b>401</b>	Unauthorized	I don't know you. Tell me who you are and I'll check your credentials.
<b>403</b>	Forbidden	You are properly authenticated, but you are not authorized to access the resource.
<b>404</b>	Not Found	The resource you are asking for does not exist.

### II.4.4.3 Server Errors: 5XX

---

These indicators represent statuses for which the customer cannot do anything. The server is then in an insolvent state, and depending on the request the service is suspended indefinitely (partially or totally).

Status	Code	Description
<b>500</b>	Server Error	Calling the resource is valid but an execution problem is encountered. There's nothing the client can do about it.
<b>501</b>	Not Implemented	The server does not refuse the request, but it cannot succeed because the method is not yet implemented. If the service is open before the API is finalized, the less important functions may be inaccessible and available in a later iteration.
<b>503</b>	Service not available	The server cannot respond due to maintenance, or other reasons of unavailability. The Retry-After header can be used with this code to specify when the service should be operational again.



## II.4.5 Managing errors/status

---

The structure of an error/status is in the following format:

```
{
  "errorCode": "001",
  "errorLabel": "CAB NOT FOUND",
  "errors": [
    {
      "code": 153,
      "message": "Parcel not found"
    }
  ]
}
```

### II.4.5.1 Modelling

---

#### II.4.5.2 Error/status block

---

Data	Format	Mandatory	Comment
errorCode	AN	O	Global return status indicator. See details of the statutes in section III.4.1
errorLabel	AN	O	Message linked with the global return status. See details of the statutes in section III.4.1
errors	L<SUB>	C	List detailing errors if the global status indicator is other than 000. See details of the paragraph block II.4.5.3

#### II.4.5.3 Error block

---

Data	Format	Mandatory	Comment
code	AN	O	Error code. See details of the errors in the section <b>Erreur ! Source du renvoi introuvable.</b>
message	AN	O	Message details the error. See details of the errors in the section <b>Erreur ! Source du renvoi introuvable.</b>

## III. API Documents Technical Integration Guide

### III.1 Access to the API Documents

---

The URL of the API Documents is: <https://ws.colissimo.fr/api-document/>

### III.2 Credential block

---

The table below describes the credential block to access the API Documents.

Data	Format	Mandatory	Comment
login	AN	<b>C</b>	Login of the account that wishes to access the resource. If informed do not inform the apiKey.
password	AN	<b>C</b>	Password of the account that wishes to access the resource. If informed do not inform the apiKey.
apiKey	AN	<b>C</b>	API key of the account that wishes to access the resource. If filled in do not fill in the login and password.

## III.3 REST Resources

---

### III.3.1 The resource /documents

---

Method for obtaining documents from a parcel.

#### III.3.1.1 Description

---

Calling this resource allows to obtain a list of documents (receiver's signature, delivery certificate, CN23, etc.) of a parcel from a CAB.

#### III.3.1.2 Access URL

---

The access URL to the resource is as follows:

```
/api-document/rest/documents
```

#### III.3.1.3 HTTP method

---

To access this resource you must use the verb:

```
POST
```

#### III.3.1.4 HTTP return code/message if successful

---

The resource returns the following HTTP code:

```
200: Successful
```

#### III.3.1.5 HTTP return code/message if Error

---

The resource returns the following HTTP code:

```
400: Bad request  
401: Unauthorized  
403: Forbidden  
404: Not Found  
503: Service not available
```

#### III.3.1.6 Input parameters

---

The input parameters expected by the method are listed below. Any element sent that is not shown in this table will be ignored by the API.

Data	Format	Mandatory	Comment
credential	SUB	O	Credential to access the API documents. See section III.2
lang	AN	N	Language code for response. By default, fr_FR. See section III.4.3
cab	AN	C	Parcel number

### III.3.1.7 Response

The structure of the response is as follows:

Data	Format	Mandatory	Comment
errorCode errorLabel errors		<b>O</b>	See details in section II.4.5
documents	SUB	<b>O</b>	Parcel document list

### III.3.1.8 Document block

This block describes the document block.

Data	Format	Mandatory	Comment
URL	AN	<b>O</b>	The call URL of the API Documents to retrieve the document
eventCode	AN	<b>N</b>	Code of event
eventDate	AN	<b>N</b>	Date of event
eventSiteCode	AN	<b>N</b>	The code of the site of the event
eventSiteName	AN	<b>N</b>	The name of the event site
eventSiteType	AN	<b>N</b>	The name of the issuing depot of the event
documentType	AN	<b>O</b>	Type of document
path	AN	<b>N</b>	
uuid	AN	<b>N</b>	Parcel technical ID
cab	AN	<b>N</b>	Parcel number

### III.3.1.9 Example

Example of call in curl:

```
curl -X POST 'https://ws.colissimo.fr/api-document/rest/documents' \
-H 'Content-Type: application/json' \
-d
'{"credential":{"login":"mylogin","password":"mypassword"},"cab":"mycab}"'
```

Return example:

```
{
  "errorCode": "000",
  "errorLabel": "OK",
  "documents": [
    {
      "URL": "/document",
      "eventCode": null,
      "eventDate": null,
      "eventSiteCode": null,
      "eventSiteName": null,
      "eventSiteType": null,
      "documentType": "CN23",
      "carrier": "COL",
      "path": "/uds/e35f56ae-e1f2-3d3d-9901-347c5d1d1b88.pdf",
```

```
"uuid": "912d976c-30e1-3c3a-b24d-dd9420d48a52",  
"cab": "mycab"  
  }  
]  
}
```

## III.3.2 The resource /document

---

Method for retrieving a document

### III.3.2.1 Description

---

Calling this resource allows the retrieval of a document from the URL returned by the resource "/documents".

### III.3.2.2 Access URL

---

The access URL to the resource is as follows:

```
/api-document/rest/document
```

### III.3.2.3 HTTP method

---

To access this resource you must use the verb:

```
POST
```

### III.3.2.4 HTTP return code/message if successful

---

The resource returns the following HTTP code:

```
200: Successful
```

### III.3.2.5 HTTP return code/message if error

---

The resource returns the following HTTP code:

```
400: Bad request  
401: Unauthorized  
403: Forbidden  
404: Not Found  
503: Service not available
```

### III.3.2.6 Input parameters

---

The input parameters expected by the method are listed below. Any element sent that is not shown in this table will be ignored by the API.

Data	Format	Mandatory	Comment
credential	SUB	0	Credential to access the API documents. See section III.2
path	AN	0	The path to access the resource, returned when calling the resource "documents" or "signatures"
cab	AN	0	The cab to access the resource, returned when calling the resource "documents" or "signatures"

Data	Format	Mandatory	Comment
uuid	AN	0	The uuid to access the resource, returned when calling the resource "documents" or "signatures"

### III.3.2.7 Response

---

The call of the URL allows the download (display) of the document.

### III.3.2.8 Example

---

Example of call in curl:

```
curl -X POST 'https://ws.colissimo.fr/api-document/rest/document' \  
-H 'Content-Type: application/json' \  
-d \  
'{"credential":{"login":"mylogin","password":"mypassword"},"path":"/uds/11684757-49c4-3c60-a35f-4b1e87dd7d6c.pdf","uuid": "912d976c-30e1-3c3a-b24d-dd9420d48a52","cab":"mycab"}'
```

This call returns the bitstream of the file

### **III.3.3 The resource /storedocument**

---

Method for storing a new document.

#### **III.3.3.1 Description**

---

Calling this resource allows you to store a document.

#### **III.3.3.2 Access URL**

---

The access URL to the resource is as follows:

```
/api-document/rest/storedocument
```

#### **III.3.3.3 HTTP method**

---

To access this resource you must use the verb:

```
POST
```

#### **III.3.3.4 HTTP return code/message if successful**

---

The resource returns the following HTTP code:

```
200: Successful
```

#### **III.3.3.5 HTTP return code/message if Error**

---

The resource returns the following HTTP code:

```
400: Bad request  
401: Unauthorized  
403: Forbidden  
404: Not Found  
503: Service not available
```



### III.3.3.6 Input parameters

---

The input parameters expected by the method are listed below. Any element sent that is not shown in this table will be ignored by the API.

The input parameters must be set in the header for the credentials and in the body for the form parameters.

Below are the parameters to be set in the header:

Data	Format	Mandatory	Comment
login	AN	<b>C</b>	Login of the account that wishes to access the resource. If informed do not inform the apiKey.
password	AN	<b>C</b>	Password of the account that wishes to access the resource. If informed do not inform the apiKey.
apiKey	AN	<b>C</b>	API key of the account that wishes to access the resource. If filled in do not fill in the login and password.

Below are the parameters to be set in the body:

Data	Format	Mandatory	Comment
accountNumber	AN	<b>O</b>	Account number (third party code)
parcelNumber	AN	<b>O</b>	Parcel number
documentType	AN	<b>O</b>	Functional nature of the document. See details in section III.4.4
file	Binary	<b>O</b>	Binary content of the document to be stored
filename	AN	<b>O</b>	File name
parcelNumberList	AN	<b>N</b>	List of parcels following the master parcel in the case of multi-parcel shipments. Parcels are separated by a comma.

### III.3.3.7 Response

---

The structure of the response is as follows:

Data	Format	Mandatory	Comment
errorCode errorLabel errors		<b>O</b>	See details in section II.4.5
documentId	AN	<b>O</b>	The document storage id

### III.3.3.8 Example

---

Example of call in curl:

```
curl -X POST 'https://ws.colissimo.fr/api-document/rest/storedocument' \
-H 'login: myLogin' \
-H 'password: myPassword' \
-H 'Content-Type: multipart/form-data' \
--form 'accountNumber=101102' \
--form 'parcelNumber=EY607748960FR' \
```

```
--form 'documentType=CN23' \  
--form 'file=@/tmp/EY607748960FR-CN23.pdf' \  
--form 'filename=EY607748960FR-CN23.pdf'
```

Return example:

```
{  
  "errorCode": "000",  
  "errorLabel": "OK",  
  "documentId": "50c82f93-015f-3c41-a841-07746eee6510.pdf"  
}
```

### III.3.4 The resource /updatedocument

---

Method for replacing a document type already stored.

#### III.3.4.1 Description

---

Calling this resource allows you to update a document. The document must exist.

#### III.3.4.2 Access URL

---

The access URL to the resource is as follows:

```
/api-document/rest/updatedocument
```

#### III.3.4.3 HTTP method

---

To access this resource you must use the verb:

```
POST
```

#### III.3.4.4 HTTP return code/message if successful

---

The resource returns the following HTTP code:

```
200: Successful
```

#### III.3.4.5 HTTP return code/message if Error

---

The resource returns the following HTTP code:

```
400: Bad request  
401: Unauthorized  
403: Forbidden  
404: Not Found  
503: Service not available
```

### III.3.4.6 Input parameters

---

The input parameters expected by the method are listed below. Any element sent that is not shown in this table will be ignored by the API.

The input parameters must be set in the header for the credentials and in the body for the form parameters.

Below are the parameters to be set in the header:

Data	Format	Mandatory	Comment
login	AN	<b>C</b>	Login of the account that wishes to access the resource. If informed do not inform the apiKey.
password	AN	<b>C</b>	Password of the account that wishes to access the resource. If informed do not inform the apiKey.
apiKey	AN	<b>C</b>	API key of the account that wishes to access the resource. If filled in do not fill in the login and password.

Below are the parameters to be set in the body:

Data	Format	Mandatory	Comment
accountNumber	AN	<b>O</b>	Account number (third party code)
parcelNumber	AN	<b>O</b>	Parcel number
documentType	AN	<b>O</b>	Functional nature of the document. See details in section III.4.4
file	Binary	<b>O</b>	Binary content of the document to be stored
filename	AN	<b>O</b>	File name
parcelNumberList	AN	<b>N</b>	List of parcels following the master parcel in the case of multi-parcel shipments. Parcels are separated by a comma.

### III.3.4.7 Response

---

The structure of the response is as follows:

Data	Format	Mandatory	Comment
errorCode errorLabel errors		<b>O</b>	See details in section II.4.5
documentId	AN	<b>O</b>	The document storage id

### III.3.4.8 Example

---

Example of call in curl:

```
curl -X POST 'https://ws.colissimo.fr/api-document/rest/updatedocument' \
-H 'login: myLogin' \
-H 'password: myPassword' \
-H 'Content-Type: multipart/form-data' \
--form 'accountNumber=101102' \
--form 'parcelNumber=EY607748960FR' \
```

```
--form 'documentType=CN23' \  
--form 'file=@/tmp/EY607748960FR-CN23.pdf' \  
--form 'filename=EY607748960FR-CN23.pdf'
```

Return example:

```
{  
  "errorCode": "000",  
  "errorLabel": "OK",  
  "documentId": "50c82f93-015f-3c41-a841-07746eee6510.pdf"  
}
```

## III.4 Appendices

### III.4.1 Appendix 1: Global status codes

---

The table below lists the return statuses for the API Documents.

Code	Message	Comment
000	OK	The request has been processed successfully.
001	CAB NOT FOUND	
002	CAB NOT FOUND	
003	DOCUMENT NOT FOUND	
004	CREDENTIAL INVALID	
005	BODY INVALID	
006	DOCUMENTUPLOADERREQUEST INVALID	
999	KO	Request failure

### III.4.2 Appendix 2: Error Codes

---

The table below lists the error codes for the API Documents.

Code	Message	Comment
10	Missing "credential" parameter	
11	Invalid login	
12	Invalid password	
13	Please enter either the login or the apiKey. But not both.	
14	Please enter either the login or the token. But not both.	
15	Please enter either the apiKey or the token. But not both.	
20	Invalid language	
30	Document upload failed	
32	Updating document failed (document not found)	
34	Updating document failed (different document type)	
35	Updating document failed (more than 2 documents)	
130	The path {0} is incorrect	
131	The file {0} cannot be found	
146	Invalid file name	
147	Invalid upload file	
148	Invalid document type	Accepted type : .pdf .jpg .png .tiff
149	Invalid account number	
150	Invalid parcel number	
152	Problem when retrieving the parcel	
153	Parcel not found	
159	Mime type document invalid	

<b>Code</b>	<b>Message</b>	<b>Comment</b>
190	Problem generating the Colissimo compensation balance sheet	
191	Problem generating the Colissimo delivery certificate	
401	Account not authenticated	
403	Unauthorized account	
413	Document too large	The document size exceeds the allowed size of 500 KB.

### III.4.3 Appendix 3: Language codes

---

The table below lists the languages translated for the API Documents. These 5 languages are used in the PDF delivery certificate.

<b>Code</b>	<b>Comment</b>
fr_FR	French (France)
en_GB	English (UK)
es_ES	Spanish (Spain)
de_DE	Deutsch (German)
it_IT	Italian (Italy)



### III.4.4 Appendix 4: Functional nature of documents

The table below lists the functional natures of the documents allowed in storage by the API Documents.

Type of document	Description	Overseas parcels	Parcels with FTD/DDP option
CERTIFICATE_OF_ORIGIN	Certificate of origin	X	X
EXPORT_LICENSE	Export license	X	X
COMMERCIAL_INVOICE	Invoice of the parcel	X	N/A
OTHER	Other	X	X

The table below lists the functional natures of the documents that can be returned by the API Documents in consultation.

Type of document	Description
CERTIFICATE_OF_ORIGIN	Certificate of origin
EXPORT_LICENSE	Export license
COMMERCIAL_INVOICE	Invoice of the parcel
OTHER	Other
C50	C50
COMPENSATION	Compensation balance sheet
DAU	Invoice for duties and taxes
DELIVERY_CERTIFICATE	Delivery certificate
SIGNATURE	Proof of delivery