CouriersPlease International Label API Documentation

CouriersPlease®

CouriersPlease

API Version: 1.0.0

1. VERSION CONTROL

Version	Date	Notes	Author
1.0.0	10/06/2016	Initial version.	Jeff Embro (CouriersPlease)
1.0.1	05/07/2016	Label printing instructions	Jeff Embro (CouriersPlease)
1.0.2	25/10/2016	Fixed urls.	Jeff Embro (CouriersPlease)

2. TABLE OF CONTENTS

1. Version Control 1
2. Table of Contents 1
3. Overview 2
4. Label Printing Instructions 2
5. Consumes 2
6. Output Formats 2
7. HTTP Method 2
8. URLs 2
   8.1. Sandbox 2
   8.2. Production 2
9. Security 3
   9.1. Authentication 3
   9.2. SSL Encryption 3
10. Request 3
   10.1. Sample Request That Returns JSON 3
      10.1.1. Request 3
      10.1.2. Request Header 4
   10.2. Sample Request That Returns XML 4
      10.2.1. Request 4
      10.2.2. Request Header 4
11. Response 4
   11.1. HTTP Status Codes 4
   11.2. Response Codes 4
   11.3. Output 5
   11.4. Sample JSON Response (Success – 200) 5
   11.5. Sample XML Response (Success – 200) 5
   11.6. Sample JSON Response (Invalid Input - 400) 6
   11.7. Sample XML Response (Invalid Input - 400) 6
   11.8. Sample JSON Response (Unauthorized - 401) 6
   11.9. Sample XML Response (Unauthorized - 401) 6

Glossary 6

3. OVERVIEW

This document defines the API in which shipping labels and a commercial invoice can be retrieved for an international consignment (DHL or SingPost).

4. LABEL PRINTING INSTRUCTIONS

The shipping labels are to be affixed to the freight before shipping. The commercial invoice also needs to be included with the consignment in order to clear customs. This should be done as follows:

DHL – Customers must print one copy of the label and two copies of the commercial invoice. The labels must be affixed to the freight. The archive label (which is part of the label) should be handed to the driver along with both copies of the commercial invoice.

SingPost – Customers must print one copy of the label and one copy of the commercial invoice. Customers should affix the label (which includes a commercial invoice as part of the label) to the freight. The commercial invoice needs to be handed to the driver.

Rule: If the consignment number contains “SAV” it is SingPost consignment. If the consignment number contains “EXP” it is a DHL consignment.

Please ensure you follow the CouriersPlease label printing specifications.

5. CONSUMES

JSON

6. OUTPUT FORMATS

JSON
XML

7. HTTP METHOD

GET

8. URLS

8.1. SANDBOX

While developing and testing use the sandbox URL:

https://api-test.couriersplease.com.au/v1/international/shipment/label?consignmentNumber=

8.2. PRODUCTION

After deploying to live use the production URL:

https://api.couriersplease.com.au/v1/international/shipment/label?consignmentNumber=

9. SECURITY

API security is controlled by CouriersPlease. CouriersPlease controls the following:

1. Access to the Sandbox environment
2. Access to the Production environment
3. Access to each individual API
4. Enable/disable a token
5. Hourly and daily limits

If you require security changes please contact CouriersPlease API support: apisupport@couriersplease.com.au.

9.1. AUTHENTICATION

API Validation is performed through HTTP Basic Authentication using the CouriersPlease Account Number and Authorization Token.

User Name: CouriersPlease Account Number

Password: Sandbox or Production Token provided from the CouriersPlease API Developer Portal

When using HTTP Basic Authentication the User Name and Password must be Base64 encoded in the header.

If your CouriersPlease Account Number is W99999 and your token is ABC123456789 your Authentication Header would be “Authorization: Basic W99999:ABC123456789” before encoding it. Once encoded, it should look as follows: “Authorization: Basic Vzk5OTk5OkFCQzEyMzQ1Njc4OQ==”

9.2. SSL ENCRYPTION

The API Endpoints and Portal are encrypted using SSL as follows:

SSL Attribute	SSL Detail
Signature Hash Algorithm	sha256
Public Key	RSA (2048 Bits)

10. REQUEST

Name	Type
consignmentNumber (Required)	String International Consignment Number

10.1. SAMPLE REQUEST THAT RETURNS JSON

Below is a sample request that returns JSON. The HTTP Request Header must contain application e.g.) “Accept: application/json”. Ensure the user name and password is Base64 encoded in the authorization header as specified in the Authentication section of this document.

10.1.1. REQUEST

```
GET
https://api.couriersplease.com.au/v1/international/shipment/label?consignmentNumber=CPWEXPA999999999
HTTP/1.1
```

10.1.2. REQUEST HEADER

```
Host: api-test.couriersplease.com.au
Accept: application/json
Authorization: Basic Vzk5OTk5OkFCQzEyMzQ1Njc4OQ==
```

10.2. SAMPLE REQUEST THAT RETURNS XML

Below is a sample request that returns XML. The HTTP Request Header must contain application e.g.) "Accept: application/xml". Ensure the user name and password is Base64 encoded in the authorization header as specified in the Authentication section of this document.

10.2.1. REQUEST

```
GET
https://api-test.couriersplease.com.au/v1/international/shipment/label?consignmentNumber=CPWEXPA999999999
HTTP/1.1
```

10.2.2. REQUEST HEADER

```
Host: api.couriersplease.com.au
Accept: application/xml
Authorization: Basic Vzk5OTk5OkFCQzEyMzQ1Njc4OQ==
```

11. RESPONSE

Each request will return both an HTTP Status Code and a Response Code, which can be used to understand if the request was successfully processed. Either JSON or XML is returned based on the request Accept header.

11.1. HTTP STATUS CODES

Code	Details	Action
200	The request was successfully processed.	
400	There was an issue with the input request or data.	Based on the responseCode, an exception should be raised in the integrating application.
401	There was an issue with authorization.	Based on the responseCode, an exception should be raised in the integrating application.
500	The request could not be processed because of an internal API issue.	Retry the request. If the problem persists please contact CouriersPlease API support apisupport@couriersplease.com.au

11.2. RESPONSE CODES

Code	Details	Action
SUCCESS	The request was successfully processed.	
INVALID_INPUT	The request could not be completed because the input provided was invalid.	The msg element of the response will include the error message.

UNAUTHORIZED	The request could not be completed due to insufficient access.	The msg element of the response will include the error message.
SERVICE_UNAVAILABLE	The service is unavailable.	Raise an exception.

11.3. OUTPUT

The output returned from this API will return a Base64 encoded shipping label and a Base64 encoded commercial invoice in PDF format.

A successful API response where HTTPS Status Code = 200 and Response Code = SUCCESS will return an object structured as follows.

Name	Description
label	Base64 encoded label in PDF format **See Label Printing Instructions Section of this document before printing.**
Invoice	Base64 encoded commercial invoice in PDF format **See Label Printing Instructions Section of this document before printing.**

11.4. SAMPLE JSON RESPONSE (SUCCESS – 200)


```
{
 "responseCode": "SUCCESS",
 "msg": "",
 "data": {
  "label": "JVBERi0xLjQKJeLjz9MKMiAwIG9iago8PC9UeXBlL1hPYmplY3QvU3VidHl",
  "invoice": "JVBERi0xLjQKJeLjz9MKMyAwIG9iago8PC9MZW5ndGggMTUxMy9GaWx0ZX"
 }
}
```


11.5. SAMPLE XML RESPONSE (SUCCESS – 200)


```
<ResponseTemplateOfInternationalLabelResponse xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <responseCode>SUCCESS</responseCode>
  <msg />
  <data>
    <label>JVBERi0xLjQKJeLjz9MKMiAwIG9iago8PC9UeXBlL1hPYmplY3QvU3VidHl</label>
    <invoice>JVBERi0xLjQKJeLjz9MKMyAwIG9iago8PC9MZW5ndGggMTUxMy9GaWx0ZX</invoice>
  </data>
</ResponseTemplateOfInternationalLabelResponse>
```

11.6. SAMPLE JSON RESPONSE (INVALID INPUT - 400)


```
{
 "responseCode": "INVALID_INPUT",
 "msg": "Missing consignment number",
 "data": ""
}
```


11.7. SAMPLE XML RESPONSE (INVALID INPUT - 400)


```
<ResponseTemplateOfString xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <responseCode>INVALID_INPUT</responseCode>
   <msg>Missing consignment number</msg>
   <data />
</ResponseTemplateOfString>
```


11.8. SAMPLE JSON RESPONSE (UNAUTHORIZED - 401)


```
{
 "responseCode": "UNAUTHORIZED",
 "msg": "Not a valid token.",
 "data": ""
}
```


11.9. SAMPLE XML RESPONSE (UNAUTHORIZED - 401)


```
<ResponseTemplateOfString xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <responseCode>UNAUTHORIZED</responseCode>
   <msg>Not a valid token.</msg>
   <data />
</ResponseTemplateOfString>
```


GLOSSARY

Term	Definition
API (Application Programming Interface)	A set of computer functions for software. There is no graphical user interface like a widget.
Widget	A graphical user interface for software
Plugin	A software component that adds a feature to an existing computer program. Also called an add-in or add-on.
Domestic	Freight shipped within Australia
International	Freight shipped to and/or from outside of Australia
Sandbox	API environment for testing and integration
Production	Live API environment