

# CouriersPlease Domestic Label API Documentation



CouriersPlease

API Version: 1.0.0

## 1. VERSION CONTROL

Version	Date	Notes	Author
<b>1.0.0</b>	16/06/2016	Initial version.	Jeff Embro (CouriersPlease)
<b>1.0.1</b>	26/10/2016	Fixed urls.	Jeff Embro (CouriersPlease)

## 2. TABLE OF CONTENTS

1. Version Control.....	1
2. Table of Contents.....	1
3. Overview.....	2
4. Consumes .....	2
5. Output Formats .....	2
6. HTTP Method.....	2
7. URLs .....	2
7.1. Sandbox.....	2
7.2. Production.....	2
8. Security .....	2
8.1. Authentication .....	3
8.2. SSL Encryption .....	3
9. Request.....	3
9.1. Sample Request That Returns JSON.....	3
9.1.1. Request URL.....	3
9.1.2. Request Header .....	3
9.2. Sample Request That Returns XML.....	3
9.2.1. Request URL.....	4
9.2.2. Request Header .....	4
10. Response .....	4
10.1. HTTP Status Codes .....	4
10.2. Response Codes .....	4
10.3. Output.....	5
10.4. Sample JSON Response (Success – 200).....	5
10.5. Sample XML Response (Success – 200).....	5
10.6. Sample JSON Response (Invalid Input - 400).....	5
10.7. Sample XML Response (Invalid Input - 400).....	6
10.8. Sample JSON Response (Unauthorized - 401).....	6
10.9. Sample XML Response (Unauthorized - 401).....	6
Glossary .....	6

### 3. OVERVIEW

This document defines the API in which E-Merchants can retrieve the shipping labels for a Domestic consignment. These labels are to be affixed to the freight before shipping.

Please ensure you follow the CouriersPlease label printing specifications.

### 4. CONSUMES

JSON

### 5. OUTPUT FORMATS

JSON

XML

### 6. HTTP METHOD

GET

### 7. URLS

#### 7.1. SANDBOX

While developing and testing use the sandbox URL:

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

#### 7.2. PRODUCTION

After deploying to live use the production URL:

<https://api.couriersplease.com.au/v1/domestic/shipment/label?consignmentNumber=>

### 8. 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](mailto:apisupport@couriersplease.com.au).

## 8.1. 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=="

## 8.2. 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)

## 9. REQUEST

Name	Type
consignmentNumber (Required)	String Domestic Consignment Number

### 9.1. 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.

#### 9.1.1. REQUEST URL

```
GET
https://api.couriersplease.com.au/v1/domestic/shipment/label?consignmentNumber=CPWREC999999999
HTTP/1.1
```

#### 9.1.2. REQUEST HEADER

```
Host: api-test.couriersplease.com.au
Accept: application/json
Authorization: Basic VzK5OTk5OkFCQzEyMzQ1Njc4OQ==
```

### 9.2. 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.

### 9.2.1. REQUEST URL

```
GET
https://api-test.couriersplease.com.au/v1/domestic/shipment/label?consignmentNumber=CPWREC999999999
HTTP/1.1
```

### 9.2.2. REQUEST HEADER

```
Host: api.couriersplease.com.au
Accept: application/xml
Authorization: Basic VzK5OTk5OkFCQzEYmZQ1Njc4OQ==
```

## 10. 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.

### 10.1. HTTP STATUS CODES

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

### 10.2. RESPONSE CODES

Code	Details	Action
<b>SUCCESS</b>	The request was successfully processed.	
<b>INVALID_INPUT</b>	The request could not be completed because the input provided was invalid.	The msg element of the response will include the error message.
<b>UNAUTHORIZED</b>	The request could not be completed because the CouriersPlease Account Number and/or the Token is invalid.	The msg element of the response will include the error message.
<b>SERVICE_UNAVAILABLE</b>	The service is unavailable.	Raise an exception.

### 10.3. OUTPUT

The output returned from this API will return a Base64 encoded shipping label 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
<b>label</b>	Base64 encoded label in PDF format  The shipping labels are to be affixed to the freight before shipping.

### 10.4. SAMPLE JSON RESPONSE (SUCCESS – 200)

```
{
  "responseCode": "SUCCESS",
  "msg": "",
  "data": {
    "label":
    "JVBERi0xLjQKJeLjz9MKMiAwIG9iago8PC9UeXBIL1hPYmplY3QvU3VidHlwZS9JbWFnZS9XaWR0aCAxODAvSGVp
    Z2h0IDU5L0xlbmd0aCAxMDU5MC9Db2xvcINwYWNIWy9DYWxsSR0I8PC9HYW1tYVsyLjlgMi4yIDluMI0vTWFOcmI
    4WzAuNDEyMzkgMC4yMTI2NCAwLjAxOTMzIDAuMzU3NTggMC43MTUxN "
  }
}
```

### 10.5. SAMPLE XML RESPONSE (SUCCESS – 200)

```
<ResponseTemplateOfDomesticLabelResponse xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <responseCode>SUCCESS</responseCode>
  <msg />
  <data>

  <label>JVBERi0xLjQKJeLjz9MKMiAwIG9iago8PC9UeXBIL1hPYmplY3QvU3VidHlwZS9JbWFnZS9XaWR0aCAxODAv
  vSGVpZ2h0IDU5L0xlbmd0aCAxMDU5MC9Db2xvcINwYWNIWy9DYWxsSR0I8PC9HYW1tYVsyLjlgMi4yIDluMI0vT
  WF0cmI4WzAuNDEyMzkgMC4yMTI2NCAwLjAxOTMzIDAuMzU3NTggMC43MTUxN</label>

  </data>
</ResponseTemplateOfDomesticLabelResponse>
```

### 10.6. SAMPLE JSON RESPONSE (INVALID INPUT - 400)

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

**10.7. 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>
```

**10.8. SAMPLE JSON RESPONSE (UNAUTHORIZED - 401)**

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

**10.9. 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
<b>API (Application Programming Interface)</b>	A set of computer functions for software. There is no graphical user interface like a widget.
<b>Widget</b>	A graphical user interface for software
<b>Plugin</b>	A software component that adds a feature to an existing computer program. Also called an add-in or add-on.
<b>E-Merchant</b>	Electronic retailer
<b>Domestic</b>	Freight shipped within Australia
<b>International</b>	Freight shipped to and/or from outside of Australia
<b>Sandbox</b>	API environment for testing and integration
<b>Production</b>	Live API environment