Introduction to the UPS Developer Kit

March 07, 2022July 22, 2022

Important Information

UPS Developer Kit APIs

Your development of an application using UPS Developer Kit APIs is governed by the UPS Technology Agreement you entered into with UPS. The following are key legal requirements from these agreements for the UPS Developer Kit APIs. For more information on all requirements for the UPS Developer Kit APIs, please refer to the UPS Technology Agreement.

Defined terms used but not defined in this document have the meaning set forth in the UPS Technology Agreement.

Key Legal Requirements for UPS Developer APIs

Permitted Territories

This document can only be used in the countries of the Permitted Territory as defined in the UPS Technology Agreement, as applicable.

Use

The application must not be designed to allow distribution of information received through the UPS Developer Kit APIs to third parties, other than to persons having a bona fide interest in such information (e.g., the shipper, receiver, or the third party payer, or to your service providers authorized by UPS).

Consent to Use of UPS Mark

- All screens or forms generated by your application including information received through the UPS Developer Kit APIs must include (1) the UPS Mark positioned in reasonable proximity to the Information and of an appropriate size to readily identify the source of the Information as UPS and (2) the following language at the bottom of every screen that displays the UPS Mark: "UPS, the UPS brand mark, and the Color Brown are trademarks of United Parcel Service of America, Inc. All Rights Reserved." Except as set forth in the preceding sentence, you have no right to use the UPS Mark without the prior written approval of UPS.

- You shall not use the UPS Mark in association with any third party trademarks in a manner that might suggest co-branding or otherwise create potential confusion as to source or sponsorship of the application, or ownership of the UPS Mark.

- The UPS Mark shall be used only as provided by UPS electronically or in hard copy form. The UPS Mark may not be altered in any manner, including proportions, colors, elements, etc., or animated, morphed or otherwise distorted in perspective or dimensional appearance.

- The UPS Mark may not be combined with any other symbols, including words, logos, icons, graphics, photos, slogans, numbers, or other design elements. A minimum amount of empty space must surround the UPS Mark separating it from any other object, such as type, photography, borders, edges, etc. The required area of empty space around the UPS Mark must be 1/3x, where x equals the height of the UPS Mark.

Copyright and Proprietary Notice

In your application and any POD Letters you prepare, you must include a prominent reproduction of UPS's copyright and proprietary notices in a form and format specified by UPS (See the Copyright section of this document).

Display of Information

The application must not display information concerning any other provider of shipping services or such other shipping services on any page, whether comprising one or more frames, displaying information your application receives from the UPS Developer Kit APIs. Your application must present all data within each field received through the UPS Developer Kit APIs without amendment, deletion, or modification of any type.

Notice

In all communications with UPS concerning this document, please refer to the document date located on the cover.

Copyright

© 2022 United Parcel Service of America, Inc. All Rights Reserved. Confidential and Proprietary

The use, disclosure, reproduction, modification, transfer, or transmittal of this work for any purpose in any form or by any means without the written permission of United Parcel Service is strictly prohibited.

Trademarks

Some of the UPS corporate applications use United States city, state, and postal code information obtained by United Parcel Service of America, Inc. under a non-exclusive license from the United States Postal Service.

Table of Contents

Chapter 1: Key Information

This document contains key information applicable to all UPS Developer Kit APIs and their respective Developer Guides.

Intended Audience

This guide is intended for developers who will be using one or more of the UPS Developer Kit APIs.

Required Knowledge & Skills

This guide assumes you have a basic understanding of Web Services, XML syntax and structure, and JSON syntax and structure.

Keeping Up-To-Date

- UPS suggests you periodically review and update your My UPS profile contact information.

- UPS releases new features and updated APIs along with the guides twice a year. In addition, maintenance updates may also be released intermittently throughout the year. These changes are documented in the *Release Features* section of the API Developer Guide.

- Announcements and release information can be found in the *Announcements and API Updates* section at UPS.com.

Getting Started

The resources below are available at UPS.com to assist you in developing your UPS APIs.

Resource	Information
UPS Developer Kit Access: UPS Developer Kit	This section of ups.com® provides access for developers to: • Request an Access Key to gain access to UPS test and production API servers. • Access and download developer API kits containing developer guides, schema, sample code, and UPS logos.
UPS Developer Resource Center Access: Developer Resource Center	**Integration/Planning** Learn About Integrating UPS Developer Kit APIs Plan Your Integration **Getting Started/Instructions** Learn how to access UPS Developer Kit and servers to test and transact with UPS. View the UPS Developer Kit User Guide **Download APIs** Obtain and manage your API credentials and download developer kits. Access the UPS Developer Kit Access Gemini Downloads and Tool Kit

Resource	Information
Technical Support For additional information, refer to the Technical Support chapter.	Send us an email with your question and attach a file with your XML/SOAP/JSON request and xml response and error messages using the link below. Our goal is to reply to your message within four hours during support business hours. Please review all documentation, including the Developer's Guide and Knowledge Base PDF document, before you contact UPS for support. E-mail UPS General Questions: Call toll free at 800-247-9035 (Monday-Friday, 7:30 a.m. to 9 p.m. ET; Saturday-Sunday, 9 a.m. to 6 p.m. ET) Integration/Production Support: Email us using the support form at left and receive a response within four hours during business hours (Monday-Friday, 8 a.m. to 8 p.m. ET)

Additional Resources

Resource	Information
XML Syntax and Structure Tutorial	W3 Schools XML Tutorial http://www.w3.org/XML/
Web Services Tutorial	W3 Schools Web Services Tutorial http://www.w3schools.com/webservices/
JSON Syntax and Structure Tutorial	W3 Schools JSON Tutorial http://www.w3schools.com/json/

Chapter 2: API Basics

UPS APIs offer businesses a simple yet powerful way to integrate UPS functionality. From shipping rate requests to shipping orders, the APIs allow virtually any application, service, or system that requires instant, up-to-date access to a UPS functionality, a straight-forward means to do so.

UPS provides three types of APIs, based on the request-response message framework: XML, SOAP, and JSON.

The APIs offered by UPS are classified by four key characteristics:

API functionality	Examples: Rate, Ship, Track
API request-response framework	XML, SOAP, JSON
Integration by country or territory	The countries or territories that the API may be legally integrated with are listed in the Availability section, on the respective UPS Developer Kit API page at UPS.com. Note: This information is subject to change.
Functionality by country or territories	Countries or territories that support the API functionality are found in the Appendix of the API's Developer Guide.
	Example: The Shipping API lists the Origin Ship-From countries or territories that are supported by the functionality within the API. In this example, the shipping API may not be integrated in Azerbaijan but a shipment can be sent from Azerbaijan to all UPS destinations using the Shipping API.

UPS Integration Environments

UPS has two environments for use during the development process, the Customer Integration Environment (CIE) for testing and bug fixes, and the Production environment for going live.

📄 All API requests are submitted in the form of an HTTP POST. Due to the nature of the information being transmitted, it is recommended that you use a secure HTTP request (HTTPS).

📄 IMPORTANT NOTE: Do not use CIE as a production environment. It does not support all API functionality and contains only limited data.

📄 IMPORTANT NOTE: Do not engage in any load or performance testing against the UPS CIE or Production environments. Doing so may result in your access being revoked.

Supported Standards

- Extensible Markup Language (XML)

- Simple Object Access Protocol (SOAP)

- JavaScript Object Notation (JSON) as described by RFC 7159 and ECMA-404

- Web Service Description Language (WSDL)

- Hypertext Transfer Protocol (HTTP)

- Transport Layer Security (TLS)

XML API Overview

The XML API allows you to make web requests directly to the UPS system in the form of an HTTP POST. Include the following HTTP content-type header in your request:

Content-Type: Application/x-www-form-urlencoded

The endpoint URLs for each API differs by the UPS functionality they are accessing and the request-response type. Endpoint URLs can be found in the Developer Guide for the API.

XML schema and more specifically element names differ from API to API unless otherwise noted. Unless explicitly verified, the schema (element names and hierarchy) for the XML version of the API differs from the SOAP version of the API.

📄 XML element names are case sensitive. Address, address, and aDDress represent 3 different elements.

XML API Request

The body of the API request consists of two distinct XML documents:

- Access request
- Functionality request

Each part must be its own well-formed XML document with the access request positioned before the functionality request.

A valid, well-formed XML request complies with the following guidelines:

- XML syntax must be correct
- Element names must be correct
- XML hierarchy must be correct
- Values associated with each element must be valid

The containers and elements of the functionality request will differ for each XML API. The schema is documented in detail in the API-specific Developer Guides.

The access request contains the authentication credentials portion of the request to the UPS system. It consists of a single container named AccessRequest that holds the elements used to identify the message as originating with an authorized customer. The access request will be the same across all functionality types.

XML Request Schema
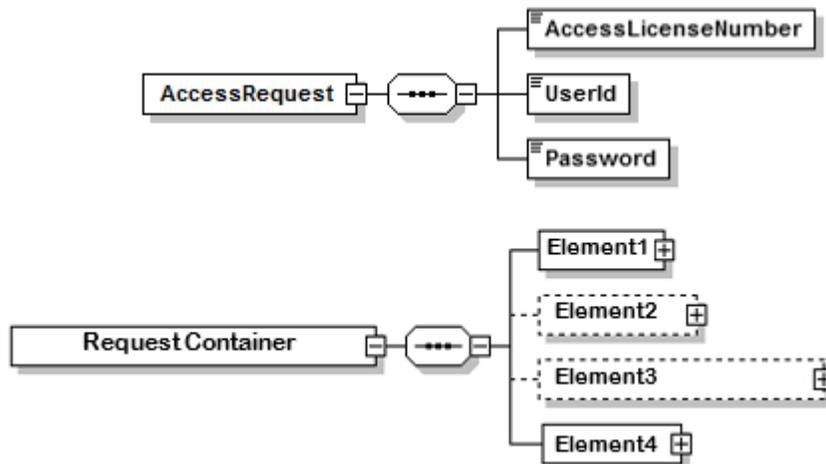
XML Elements

Name	Constraint	Description
/AccessRequest		
AccessRequest	Required: Yes Type: Container Max Allowed: 1 Length: N/A	Holds the authentication credentials.
/AccessRequest/AccessLicenseNumber		
AccessLicenseNumber	Required: Yes Type: Container Max Allowed: 1 Length: N/A	Holds the UPS Access Key associated with the UPS account.
/AccessRequest/UserId		
UserId	Required: Yes Type: String Max Allowed: 1 Length:	Holds the username associated with your My UPS account.
/AccessRequest/Password		
Password	Required: Yes Type: String Max Allowed: 1 Length:	Holds the password associated with your My UPS account.

XML Access Request Example

📄 NOTE: XML element names are case sensitive.

```
<?xml version="1.0" ?>
<AccessRequest xml:lang='en-US'>
    <AccessLicenseNumber>YOURACCESSLICENSENUMBER</AccessLicenseNumber>
    <UserId>YOURUSERID</UserId>
    <Password>YOURPASSWORD</Password>
</AccessRequest>
```

XML API Response

The body of the XML API response, similar to the request, contains an XML document. The schema for the document varies based on the functionality type. For example, the schema and values in the response of a Rate request will differ from that of a Tracking request. Response details are discussed in the Developer Guide for each API.

SOAP API Overview

The SOAP API relies on standard Web Services technology as defined by W3C.org and OASIS for submitting requests and handling responses. A set of WSDLs for each API is available in the UPS Developer Kit. The WSDLs define the exact services exposed by the API along with the location of the endpoints.

📄 UPS WSDLs are included in the Developer Kit files. They are not published through a public repository.

📄 UPS DEWSDLs do not have embedded schemas; they contain references to the Schema files that are distributed with the WSDLs.

As with the XML API, a schema defines the valid structure and legal values of both request and response content. In addition, SOAP defines three required elements that act as containers for all other parts of the HTTP message: envelope, header, body.

The envelope of the SOAP request contains the following:

- Header
    - o Security block
- Body
    - o Common block: information common to all UPS SOAP APIs
    - o Request block: information associated with the functionality request

SOAP API Request

The security header, UPSSecurity, provides a mechanism for attaching authentication credentials to the request.

The header holds the security block with 2 containers as follows:

SOAP Request Security Block Schema
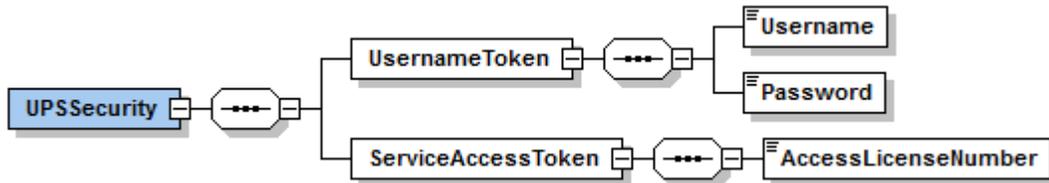
Name	Constraint	Description
/UPSSecurity		
UPSSecurity	Required: Yes Type: Container Max Allowed: 1 Length: N/A	Request Container.
/UPSSecurity/UsernameToken		
UsernameToken	Required: Yes Type: Container Max Allowed: 1 Length: N/A	Container used to store the username and password.
/UPSSecurity/UsernameToken/Username		
Username	Required: Type: Max Allowed: Length:	

Name	Constraint	Description
/UPSSecurity/UsernameToken/Password		
Password	Required: Type: Max Allowed: Length:	
/UPSSecurity/ServiceAccessToken		
ServiceAccessToken	Required: No Type: Container Max Allowed: 1 Length: N/A	Container used to store the AccessLicenseNumber.
/FreightRateRequest/Request/ServiceAccessToken/AccessLicenseNumber		
AccessLicenseNumber	Required: Type: Max Allowed: Length:	

SOAP Security Header Example

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:upss="http://www.ups.com/XMLSchema/XOLTWS/UPSS/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:common="http://www.ups.com/XMLSchema/XOLTWS/Common/v1.0">
    <env:Header>
        <upss:UPSSecurity>
            <upss:UsernameToken>
                <upss:Username>Your User Id</upss:Username>
                <upss:Password>Your Password</upss:Password>
            </upss:UsernameToken>
            <upss:ServiceAccessToken>
                <upss:AccessLicenseNumber>Your Access License</upss:AccessLicenseNumber>
            </upss:ServiceAccessToken>
        </upss:UPSSecurity>
    </env:Header>
</envr:Envelope>
```

SOAP Request Body Schema

The full request schema for the SOAP body is discussed in detail in the API-specific Developer Guide.
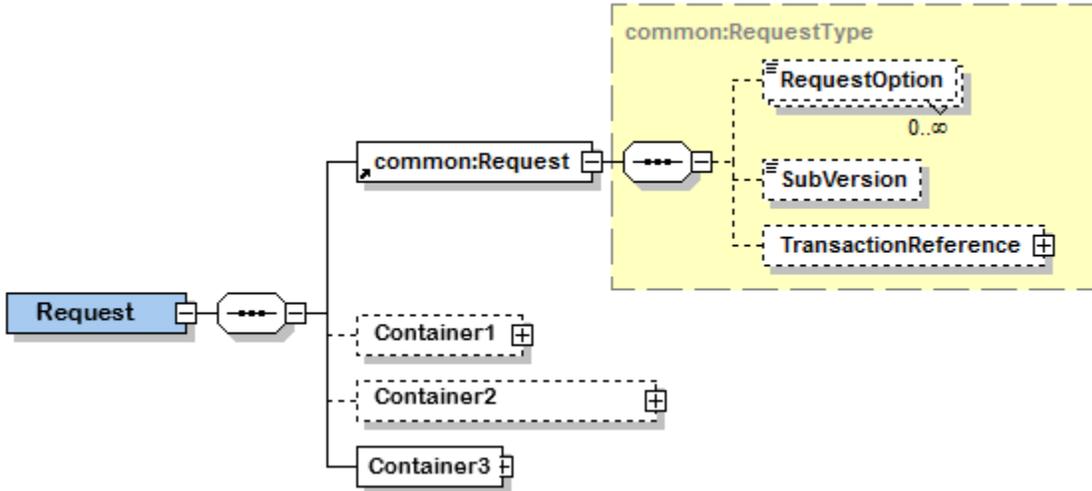
SOAP API Response

The body of the SOAP response returns the UPS response message. The elements and values differ based on the requested functionality. Details and schema for the response are discussed in the Developer Guide for the API.

JSON API Overview

The JSON API uses the same underlying data model and standard objects as those in the SOAP Web Service/XML API. Similarly, it allows you to make a request to the UPS system via an HTTP/s POST.

Unlike SOAP Web Service/XML, JSON defines a lightweight data interchange format consisting of attribute-value pairs. The format is based on a subset of the JavaScript programming language and supports a simplified java-like format. It does not define any of the standard SOAP elements (*envelope, header, body*).

UPS Web Services also support REST - JSON requests. The structure of JSON and SOAP requests and responses are similar except for a few elements.

IMPORTANT:

- For additional XPath and API information, refer to the appropriate Web Service or XML Developer Guide.

- For a list of all errors applicable to the API, refer to the Error Codes topic in the Appendix of the Web Service or XML Developer Guide.

The JSON API expects the following HTTP headers:

Access-Control-Allow-Headers:	Origin, X-Requested-With, Content-Type, Accept
Access-Control-Allow-Methods:	POST
Access-Control-Allow-Origin:	*
Content-Type:	application/json

Differences between SOAP and JSON

UPS Web Services also support REST – JSON requests. Except for a few elements, the structure of JSON and SOAP requests and responses are similar.

SOAP specific elements

Do not include the following SOAP elements in JSON requests:

- Envelope
- Header
- Body

Lists

To create a XML – SOAP list, repeat the element name:

 <Element>value1</Element>

 <Element>value2</Element>

 <Element>value3</Element>

To create JSON list, repeat only the value:

 Element: [value1, value2, value3]

JSON Headers

The JSON API expects the following HTTP headers:

Access-Control-Allow-Headers:	Origin, X-Requested-With, Content-Type, Accept
Access-Control-Allow-Methods:	POST
Access-Control-Allow-Origin:	*
Content-Type:	application/json

SOAP vs JSON Example

SOAP Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://www.ups.com/XMLSchema/XOLTWS/UPSS/v1.0"
xmlns:v11="http://www.ups.com/XMLSchema/XOLTWS/Login/v1.1"
xmlns:v12="http://www.ups.com/XMLSchema/XOLTWS/Common/v1.0">
    <soapenv:Header>
        <v1:UPSSecurity>
            <v1:UsernameToken>
                <v1:Username>username1</v1:Username>
                <v1:Password>password1</v1:Password>
            </v1:UsernameToken>
            <v1:ServiceAccessToken>
                <v1:AccessLicenseNumber>accessLicenseNumber1</v1:AccessLicenseNumber>
            </v1:ServiceAccessToken>
        </v1:UPSSecurity>
    </soapenv:Header>
    <soapenv:Body>
        <v11:LoginAcceptTermsAndConditionsRequest>
            <v12:Request/>
        </v11:LoginAcceptTermsAndConditionsRequest>
    </soapenv:Body>
</soapenv:Envelope>
```

JSON Request

```
{"UPSSecurity": {
            "UsernameToken": {
                "Username":" username1",
                "Password":" password1"
            },
            "ServiceAccessToken": {
                "AccessLicenseNumber":"AccessLicenseNumber1"
            }
        },
        "LoginAcceptTermsAndConditionsRequest": {
            "Request":"" }}
```

Chapter 3: Error Responses

When the UPS system is unable to respond to a request, be it from a malformed request, an illegal or invalid value, or other issues, the API generates an error response.

For additional help and support with API errors, first consult the Developer Guide for the API. If you are still unable to resolve the issue, UPS Email Support is available for technical questions. To find out more about UPS Technical Support, review the Technical Support chapter in this document.

API Error Response Types

In general, an API will return one of the four following response types:

- **Success without warning** - Indicates the request has been processed as anticipated.

- **Success with warning** - Indicates the UPS system processed the request with potentially unanticipated results.

- **Transient Error** - Indicates an error that is temporary in nature. Such errors may be caused by response timeouts due to high server loads or scheduled maintenance in progress. The request may be re-issued successfully at a later time.

- **Hard Error** - Indicates the request has a problem that the system is not able to resolve. Also known as a critical error, hard errors prevent the system from handling the request.

📄 **IMPORTANT NOTE**: do not resubmit requests with Hard Errors until the problems causing the error have been resolved.

When an error occurs, the system responds with an error code, severity, and description that define the general problem encountered.

- **Error Code -** A unique numerical value used to identify the error. Error code 1001, for example, is associated with a hard error and indicates the XML document is not well formed.

- **Error Severity** - The error type, either hard or transient as defined above.

- **Error Description** - A description of the general problem. For example, "The XML document is not well formed" is the description associated with Error Code 10001.

Refer to the Common Errors section of this document for a discussion on common errors associated with all APIs.

A complete list of API specific error codes is included in the Appendix of the Developer guide for the API.

Debugging an API Request

Debugging an API request to resolve an execution error is no different from any other debugging exercise. It is an iterative process where each error is identified and fixed until no further errors exist.

The API, in most cases, returns the error information for the first problem it encounters. Resolving this first issue, allows the system to expose the next problem, if one exists. Continue to resolve each issue as they are returned until the API is able to handle the request without error.

XML API Errors

In the response from the XML API, the value returned in the **ResponseStatusCode** element indicates the success or failure of the request.

When the API is able to respond successfully, with or without warnings, the ResponseStatusCode value is 1, otherwise the value is 0.

When an error occurs, the **ErrorCode**, **ErrorSeverity**, and **ErrorDescription** elements are returned in the Error container. For some elements, the name of the element where the problem first occurred is returned in the **ErrorLocation**.

XML Error Response Schema
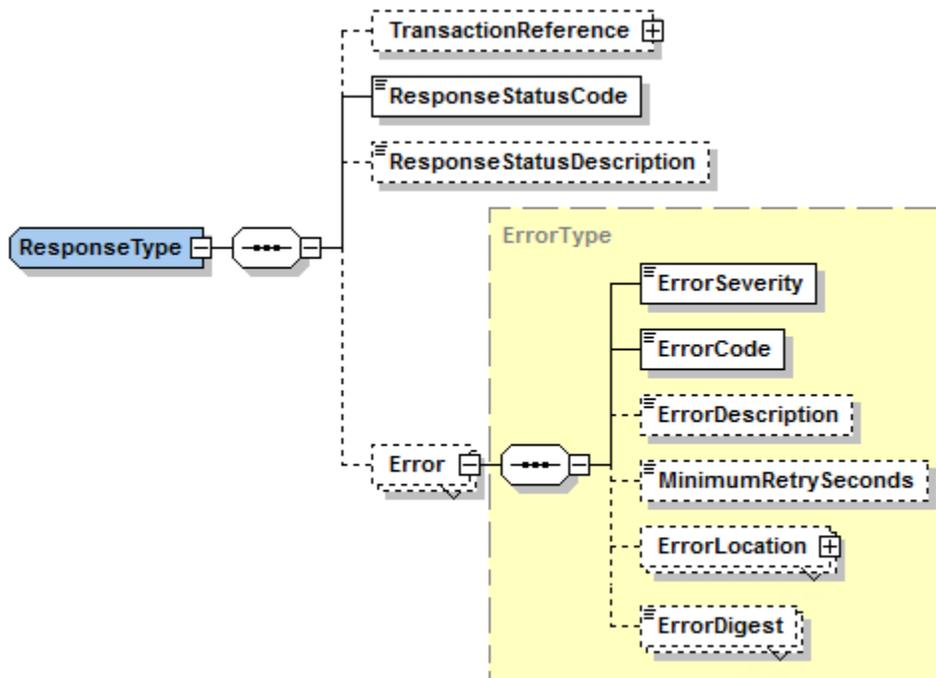
XML Error Response Example

Warning messages are returned for the *Success with Warning* response type in the ResponseStatusDescription element.

```
<?xml version="1.0"?>
<ShipmentConfirmResponse>
<Response>
    <TransactionReferece>
        <CustomerContext>Request Types</CustomerContext>
    </TransactionReferece>
    <ResponseStatusCode>0</ResponseStatusCode>
    <ResponseStatusDescription>Failure</ResponseStatusDescription>
    <Error>
        <ErrorSeverity>Hard</ErrorSeverity>
        <ErrorCode>120508</ErrorCode>
        <ErrorDescription>Missing/Invalid shipment pickup date</ErrorDescription>
    </Error>
</Response>
</ShipmentConfirmResponse>
```

SOAP Errors

A SOAP response will take one of two forms:

- Success response

- Error response

When a SOAP request responds successfully, it returns a ResponseStatus container and the value of the ResponseStatus\Code element is 1. The Description element in the ResponseStatus container may or may not contain warnings.
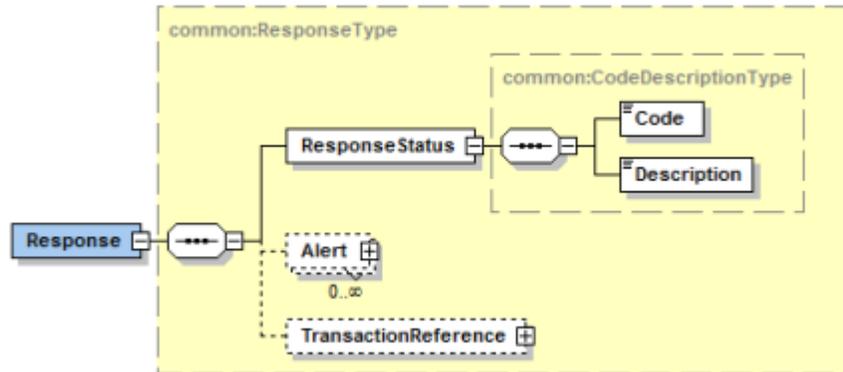
However, when an error is generated, the SOAP API returns the error and status information in a standard SOAP Fault element. The SOAP Fault is only present if there is an error.

Fault Element Description

The fault contains four elements: *faultcode*, *faultstring*, *faultactor*, and *detail*.

faultcode - An element that returns a small set of fault codes intended to identify the fault. The faultcode must be present in a SOAP fault element and the value must be a qualified name. SOAP defines the following four fault codes:

Fault Codes	Description
VersionMismatch	The message sent by the client used a version of the SOAP protocol the UPS API does not understand.
MustUnderstand	The message sent by the client includes an element in the header the UPS API does not understand.
Client	The request sent by the client application is not valid.
Server	Although the client application's request did not have any errors itself, the UPS API encountered an error when trying to process the request.

faultstring - Contains an explanation of the fault and a basic description of the error.

faultactor - Indicates the cause of the problem, a URI is typically returned.

detail - Contains the API-specific Error Code, Severity, and Description returned by the UPS system.

SOAP Error Response Example

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/enveloper/">
<soapenv:Header/>
    <soapenv:Body>
        <soapenv:Fault>
            <faultcode>Client</faultcode>
            <faultstring>An exception has been raised as a result of client data.</fautlstring>
            <detail>
                <err:Errors xmlns:err="http://www.ups.com/schema/xpci/1.0/error">
                <err:ErrorDetail>
                    <err:Severity>Hard</err:Severity>
                    <err:PrimaryErrorCode>
                        <err:Code>120108</err:Code>
                        <err:Description>Missing or invalid Shipper country code</err:Description>
                    </err:PrimaryErrorCode>
                </err:ErrorDetail>
                </err:Errors>
            </detail>
        </soapenv:Fault>
    </soapenv:Body>
</soapenv:Envelope>
```

JSON Errors

The API response to a JSON request is either success or failure. In the event of a failure, there are two possible response formats:

- Simple JSON Error
- JSON Error Fault

IMPORTANT: For a list of all errors applicable to the API, refer to the Error Codes topic in the Appendix of the Web Service or XML Developer Guide.

Simple JSON Error

A simple error response returns the error information in a simplified format consisting of two keyword-value pairs, one for the error code and one for the description.

```
{
    "Error": {
        "Code": "2",
        "Description": "Unknown operation"
    }
}
```

JSON simple errors are as follows:

Error Code	Description
1	**Unrecognized Error** Any error, other than those listed below, causing an issue with the request. • Verify the request headers are correct. • Retry the request again later.
2	**Unknown Operation** The UPS system does not understand the operation and is unable to forward the request to the correct API. • Verify the request headers are correct. • Check request structure.
3	**Network Error** Transient error indicating the network is hindering request submission. • Retry the request again later.
4	**JSON Syntax Error** Indicates a problem with the JSON syntax. • Verify your request syntax is correct.

JSON Error Fault

In addition to a simple error, the JSON API may return an Error Fault. This Fault is modeled on the standard SOAP Fault except the data elements are returned in the JSON format. As with the SOAP Web Services/XML API, the JSON Fault element will only be returned when there is error with the request.

A detailed explanation of a Fault, its elements, and possible values, may be found in the SOAP Error section of this document.

Similar to the SOAP Fault, the API-specific information (error Code, Severity, and Description) are found in the detail section of the JSON Fault.

IMPORTANT: For a list of all errors applicable to the API, refer to the Errors topic in the Appendix of the Web Service or XML Developer Guide.

JSON Fault Response Example

```
{
    "Fault": {
        "faultcode"; "Client",
        "faultstring": "An exception has been raised as a result of client data.",
        "detail: {
            "Errors": {
                "ErrorDetail":
                {
                 "Severity": "Authentication",
                 "PrimryErrorCode": {
                     "Code": "250002",
                     "Description": "Invalid Authenticatoin Information."
                     }
                }
            }
        }
    }
}
```

Common Errors

Common causes of Hard Errors include the following:

- Sending a request to the wrong endpoint such as a *Ship* request sent to the *Rate* endpoint.
- Missing containers or elements that are required in the request.
- Missing or invalid values that are required in the request.
- Misspelled or wrong case in XML element names or JSON attribute names.
- Malformed requests.

The common errors listed in the table below comprise the errors that may apply to all APIs.

Error Code	Severity	Description
10001	Hard	**The XML document is not well formed.** Indicates a problem with XML document syntax. • Verify the XML syntax.
10002	Hard	**The XML document is well formed but the document is not valid.** Indicates the XML document does not conform to the schema for the request. It may also signal you are sending the request to the wrong endpoint e.g. Rate request sent to the Shipping endpoint. • Check the request structure. • Refer to the schema for information on the request structure.
10003	Hard	**The XML document is either empty or null.** Indicates an empty request is being submitted to the API. • Verify your HTTP/s request is being populated.
10006	Hard	**Although the document is well formed and valid, the element content contains values that do not conform to the rules and constraints contained in the specification.** A sign there are problems with the container contents or element values. • Verify the contents and values in the request. • Verify the constraints have been followed for the request type. • Refer to the schema for additional constraint information.
20001	Transient	**General process failure.** A general error has occurred that is not attributable to the request. • Retry your request later.
250000	Hard	**No XML declaration in the XML document.** Indicates a syntax problem where the declaration portion of the XML document is not recognized. This may also signal that the request is being sent to the wrong endpoint. • Verify the XML document syntax.
250001	Hard	**Invalid Access License for the tool. Please re-license.** Indicates there is an issue with the Access License. • Verify the license is correct. • Verify you have permission to access the API for the given environment. • Email Support if the problem persists.

Error Code	Severity	Description
250002	Hard	**Invalid Authentication Information.** Authentication credentials are wrong. • Verify the UserId/Username and password are correct.
250003	Hard	**Invalid Access License number.** Indicates an issue with the Access License Key. • Verify the number is correct. • Verify you have permissions setup to access the API for the given environment. • Email Support if the problem persists.
250004	Hard	**Incorrect UserId or Password.** Indicates the user / password combination is incorrect. • Verify the Username/UserId and password are correct.
250005	Hard	**No Access and Authentication Credentials provided.** Indicates the system was unable to identify either the Access License Key or the user / password portion of your request. • Verify the AccessRequest container, UserId, and Password values and syntax for the XML API. • Verify the UsernameToken and ServiceAccessToken values and syntax for the SOAP and JSON APIs.
250006	Hard	**The maximum number of user access attempts was exceeded.** • Indicates too many requests with invalid or wrong credentials have been submitted. • Verify your user credentials.
250007	Hard	**The UserId is currently locked out; please try again in 24 hours.** Indicates the system has locked your account. • Retry your request later. • Email Support if the problem persists.
250009	Hard	**License Number not found in the UPS database.** Indicates the Access License Key in the request is not recognized. • Verify the Access License Key number is correct. • Email Support if the problem persists.
250050	Transient	**License system not available.** A license error has occurred that is not attributable to the request. • Retry your request later.

Chapter 4: Technical Support

UPS offers multiple avenues for getting help and technical support, from self-help documents in the Developer Resource Center to email support.

If you are unable to resolve your questions or issues through the self-help channels, UPS offers Email Support for technical questions including API questions and production issues.

Email Support

Email support is the primary technical support channel for all UPS Developer Kit and API issues, if they cannot first be resolved through self-help channels. In addition, it is the primary support channel for production related issues.

Using email support, questions can be described in detail and the xml request/response file can be sent to help troubleshoot issues.

Access the UPS Developer Kit and click on the E-mail UPS link (left menu). Then follow the instructions to submit a trouble-ticket.

In most cases, a Support Agent will contact you within four business hours of receiving your email. This is during the business hours outlined below. Otherwise, you will be contacted on the following business day.

Technical Support hours and response times

1st Level Email Technical Support

1st Level Email Tech Support email response time is planned to be within 4 business hours of the initial request. Business hours are defined as hours of operation of the tech support desks when service technicians are working.

- Hours of operation for 1st Level Email Tech Support: 8:00 a.m. to 8:00 p.m. EST, Monday through Friday.

- Hours of operation for 1st Level Phone Support:

    o 7:30 a.m. to 9:00 p.m. EST Monday through Friday

    o 9:00 a.m. to 6:00 p.m. EST Saturday and Sunday

- The desk is closed on UPS Holidays

📄 This means if you submit an email tech support request after business hours the 4 hour response clock will not begin until the following morning at 8:00 a.m. EST M-F and 9 a.m. EST Saturday and Sunday (UPS Holidays excluded).

📄 If you are sent an email from 1st Level Email Tech Support stating your case log # has been escalated, the 24 hour planned response clock starts from the time you receive the email from 1st Level Email Tech Support (provided the Escalation Support desk is open). If not, then the clock will begin when the Escalation Support desk opens.

Escalation Support

- Hours of operation: 8:00 a.m. to 8:00 p.m. EST Monday through Friday. The desk is closed weekends and on UPS Holidays.

- Escalation Support desk response time is planned to be within 24 hours during normal hours of operation.

📄 This means if your case log is escalated at 8:30 p.m. on Friday, you will not receive an email response from the Escalation Support desk at the latest until Monday at 8:00 p.m. (Saturdays, Sundays, and Holidays are excluded). Response times are from the time the case log was escalated plus 24 hours that do NOT include the weekends or UPS holidays.

There may be some scenarios when the tech support team is meeting planned response times but it appears as if it is taking too long. Below is a brief example where the planned response time is met but nothing happened over the weekend.

Example

Customer submits tech support email at 7:00 a.m. on Friday. The clock for the 4-hour response starts at 8:00 a.m. EST for the 1st level tech support team. They respond within 4 hours at 12:00 p.m. EST the same Friday stating the case log has been escalated and provide the case log number.

The escalation desk receives the case log at 11:30 a. m. EST. They begin work on evaluating the problem and update the customer within 24 hours or solve the problem within 24 hours. Either way, the customer will hear from the escalation desk within 24 hours with some type of disposition on their case log.

The escalation desk clock begins when they receive a case log from 1st level tech support during business hours, M-F 8:00 a.m. to 8:00 p.m. EST, in this case, the clock begins at 11:30 a.m. EST. If the escalation desk does not have an answer by 8:00 p.m. EST on Friday, the desk has until 11:30 a.m. EST the following Monday to provide a resolution or a status update to the customer on the case log.

The Escalation Desk continues work on the case log on Monday beginning at 8:00 a.m. EST. They receive resolution at 10:30 a.m. EST Monday and send the resolution to the customer. The Escalation Desk has met their 24 hour planned response time.

Non-US Technical Support

📄 **NOTE**: Email response times for non-US requests vary. Translations and escalations may delay the process. If a quicker response is required, please access the US web site directly and complete an email technical support form from the US web site.

Maintenance & System Outage Notices

Chapter 5: FAQs

Common FAQs

Question	Answer
Is XML coding knowledge required for the UPS Developer Kit APIs?	Prior knowledge of XML and other programming languages is required. The main programming requirement for the XML and SOAP APIs is the ability to code a well-formed and valid XML document along with an application to submit a request to UPS via HTTP/s POST and capture the UPS response. Depending upon business needs and programming knowledge, the application can be simple or elaborate. UPS does not provide coding support. However, code samples are supplied in the Developer Kits that may help guide application development.
Does UPS support chained or unchained digital certificates?	UPS supports chained digital certificates.
Within what countries or territories are the Developer APIs available?	The APIs are available in the countries or territories listed on UPS.com and vary by API and by country/territory. At the bottom of the API detail page in the UPS Developer Kit area, you will find a listing of countries or territories associated with the API.
Are there any designated system down times for the Developer APIs?	The overall reserved downtime for UPS servers is Saturday 10:00 p.m. ET through Sunday 12:00 p.m. ET. However, the window may be shortened to two 15-minute intervals with one starting at 11 p.m. and the other occurring sometime between 1 and 3 a.m. ET Sunday morning for most weekends. Typically, traffic is handled in such a way that there is very little impact to customers, and any impact that does occur does so during the small 15-minute intervals. This entire time is reserved for maintenance. UPS informs customers of the possibility of experiencing issues during this time. Sunday maintenance includes account-based rates and UPS Freight®.

UPS Developer Kit APIs

Question	Answer
Are there any examples of implementations that we can review to understand how best to utilize the APIs?	Refer to various UPS.com functionalities as examples of how the APIs can be implemented for each function. For example, the Calculate Time & Cost application on UPS.com uses our Rating and Time in Transit APIs; Shipping depicts how the Shipping API could behave. Tracking on UPS.com is an example of how the Tracking API can be integrated.
Are the UPS APIs RESTful?	REST (Representational State Transfer) is an architectural style developed as a simpler alternative to SOAP and WSDL-based Web Services. As such, the SOAP APIs provided by UPS are not RESTful. The XML and JSON APIs may be considered "REST-Like" in that they use stateless operations over an HTTP POST request.
Do the APIs support PHP or Perl with code samples within the Developer Guides or the Developer Kit zip files?	Yes, sample code in PHP and Perl is provided in the Developer's Kit zip file.
Are the Web Services versions of the APIs ASMX based?	No, all Web Services are XML based.

Question	Answer
Can Japanese Kanji character be recognized by UPS Developer APIs?	No.
What does the error message "XML document is well formed but the document is not valid" mean?	The "XML document is well formed but the document is not valid" error message is generally returned when an element in the XML request does not adhere to the formatting defined within the XPath or Schema section of that API-specific Developer Guide. When the API returns this error, the element that is not valid may be found in the ErrorLocationElementName element in the XML response. It may also signal the request is being sent to the wrong URL. The URL for each API is different and can be found in the API-specific Developer Guide.

UPS Branding and Logos

Questions	Answers
How do I obtain higher resolution UPS logos than those available within the UPS Developer Kit APIs?	Customers are not routinely provided UPS logos and graphics without having a design and layout submitted to UPS Brand Management for approval. The UPS Brand website provides the ability to download a limited set of sample images for layout and outlines the process of how to secure an approval for use of these assets. Visit the UPS Brand website at https://www.upsbrandexchange.com for additional information.

Technical Support

Question	Answer
How do I get technical support for the APIs at UPS.com?	Go to ups.com. Under the Support tab, select Technology Support. In the Tools and Resources section, click UPS Developer Resource Center. If your questions are not answered on this page, then click on E-mail UPS.
Is phone support provided for the UPS Developer Kit Developer APIs? If so, what is the number and what are the hours of operation?	Yes. Phone support is provided at 1st Level only for basic questions. This includes integration questions and production questions. However, customer's questions will be referred to a Support Analyst for Email Support. Questions are not answered at the time of call. Phone Support Hours: Monday - Friday 7:30 a.m. to 9:00 a.m. ET Saturday - Sunday 9:00 a.m. to 6:00 p.m. ET 800-247-9035 To learn more about UPS Technical Support, refer to the Technical Support chapter in this document.

Technical Support Form

Question	Answer
Is XPCI a required field within the email support form?	No.
What does XPCI mean and what version should be used in the Developer Kit APIs?	XPCI stands for XML Package Carrier Interface (XPCI) and defines a vocabulary and structure for describing packages, shipments, and the activity details for package carriers and their customers. XPCI consists of a set of DTDs that defines the terminology, transaction enveloping, and XML message definitions. For a client to be XPCI compliant, the client must generate a well-formed XML message that validates against the XPCI DTDs. Several DTDs, organized into three categories, define XPCI: XPCI Vocabulary This DTD defines the basic business vocabulary of XPCI. All tags used in a message are defined in this DTD. XPCI Interchange This DTD defines the transaction-enveloping schema. Every message includes transaction information. XPCI Message Each message has an associated DTD that defines the vocabulary of the message. The version and date pertains to version of the API. However, the APIs were not versioned so they currently do not carry significance. UPS will use these elements in the event the UPS Developer Kit APIs is versioned.

Glossary

A - F

Access Key

Data string that verifies the authorized access to an API in a designated UPS environment (Test or Production).

Access License Number

The data string associated with an Access Key.

Access Request

The portion of the HTTP/s request, containing the credentials, that allows the UPS system to authenticate the request as coming from an authorized requestor.

API Functionality

The range of operations performed by the specific API.

API Request-Response Framework

The underlying language, structure, protocol, and content rules associated with an HTTP/s request to or response from an API provided by UPS. UPS defines 3 frameworks for the functionality APIs: XML, SOAP, and JSON.

CIE

Acronym for Customer Integration Environment.

Common Request

The portion of the SOAP API request that exists in the Common container. It is similar across all SOAP APIs.

Customer Account

A UPS account associated with a person, business, or other entity for shipping services as defined by the agreement between the customer and UPS. This is different from the My UPS account.

Customer Integration Environment (CIE)

An environment for developers to test integration with UPS APIs.

Electronic Export Information (EEI)

Electronic Export Information (EEI) filing is generally required by the U.S. Census Bureau for U.S. exports that contain a single commodity's value exceeding US$2,500.00. All EEI information is provided to the U.S. Census Bureau and is used for export compliance and governmental reporting.

Error Code

A unique numerical value returned in an API error response used to identify the error encountered when handling the request.

Error Description

Information returned in an API error response in human readable format that indicates the possible cause of the error.

Error Response

HTTP/s response returned by the API when it is not possible to process the request.

Error Severity

Information returned in an API error response that indicates if the error is Hard or Transient.

FLT

Full truckload

Functionality Request

An HTTP/s request to one of the APIs exposed by UPS in order to submit information to or retrieve information from one of the UPS shipping applications.

LTL

Less than truckload

G - L

Hard Error

A type of error severity that is considered critical and precludes the UPS system from processing the request.

JSON API

A type of UPS API from UPS that supports the JSON data-interchange format in conjunction with a set of schema that define the structure of the API request and response. HTTP/s is used for message transport.

JSON API Request

An HTTP/s request to a UPS API that complies with the structure, rules, and constraints associated with the JSON API.

JSON API Response

An HTTP/s response from a UPS API that is returned in a format that complies with the structure, rules, and constraints associated with the JSON API.

Lane Pair

Origin and destination combination.

LTL

Acronym for Less than a Tuckload.

M – S

My UPS Account

The UPS.com account enabling access and privileges to the various areas of the UPS website including the ability to request and manage access keys and download the UPS Developer Kit packages. This is different from the *UPS Customer Account* (*UPS Account)*.

My UPS Profile

Found in the *Administration* section of the *UPS Developer Kit* area on UPS.com, the profile stores the contact information for the customer and may be associated with one or more UPS Accounts.

OASIS

A nonprofit consortium that drives the development, convergence, and adoption of open standards for the global information society.

OASIS WS-I is a member Section within the OASIS open standards consortium that advances Best Practices for Web services interoperability, for selected groups of Web services standards, across multiple platforms, operating systems and programming languages.

http://www.oasis-ws-i.org/

PLD

Package Level Detail

QVM

Quantum View Manage is a free web-based application that enables one or more users within your organization to monitor the status of all your UPS shipments from any computer connected to the Internet.

Quantum View Manage can be configured to provide you with visibility to your outbound, inbound, and/or third-party UPS shipments. Visibility includes both domestic and international shipments utilizing UPS package services as well as ground, air, and ocean freight services.

QVD

Quantum View Data is ideal for businesses that want to utilize UPS visibility information in their own software, applications, and databases. Information about your inbound, outbound, and third party package shipments is available in CSV, flat file, and XML-formatted data files for easy integration into your own systems.

ResponseStatusCode

An element returned with the value of 1, in the success response by each of the API types (XML, SOAP, JSON). Only the XML API returns the element in an error response and the value is set to 0.

ResponseStatusDescription

An element returned with the *Success with Warning* response from an XML API that holds any warning messages sent by the UPS system.

Security Request

The access request portion of a request in a JSON API. The security request contains the credentials that allow the UPS system to authenticate whether the request is coming from an authorized UPS API requestor.

ServiceAccessToken

A container in the security header portion of the SOAP API request. It holds the *Access License Number* for the request.

SOAP API

A type of functionality API from UPS that supports the SOAP messaging protocol. It is used in conjunction with an XML message to communicate with the UPS system. HTTP/s is used for message transport.

SOAP API Request

An HTTP/s request to a functionality API that complies with the structure, rules, and constraints associated with the SOAP API.

SOAP Fault

The SOAP element used to carry error and/or status information within a SOAP message. When present, it contains the UPS error code, severity, and description of the error in addition to the fault information.

Success Response

HTTP/s response from a functionality API upon successfully processing the request.

T - Z

Transient Error

A type of error severity that is due to a problem that is temporary in nature and does not originate with the request.

UPS Account

An agreement between a person, business, or other entity with UPS for shipping services. This is different from the My UPS account.

UPS Account Number

The number associated with your UPS account that uniquely identifies a customer.

UPS Developer Kit

The zip file, associated with a single API, containing all the API integration files including Developer Guides, code samples, schema, and WSDLs if applicable.

UPS Developer Kit Page

The UPS.com web page that is defined as the main area where developers will find technical information about the functionality APIs provided by UPS.

UPS Developer Resource Center

The central information hub and starting point on the UPS web site for working with the APIs.

UPSSecurity

The security header for the SOAP API request. It contains the security information block with authentication and access credentials.

UsernameToken

The portion of the SOAP security block that stores the Username and Password for API authentication.

W3C

World Wide Web Consortium. An international community of member organizations, full-time staff, and the public who work together to develop Web Standards.

http://www.w3.org

XML API

A type of functionality API from UPS that defines the format of the API request and response as valid XML documents transported in the body of an HTTP/s POST.

XML API Request

An HTTP/s request to a functionality API that complies with the structure, rules, and constraints associated with the XML API.

XML API Response

An HTTP/s response from a functionality API that is returned in a format that complies with the structure, rules, and constraints associated with the XML API.