

XML 4.0 for Wholesalers (XML 4)

User Guide

Version: 1.13

Mar, 15 2022

**Contents**

Chapter 1: About this Document 6

Purpose 6

Understanding this Document 6

What is an XML Interface 6

Document Audience 6

Text Conventions 6

Chapter 2: XML for Wholesalers 7

Purpose 7

Inventory Solution 7

Overview 7

Why RCI 7

How wholesalers access RCI inventory 7

What is XML 7

How does this connection using XML help my customers? 8

Is it safe? 8

How do I connect to RCI’s servers test my code? 8

How do I connect to RCI’s production servers? 9

How do I get detailed resort information? 9

Where do I go to obtain resort images? 10

Chapter 3: Methodology 11

Purpose 11

XML Logical Function Flows 11

How to understand the logical function flow 11

Search/Reserve requests 12

Traveler requests 13

Activity requests 14

Chapter 4: XML Requests 15

Purpose 15

Quick Reference to XML Requests 15

“addTraveler” 16

Purpose 16

Additional Information/Restrictions 16

Prerequisites 16

Expected Response 16

DTD 16

Noteworthy Elements 17

XML Example 18

“cancel” 19

Purpose 19

Additional Information/Restrictions 19

Prerequisites 19

Expected Response 19

DTD 19

Noteworthy Elements 19

XML Example 20

Confirm – “confirm” 21

Purpose 21

Additional Information/Restrictions 21

Prerequisites 21

Expected Response 21

DTD 21

Noteworthy Elements 22

XML Example 22

Confirm – “confirmHold” 24

Purpose 24

Additional Information/Restrictions 24

Prerequisites 24

Expected Response 24

DTD 24

Noteworthy Elements 24

XML Example 25

“hold” 26

Purpose 26

Additional Information/Restrictions 26

Prerequisites 26

Expected Response 26

DTD 26

Noteworthy Elements 27

XML Example 28

“inquireTraveler” 29

Purpose 29

Prerequisites 29

Expected Response 29

DTD 29

Noteworthy Elements 29

XML Example 29

“release” 31

Purpose 31

Additional Information/Restrictions 31

Prerequisites 31

Expected Response 31

DTD 31

Noteworthy Elements 31

XML Example 32

“retrieveActivity” 33

Purpose 33

Additional Information/Restrictions 33

Prerequisites 33

Expected Response 33

DTD 33

Noteworthy Elements 34

XML Example 34

Search – “priceSearch” 36

Purpose 36

Additional Information/Restrictions 36

Prerequisites 36

Expected Response 36

DTD 36

Noteworthy Elements 36

XML Example 37

Search – “quickSearch” 39

Purpose 39

Additional Information/Restrictions 39

Prerequisites 39

Expected Response 39

DTD 39

Noteworthy Elements 39

XML Example 40

Search – “resortIdSearch” 41

Purpose 41

Additional Information/Restrictions 41

Prerequisites 41

Expected Response 41

DTD 41

Noteworthy Elements 41

XML Example 42

Chapter 5: XML Responses 43

Purpose 43

Quick Reference to XML Responses 43

ActivityResponse 44

Purpose 44

Response To 44

DTD 44

Noteworthy Elements 45

XML Example 45

Error 47

Purpose 47

Response To 47

DTD 47

Element Description 47

XML Example 47

InquireTravelerResponse 48

Purpose 48

Response To 48

DTD 48

Noteworthy Elements 48

XML Example 48

ReservationResponse 50

Purpose 50

Response To 50

DTD 50

Noteworthy Elements 50

XML Example 50

SearchResponse 52

Purpose 52

Additional Information/Restrictions 52

Response To 52

DTD 52

Noteworthy Elements 53

XML Example 54

Success 56

Purpose 56

Response To 56

DTD 56

Noteworthy Elements 56

XML Example 56

Appendix A: Detailed Resort Info 57

Appendix B: Valid Countries 58

Appendix C: Valid States 62

Australia 62

Canada 62

Mexico 62

US & US Territories 63

Appendix D: Zip/Postal Codes 65

Location of zip/postal code in traveler addresses 65

Zip/Postal code formatting 66

Canada 66

USA 66

Mexico 68

# Chapter 1: About this Document

## Purpose

This document serves as the User Guide for the XML interface for wholesalers. This document discusses all of the essential topics to understand the following:

* Wholesaler’s XML interface
* RCI XML requests
* RCI XML responses

## Understanding this Document

### What is an XML Interface

The XML interface is a set of XML streams whereby RCI’s wholesalers make requests such as availability search, hold, confirm, etc. and receive responses such as resort availability, confirmations, error messages, etc.

### Document Audience

This Document is intended to be used by the following groups of people:

* Wholesaler’s technical staff
* Wholesaler’s process supervisors

### Text Conventions

This document utilizes the following text conventions to indicate certain processes, procedures and important information.

|  |  |
| --- | --- |
| **Text Convention** | **Meaning** |
| Courier New | DTD or XML code |

# Chapter 2: XML for Wholesalers

## Purpose

This chapter discusses the relationship between RCI and its wholesalers and how XML can make the partnership easy and profitable.

## Inventory Solution

### Overview

Many travel outlets, such as web-based sites, travel agents, etc. are constantly seeking new inventory to present to their clients. This makes finding alternate rental property inventory a necessity as well as a challenge. This section discusses this business need of the reseller and how RCI can help them present new inventory to their customers.

### Why RCI

RCI has an inventory of timeshare units which are available for vacation rental purposes. This inventory consists of over 3,700 resorts in more that 100 countries world-wide. Normally, these inventory databases are limited to RCI members and their families and not the general public.

Because travel outlets are always looking for new inventory, RCI decided to make this inventory available to selected wholesalers and their customers. This arrangement is a win-win situation for both RCI and the reseller.

When a wholesaler decides to provide its customers access to RCI’s extensive rental inventory, the wholesaler enters into a contract with RCI for the right to access the inventory. The wholesaler can still use its existing systems, but through the use of an XML feed, its customers can query and book travel to any of the RCI’s vacation rentals.

### How wholesalers access RCI inventory

RCI works hard to understand the needs of their wholesalers and how our inventory can escalate revenue. RCI and the wholesaler agree to the terms of the partnership and then RCI sets up that wholesaler in its databases so that transactions can take place.

Once a wholesaler is set up to access RCI’s databases, the wholesaler’s customers can make requests to view RCI’s inventory via the wholesaler’s web-site. The transactions the customer performs on the website (e.g. searches), are sent to RCI’s databases via an XML feed and RCI responds in kind.

### What is XML

XML stands for E**X**tensible **M**arkup **L**anguage.

XML is a robust method of transmitting data back and forth in real time. It is a standard format of data transmission, which describes data and allows users to define relevant tags for the data.

RCI has developed a standard XML format which is used in the search and rental of its inventory (see Chapters 4 and 5 of this document).

### How does this connection using XML help my customers?

The connection between RCI and the wholesaler happens behind the scenes so that a customer will not even know that they are actually accessing RCI’s databases. The wholesaler customers continue to use the familiar front-end application; it is what happens on the back end that makes the connection so seamless.

By using XML, a user can input requests, such as inventory search requests and reservation requests, on the front end. The wholesaler translates that request into XML and sends the XML to RCI. RCI then sends an appropriate response to the wholesaler who then converts the response from XML to whatever format the front-end uses.

### Is it safe?

RCI protects its data through a series of firewalls and authentication processes. RCI will assign a user ID and password for the XML application to a wholesaler when the wholesaler contracts with RCI to be able to access RCI’s inventory.

The wholesaler must provide RCI with all IP addresses from which production XML requests will be sent to RCI, and RCI will open the XML firewall to those IP addresses. Only specific IP addresses are allowed to access the RCI servers.

Authentication to the RCI server will be required for each transmission and will be accomplished by the wholesaler supplying the user ID and password assigned by RCI.

The wholesaler may not share the server address or the user ID or password with any other organization. The wholesaler may not use the server or authentication information for any purpose other than for XML transactions as implemented between RCI and the wholesaler.

XML Requests are sent to RCI via HTTPS.

### How do I connect to RCI’s servers test my code?

Wholesalers can use user ID “xmlweeksonly” with password “xmlweeksonly” in requests to the test server.

Test XML requests must be sent over HTTP to either:

* https://testb2ba.rci.com/variableduration/gateway
* https://testb2bb.rci.com/variableduration/gateway

IP addresses do not need to be opened on the XML firewall for requests to the test server. We will have one test server available at any given time.

A simple HTML page, available at either

* https://testb2ba.rci.com/apitest/XMLBP.html
* https://testb2bb.rci.com/apitest/XMLBP.html

allows wholesalers to test their raw XML without worrying about HTTPS connections, etc. This tool can prove invaluable when determining if there is a problem with the actual XML code, or if the problem is elsewhere.

The test environment is not guaranteed to be available at all times, but is usually available. One of the environments “testb2ba” or “testb2bb” will be made available. Every six months the test enviroments are swapped. Notifications are rarely published for test environment outages.

### How do I connect to RCI’s production servers?

The RCI business contact will assign a user ID and password for the XML application when the wholesaler is ready to move to the production environment. This user ID and password must be provided in each XML request to the production server. The wholesaler may not change the user ID or password directly. If the wholesaler requires a change of password, they should contact the RCI business contact.

XML requests must be sent over HTTPS to <https://b2b.rci.com/variableduration/gateway>.

### How do I get detailed resort information?

RCI provides detailed resort information in XML files, as well as an index file for accessing photos, via SFTP. RCI will assign a user ID and password for the FTP server to a wholesaler when the wholesaler contracts with RCI to be able to access RCI’s inventory.

The wholesaler must provide RCI with the IP address from which SFTP requests will be sent to RCI, and RCI will open the SFTP firewall to that IP address. Only specific IP addresses are allowed to access the RCI servers.

The wholesaler may not share the SFTP server address or the user ID or password with any other organization. The wholesaler may not use the SFTP server, authentication information, or data for any purpose other than providing resort detailed information to its customers.

Each wholesaler should load the XML files into a database on their server. The full content files are generally refreshed every Monday at approximately 7 a.m. EST/EDT.

More details may be found separately in the Resort Content Syndication Specification document.

### Where do I go to obtain resort images?

Resort images and media assets are avaialble to wholesalers through RCI’s hosted solution. RCI hosts all media though a Content Distribution Network or CDN. Wholesalers are encouraged to hotlink directly to the resort images. This will ensure that if images are updated, Wholesalers will present the correct resort asset.

There are 10 asset categories:

|  |  |
| --- | --- |
| ***Asset category abbreviation*** | ***Asset category description*** |
| E | External Image |
| I | Internal Image |
| O | Onsite Image |
| F | Floor plan Image |
| S | Site map Image |

RCI will generate three size versions of each resort image:

* Thumbnail
* Regular
* Large

A sequence number will be generated for each resort and is unique for each asset category.

For example, for resort ID 1469, if an external image is loaded, it will be tagged with the category “E” and three files will be generated:

* 1469E01T.jpg 🡪 Thumbnail image
* 1649E01.jpg 🡪 Regular size image
* 1659E01L 🡪 Large size image

The first four characters of the file name is the resort ID, the fifth character is the asset category type, the next two is the sequence number, and the last character describes the image size.

RCI will provide Wholesalers with an asset index file that lists all resort images and their URLs. The file is stored in our SFTP server with the following path:

/Distribution/Data-Transfers/ResortDir/RD-SYND-PROD/ResortAssets.xml

# Chapter 3: Methodology

## Purpose

This chapter discusses the methodology used when a wholesaler accesses RCI’s rental inventory. This chapter covers high level functional flows that explain the way data is transmitted from the wholesaler to RCI, and explains the key actions involved. The logical function flows pictorially display the flow of the user entered requests and the XML data that is generated in response to that request.

## XML Logical Function Flows

### How to understand the logical function flow

These flows show the logical flow from one request to another. Requests do not need to be made in immediate succession, but they do need to be made in the proper order; they can be made at any time, except as noted.

There are two symbols used in the logical function flows. Yellow ovals represent XML requests; red hexagons represent prior transactions that must have been completed before the request can be made.

### Search/Reserve requests

The hold and confirm requests must immediately follow a search, as the session information returned from the search must be passed into the hold/confirm request. All other requests can be made at any time without making another request during the same session.



### Traveler requests

The addTraveler request may be made at any time after a unit has been held or confirmed.



### Activity requests

The retrieveActivity request may be made at any time after a unit has been held, confirmed, or cancelled.



# Chapter 4: XML Requests

## Purpose

This chapter describes each XML request, including the list of possible responses, DTD, comments on each noteworthy XML element, and an XML example. It is geared towards the XML programmer responsible for creating the XML data.

## Quick Reference to XML Requests

|  |  |
| --- | --- |
| **Request Type/Operation Name** | **Purpose** |
| addTraveler | Adds or updates a traveler (guest certificate) to an existing hold or confirmation |
| cancel | Cancels a confirmed unit immediately, so that other customers may have access to it |
| confirm | Confirms (books) a unit directly after a search is performed |
| confirmHold | Confirms a unit that is currently being held |
| hold | Holds an available unit directly after a search is performed |
| inquireTraveler | Retrieves traveler information for an existing reservation |
| release | Releases a unit that is currently being held |
| retrieveActivity | Retrieves existing activity |
| Search – priceSearch | Searches for inventory by region code and price |
| Search – quickSearch | Searches for inventory by region code |
| Search – resortIdSearch | Searches for inventory by resort ID |

“addTraveler”

### Purpose

Adds or updates a traveler (guest certificate) to an existing hold or confirmation.

### Additional Information/Restrictions

* This request is required for all confirmations. It can happen after either the hold request or the confirmation request.
* See appendices B, C and D for additional information on countries, states, and zip/postal codes.
* Any elements that contain HTML entities must either provide the entity reference (e.g. &amp; for an ampersand), or else enclose the entire value in CDATA tags. See addressLine1 in the XML Example for an example.
* Lengths of the fields, in the addTraveler operation, differ between inventory types (1=Weeks and 2=Points). Please refer to the element details below.

### Prerequisites

* Held or confirmed reservation

### Expected Response

Success

### DTD

<?xml version=“1.0”?>

<!ELEMENT requestEnvelope (requestHeader, AddTravelerRequest)>

<!ELEMENT requestHeader (controlData, processData)>

<!ELEMENT controlData (clientName, userID, password)>

<!ELEMENT clientName (#PCDATA)>

<!ELEMENT userID (#PCDATA)>

<!ELEMENT password (#PCDATA)>

<!ELEMENT processData (operationName)>

<!ELEMENT operationName (#PCDATA)>

<!ELEMENT AddTravelerRequest (reservationToken, travelerInfo)>

<!ELEMENT reservationToken (inventoryType, referenceNumber)>

<!ELEMENT inventoryType (#PCDATA)>

<!ELEMENT referenceNumber (#PCDATA)>

<!ELEMENT travelerInfo (firstName, middleName?, lastName, addressline1, addressline2?, addressline3?, city, state, country, zipCode, homePhone, workPhone?, email)>

<!ELEMENT firstName (#PCDATA)>

<!ELEMENT middleName (#PCDATA)>

<!ELEMENT lastName (#PCDATA)>

<!ELEMENT addressline1 (#PCDATA)>

<!ELEMENT addressline2 (#PCDATA)>

<!ELEMENT addressline3 (#PCDATA)>

<!ELEMENT city (#PCDATA)>

<!ELEMENT state (#PCDATA)>

<!ELEMENT country (#PCDATA)>

<!ELEMENT zipCode (#PCDATA)>

<!ELEMENT homePhone (#PCDATA)>

<!ELEMENT workPhone (#PCDATA)>

<!ELEMENT email (#PCDATA)>

### Noteworthy Elements

|  |  |  |
| --- | --- | --- |
| **ELEMENT NAME** | **DATA TYPE** | **DATA RULES** |
| clientName | Text | * Required value: “XMLBusinessPartnerChannel” |
| userID | Text | * User name provided by RCI * Case sensitive |
| password | Text | * Password provided by RCI * Case sensitive |
| operationName | Text | * Required value: “addTraveler” |
| reservationToken | Group | * The reservationToken returned in the ReservationResponse from the hold or confirmation (excluding the unit number element) |
| middleName | Text | * This is actually the middle initial, not name |
| city | Text | * For inventoryType=1 (Weeks) - maximum length of 25 characters * For inventoryType=2 (Points) - maximum length of 20 characters * See Appendix D for formatting the city for countries that don’t use the zip code field. |
| state | Text | * Required for US, Canada, Mexico, and Australian travelers * Must be a valid state code from the list in Appendix C |
| country | Text | * Must be a valid country from the list in Appendix B |
| zipCode | Text | * Required for US, Canada, Mexico, and Australian travelers * Must be in a valid format, as described in Appendix D |
| homePhone | Text | * For inventoryType=1 (Weeks) - maximum length of 13 characters * For inventoryType=2 (Points) - maximum length of 12 characters |
| workPhone | Text | * For inventoryType=1 (Weeks) - maximum length of 13 characters * For inventoryType=2 (Points) - maximum length of 12 characters |
| firstName | Text | * For inventoryType=1 (Weeks) - maximum length of 10 characters * For inventoryType=2 (Points) - maximum length of 40 characters |
| lastName | Text | * For inventoryType=1 (Weeks) - maximum length of 20 characters * For inventoryType=2 (Points) - maximum length of 40 characters |
| addressline1 | Text | * For inventoryType=1 (Weeks) - maximum length of 25 characters * For inventoryType=2 (Points) - maximum length of 30 characters |
| addressline2 | Text | * For inventoryType=1 (Weeks) - maximum length of 25 characters * For inventoryType=2 (Points) - maximum length of 30 characters |
| addressline3 | Text | * For inventoryType=1 (Weeks) - maximum length of 25 characters * For inventoryType=2 (Points) - maximum length of 30 characters |
| email | Text | * Maximum length of 70 characters |

### 

### XML Example

<?xml version=“1.0”?>

<requestEnvelope>

<requestHeader>

<controlData>

<clientName>XMLBusinessPartnerChannel</clientName>

<userID>xmlweeksonly</userID>

<password>xmlweeksonly</password>

</controlData>

<processData>

<operationName>addTraveler</operationName>

</processData>

</requestHeader>

<AddTravelerRequest>

<reservationToken>

<inventoryType>1</inventoryType>

<referenceNumber>70840200100959001</referenceNumber>

</reservationToken>

<travelerInfo>

<firstName>John</firstName>

<middleName></middleName>

<lastName>Doe</lastName>

<addressline1><![CDATA[3rd Ave. & Main St.]]></addressline1>

<addressline2></addressline2>

<addressline3></addressline3>

<city>Carmel</city>

<state>IN</state>

<country>USA</country>

<zipCode>46032</zipCode>

<homePhone>317 805-8000</homePhone>

<workPhone>317 805-8001</workPhone>

<email>john.doe@yahoo.com</email>

</travelerInfo>

</AddTravelerRequest>

</requestEnvelope>

“cancel”

### Purpose

Cancels a confirmed unit, immediately allowing other customers to have access to it.

### Additional Information/Restrictions

* Most wholesalers’ contracts do not allow this functionality. Any cancellations would need to be done through the RCI business contact.
* The RCI business contact can provide any cancellation policy information that may apply.

### Prerequisites

* Confirmed reservation

### Expected Response

ReservationResponse

### DTD

<?xml version=“1.0”?>

<!ELEMENT requestEnvelope (requestHeader, CancelRequest)>

<!ELEMENT requestHeader (controlData, processData)>

<!ELEMENT controlData (clientName, userID, password)>

<!ELEMENT clientName (#PCDATA)>

<!ELEMENT userID (#PCDATA)>

<!ELEMENT password (#PCDATA)>

<!ELEMENT processData (operationName)>

<!ELEMENT operationName (#PCDATA)>

<!ELEMENT CancelRequest (reservationToken)>

<!ELEMENT reservationToken (inventoryType, referenceNumber)>

<!ELEMENT inventoryType (#PCDATA)>

<!ELEMENT referenceNumber (#PCDATA)>

### Noteworthy Elements

|  |  |  |
| --- | --- | --- |
| **ELEMENT NAME** | **DATA TYPE** | **DATA RULES** |
| clientName | Text | * Required value: “XMLBusinessPartnerChannel” |
| userID | Text | * User name provided by RCI * Case sensitive |
| password | Text | * Password provided by RCI * Case sensitive |
| operationName | Text | * Required value: “cancel” |
| reservationToken | Group | * The same as the ReservationToken returned in the ReservationResponse from the confirmation * Note that this element starts with a lower-case “r”, whereas the element in the ReservationResponse starts with an upper-case “R”. |

### XML Example

<?xml version=“1.0”?>

<requestEnvelope>

<requestHeader>

<controlData>

<clientName>XMLBusinessPartnerChannel</clientName>

<userID>xmlweeksonly</userID>

<password>xmlweeksonly</password>

</controlData>

<processData>

<operationName>cancel</operationName>

</processData>

</requestHeader>

<CancelRequest>

<reservationToken>

<inventoryType>1</inventoryType>

<referenceNumber>70840200100960001</referenceNumber>

</reservationToken>

</CancelRequest>

</requestEnvelope>

Confirm – “confirm”

### Purpose

Confirms (books) a unit directly after a search is performed.

### Additional Information/Restrictions

* The unit must have been available in the immediately preceding search results, using the same session ID.
* After a confirmation is completed, the search results will become invalid. A new search, yielding a new session ID, would need to be done in order to hold or confirm an additional unit.
* The amount in the “value” element should exactly match the amount returned in the “unitCost” element in the preceeding SearchResponse.
* Any elements that contain HTML entities must either provide the entity reference (e.g. &amp; for an ampersand), or else enclose the entire value in CDATA tags. See resortName in the XML Example for an example.

### Prerequisites

* Active search results

### Expected Response

ReservationResponse

### DTD

<?xml version=“1.0”?>

<!ELEMENT requestEnvelope (requestHeader, ReservationRequest)>

<!ELEMENT requestHeader (controlData, processData)>

<!ELEMENT controlData (clientName, userID, password)>

<!ELEMENT clientName (#PCDATA)>

<!ELEMENT userID (#PCDATA)>

<!ELEMENT password (#PCDATA)>

<!ELEMENT processData (operationName)>

<!ELEMENT operationName (#PCDATA)>

<!ELEMENT ReservationRequest (sessionToken?, accommodationItem)>

<!ELEMENT sessionToken (sessionId)>

<!ELEMENT sessionId (#PCDATA)>

<!ELEMENT accommodationItem (resortId, resortName, nbrOfBedrooms, nbrOfBathrooms?, kitchenType?, privacyOccupancy?, maxOccupancy?, bookingInfo)>

<!ELEMENT resortId (#PCDATA)>

<!ELEMENT resortName (#PCDATA)>

<!ELEMENT nbrOfBedrooms (#PCDATA)>

<!ELEMENT nbrOfBathrooms (#PCDATA)>

<!ELEMENT kitchenType (#PCDATA)>

<!ELEMENT privacyOccupancy (#PCDATA)>

<!ELEMENT maxOccupancy (#PCDATA)>

<!ELEMENT bookingInfo (systemId, checkInDate, checkOutDate, charge)>

<!ELEMENT systemId (inventoryType, inventoryId)>

<!ELEMENT inventoryType (#PCDATA)>

<!ELEMENT inventoryId (#PCDATA)>

<!ELEMENT checkInDate (#PCDATA)>

<!ELEMENT checkOutDate (#PCDATA)>

<!ELEMENT charge (value, currency)>

<!ELEMENT value (#PCDATA)>

<!ELEMENT currency (#PCDATA)>

### Noteworthy Elements

|  |  |  |
| --- | --- | --- |
| **ELEMENT NAME** | **DATA TYPE** | **DATA RULES** |
| clientName | Text | * Required value: “XMLBusinessPartnerChannel” |
| userID | Text | * User name provided by RCI * Case sensitive |
| password | Text | * Password provided by RCI * Case sensitive |
| operationName | Text | * Required value: “confirm” |
| sessionToken | Group | * This is the value returned in the SearchResponse * In some cases, the SearchResponse’s sessionToken will be empty. In that case, the sessionToken element does not need to be provided in this request. |
| accommodationItem | Group | * This is the value returned in the SearchResponse; all required data in this group must be identical to that returned in the SearchResponse. * Optional elements in this group need not be included. * Note that maxOccupancy and privacyOccupancy are not in the same order in the hold and confirm requests as they are in the SearchResponse. * Note that the SearchResponse may include multiple “accommodationItem” elements. Only one “accommodationItem” element may be included in this request. |
| bookingInfo | Group | * Note that the “availabilityInfo” element in the SearchResponse is called “bookingInfo” in this confirm request. * Note that the SearchResponse may include multiple “availabilityInfo” elements. Only one “availabilityInfo”/”bookingInfo” element may be included in this request. |
| charge | Group | * Note that the “unitCost” element in the SearchResponse must match the same element called “charge” in this confirm request. * As of this release, the value must be “USD”. As of the date this document was written, there are no plans to provide additional currencies. |

### XML Example

<?xml version=“1.0”?>

<requestEnvelope>

<requestHeader>

<controlData>

<clientName>XMLBusinessPartnerChannel</clientName>

<userID>xmlweeksonly</userID>

<password>xmlweeksonly</password>

</controlData>

<processData>

<operationName>confirm</operationName>

</processData>

</requestHeader>

<ReservationRequest>

<sessionToken>

<sessionId>093216580</sessionId>

</sessionToken>

<accommodationItem>

<resortId>AA99</resortId>

<resortName><![CDATA[AA99-ALPHA&BETA RESORT TEST]]></resortName>

<nbrOfBedrooms>3</nbrOfBedrooms>

<bookingInfo>

<systemId>

<inventoryType>1</inventoryType>

<inventoryId>59Lm6yhPndq30a7fDEseVFB4C5WkN+Wgvz/XmWm6+3C3tgDBlla1jxRm9H1VnyAQ</inventoryId>

</systemId>

<checkInDate>10/07/2006</checkInDate>

<checkOutDate>10/14/2006</checkOutDate>

<charge>

<value>249.99</value>

<currency>USD</currency>

</charge>

</bookingInfo>

</accommodationItem>

</ReservationRequest>

</requestEnvelope>

Confirm – “confirmHold”

### Purpose

Confirms (books) a unit that is currently being held. See the “hold” request for more information on held units.

### Additional Information/Restrictions

* If a held unit is not confirmed by approximately 11pm EST/EDT the following night, it will automatically release and be available for other customers.

### Prerequisites

* Existing hold

### Expected Response

ReservationResponse

### DTD

<?xml version=“1.0”?>

<!ELEMENT requestEnvelope (requestHeader, ConfirmRequest)>

<!ELEMENT requestHeader (controlData, processData)>

<!ELEMENT controlData (clientName, userID, password)>

<!ELEMENT clientName (#PCDATA)>

<!ELEMENT userID (#PCDATA)>

<!ELEMENT password (#PCDATA)>

<!ELEMENT processData (operationName)>

<!ELEMENT operationName (#PCDATA)>

<!ELEMENT ConfirmRequest (reservationToken)>

<!ELEMENT reservationToken (inventoryType, referenceNumber)>

<!ELEMENT inventoryType (#PCDATA)>

<!ELEMENT referenceNumber (#PCDATA)>

### Noteworthy Elements

|  |  |  |
| --- | --- | --- |
| **ELEMENT NAME** | **DATA TYPE** | **DATA RULES** |
| clientName | Text | * Required value: “XMLBusinessPartnerChannel” |
| userID | Text | * User name provided by RCI * Case sensitive |
| password | Text | * Password provided by RCI * Case sensitive |
| operationName | Text | * Required value: “confirmHold” |
| reservationToken | Group | * The same as the ReservationToken returned in the ReservationResponse from the hold * Note that this element starts with a lower-case “r”, whereas the element in the ReservationResponse starts with an upper-case “R”. |

### XML Example

<?xml version=“1.0”?>

<requestEnvelope>

<requestHeader>

<controlData>

<clientName>XMLBusinessPartnerChannel</clientName>

<userID>xmlweeksonly</userID>

<password>xmlweeksonly</password>

</controlData>

<processData>

<operationName>confirmHold</operationName>

</processData>

</requestHeader>

<ConfirmRequest>

<reservationToken>

<inventoryType>1</inventoryType>

<referenceNumber>70840200100958001</referenceNumber>

</reservationToken>

</ConfirmRequest>

</requestEnvelope>

“hold”

### Purpose

Locks a unit from being held or confirmed by another customer, allowing time for the current customer to decide whether they want to confirm (book) it.

### Additional Information/Restrictions

* If a held unit is not confirmed by approximately 11pm EST/EDT the following night, it will automatically release and be available for other customers.
* Only units of inventory type 1 (from the Weeks system) are available to be held.
* This unit must have been available in the immediately preceding search results, using the same session ID.
* The amount in the “value” element should exactly match the amount returned in the “unitCost” element in the preceeding SearchResponse.
* After a hold is completed, the search results will become invalid. A new search, yielding a new session ID, would need to be done in order to hold or confirm an additional unit.
* Any elements that contain HTML entities must either provide the entity reference (e.g. &amp; for an ampersand), or else enclose the entire value in CDATA tags. See resortName in the XML Example for an example.

### Prerequisites

* Active search results

### Expected Response

ReservationResponse

### DTD

<?xml version=“1.0”?>

<!ELEMENT requestEnvelope (requestHeader, ReservationRequest)>

<!ELEMENT requestHeader (controlData, processData)>

<!ELEMENT controlData (clientName, userID, password)>

<!ELEMENT clientName (#PCDATA)>

<!ELEMENT userID (#PCDATA)>

<!ELEMENT password (#PCDATA)>

<!ELEMENT processData (operationName)>

<!ELEMENT operationName (#PCDATA)>

<!ELEMENT ReservationRequest (sessionToken?, accommodationItem)>

<!ELEMENT sessionToken (sessionId)>

<!ELEMENT sessionId (#PCDATA)>

<!ELEMENT accommodationItem (resortId, resortName, nbrOfBedrooms, nbrOfBathrooms?, kitchenType?, privacyOccupancy?, maxOccupancy?, bookingInfo)>

<!ELEMENT resortId (#PCDATA)>

<!ELEMENT resortName (#PCDATA)>

<!ELEMENT nbrOfBedrooms (#PCDATA)>

<!ELEMENT nbrOfBathrooms (#PCDATA)>

<!ELEMENT kitchenType (#PCDATA)>

<!ELEMENT privacyOccupancy (#PCDATA)>

<!ELEMENT maxOccupancy (#PCDATA)>

<!ELEMENT bookingInfo (systemId, checkInDate, checkOutDate, charge)>

<!ELEMENT systemId (inventoryType, inventoryId+)>

<!ELEMENT inventoryType (#PCDATA)>

<!ELEMENT inventoryId (#PCDATA)>

<!ELEMENT checkInDate (#PCDATA)>

<!ELEMENT checkOutDate (#PCDATA)>

<!ELEMENT charge (value, currency)>

<!ELEMENT value (#PCDATA)>

<!ELEMENT currency (#PCDATA)>

### Noteworthy Elements

|  |  |  |
| --- | --- | --- |
| **ELEMENT NAME** | **DATA TYPE** | **DATA RULES** |
| clientName | Text | * Required value: “XMLBusinessPartnerChannel” |
| userID | Text | * User name provided by RCI * Case sensitive |
| password | Text | * Password provided by RCI * Case sensitive |
| operationName | Text | * Required value: “hold” |
| sessionToken | Group | * This is the value returned in the SearchResponse * In some cases, the SearchResponse’s sessionToken will be empty. In that case, the sessionToken element does not need to be provided in this request. |
| accommodationItem | Group | * This is the value returned in the SearchResponse; all required data in this group must be identical to that returned in the SearchResponse. * Optional elements in this group need not be included. * Note that maxOccupancy and privacyOccupancy are not in the same order in the hold and confirm requests as they are in the SearchResponse. * Note that the SearchResponse may include multiple “accommodationItem” elements. Only one “accommodationItem” element may be included in this request. |
| bookingInfo | Group | * Note that the “availabilityInfo” element in the SearchResponse is called “bookingInfo” in this confirm request. * Note that the SearchResponse may include multiple “availabilityInfo” elements. Only one “availabilityInfo”/”bookingInfo” element may be included in this request. |
| charge | Group | * Note that the “unitCost” element in the SearchResponse must match the same element called “charge” in this confirm request. * As of this release, the value must be “USD”. As of the date this document was written, there are no plans to provide additional currencies. |

### XML Example

<?xml version=“1.0”?>

<requestEnvelope>

<requestHeader>

<controlData>

<clientName>XMLBusinessPartnerChannel</clientName>

<userID>xmlweeksonly</userID>

<password>xmlweeksonly</password>

</controlData>

<processData>

<operationName>hold</operationName>

</processData>

</requestHeader>

<ReservationRequest>

<sessionToken>

<sessionId>093216585</sessionId>

</sessionToken>

<accommodationItem>

<resortId>AA99</resortId>

<resortName><![CDATA[AA99-ALPHA&BETA RESORT TEST]]></resortName>

<nbrOfBedrooms>3</nbrOfBedrooms>

<bookingInfo>

<systemId>

<inventoryType>1</inventoryType>

<inventoryId>59Lm6yhPndq30a7fDEseVFB4C5WkN+Wgkl27ue8LrSuq+MhLbgilCxRm9H1VnyAQ</inventoryId>

</systemId>

<checkInDate>10/07/2006</checkInDate>

<checkOutDate>10/14/2006</checkOutDate>

<charge>

<value>249.99</value>

<currency>USD</currency>

</charge>

</bookingInfo>

</accommodationItem>

</ReservationRequest>

</requestEnvelope>

“inquireTraveler”

### Purpose

Returns the traveler (guest certificate) information for an existing confirmation.

### Prerequisites

* Existing hold or confirmation

### Expected Response

InquireTravelerResponse

### DTD

<?xml version=“1.0”?>

<!ELEMENT requestEnvelope (requestHeader, InquireTravelerRequest)>

<!ELEMENT requestHeader (controlData, processData)>

<!ELEMENT controlData (clientName, userID, password)>

<!ELEMENT clientName (#PCDATA)>

<!ELEMENT userID (#PCDATA)>

<!ELEMENT password (#PCDATA)>

<!ELEMENT processData (operationName)>

<!ELEMENT operationName (#PCDATA)>

<!ELEMENT InquireTravelerRequest (reservationToken)>

<!ELEMENT reservationToken (inventoryType, referenceNumber)>

<!ELEMENT inventoryType (#PCDATA)>

<!ELEMENT referenceNumber (#PCDATA)>

### Noteworthy Elements

|  |  |  |
| --- | --- | --- |
| **ELEMENT NAME** | **DATA TYPE** | **DATA RULES** |
| clientName | Text | * Required value: “XMLBusinessPartnerChannel” |
| userID | Text | * User name provided by RCI * Case sensitive |
| password | Text | * Password provided by RCI * Case sensitive |
| operationName | Text | * Required value: “inquireTraveler” |
| reservationToken | Group | * The same as the ReservationToken returned in the ReservationResponse from the hold or confirmation * Note that this element starts with a lower-case “r”, whereas the element in the ReservationResponse starts with an upper-case “R”. |

### XML Example

<?xml version=“1.0”?>

<requestEnvelope>

<requestHeader>

<controlData>

<clientName>XMLBusinessPartnerChannel</clientName>

<userID>xmlweeksonly</userID>

<password>xmlweeksonly</password>

</controlData>

<processData>

<operationName>inquireTraveler</operationName>

</processData>

</requestHeader>

<InquireTravelerRequest>

<reservationToken>

<inventoryType>1</inventoryType>

<referenceNumber>70840200100959001</referenceNumber>

</reservationToken>

</InquireTravelerRequest>

</requestEnvelope>

“release”

### Purpose

Releases a unit that was previously held using the “hold” request, such that other customers may have access to it. See the “hold” request for more information on held units.

### Additional Information/Restrictions

* If a held unit is not confirmed by approximately 11pm EST/EDT the following night, it will automatically release and be available for other customers. This request causes the unit to be released before the system automatically does it.

### Prerequisites

* Existing hold

### Expected Response

ReservationResponse

### DTD

<?xml version=“1.0”?>

<!ELEMENT requestEnvelope (requestHeader, ReleaseRequest)>

<!ELEMENT requestHeader (controlData, processData)>

<!ELEMENT controlData (clientName, userID, password)>

<!ELEMENT clientName (#PCDATA)>

<!ELEMENT userID (#PCDATA)>

<!ELEMENT password (#PCDATA)>

<!ELEMENT processData (operationName)>

<!ELEMENT operationName (#PCDATA)>

<!ELEMENT ReleaseRequest (reservationToken)>

<!ELEMENT reservationToken (inventoryType, referenceNumber)>

<!ELEMENT inventoryType (#PCDATA)>

<!ELEMENT referenceNumber (#PCDATA)>

### Noteworthy Elements

|  |  |  |
| --- | --- | --- |
| **ELEMENT NAME** | **DATA TYPE** | **DATA RULES** |
| clientName | Text | * Required value: “XMLBusinessPartnerChannel” |
| userID | Text | * User name provided by RCI * Case sensitive |
| password | Text | * Password provided by RCI * Case sensitive |
| operationName | Text | * Required value: “release” |
| reservationToken | Group | * The same as the ReservationToken returned in the ReservationResponse from the hold * Note that this element starts with a lower-case “r”, whereas the element in the ReservationResponse starts with an upper-case “R”. |

### XML Example

<?xml version=“1.0”?>

<requestEnvelope>

<requestHeader>

<controlData>

<clientName>XMLBusinessPartnerChannel</clientName>

<userID>xmlweeksonly</userID>

<password>xmlweeksonly</password>

</controlData>

<processData>

<operationName>release</operationName>

</processData>

</requestHeader>

<ReleaseRequest>

<reservationToken>

<inventoryType>1</inventoryType>

<referenceNumber>70840200100958001</referenceNumber>

</reservationToken>

</ReleaseRequest>

</requestEnvelope>

“retrieveActivity”

### Purpose

Retrieves existing holds, confirmations, and cancellations.

### Additional Information/Restrictions

* There is an option to retrieve transactions within a specific date range. If the date range is not specified in the request, the system will try to bring back all the transactions (Hold, Confirm, Cancel), that were performed using the specific account. In this scenario, where all the transactions are being retrieved, there is a possibility for the request to timeout. Hence, it is recommended to use a short date range (Between 1 - 3 months) to retrieve the transactions.

### Prerequisites

* None

### Expected Response

ActivityResponse

### DTD

<?xml version=“1.0”?>

<!ELEMENT requestEnvelope (requestHeader, ActivityRequest)>

<!ELEMENT requestHeader (controlData, processData)>

<!ELEMENT controlData (clientName, userID, password)>

<!ELEMENT clientName (#PCDATA)>

<!ELEMENT userID (#PCDATA)>

<!ELEMENT password (#PCDATA)>

<!ELEMENT processData (operationName)>

<!ELEMENT operationName (#PCDATA)>

<!ELEMENT ActivityRequest (onHoldActivity?, confirmedActivity?, cancelledActivity?,

startDate?,

endDate?,

dateFlag?)>

<!ELEMENT onHoldActivity (EMPTY)>

<!ELEMENT confirmedActivity (EMPTY)>

<!ELEMENT cancelledActivity (EMPTY)>

<!ATTLIST onHoldActivity fetch (true | false) “true”>

<!ATTLIST confirmedActivity fetch (true | false) “true”>

<!ATTLIST cancelledActivity fetch (true | false) “true”>

<!ELEMENT startDate (#PCDATA)>

<!ELEMENT endDate (#PCDATA)>

<!ELEMENT dateFlag (#PCDATA)>

### Noteworthy Elements

|  |  |  |
| --- | --- | --- |
| **ELEMENT NAME** | **DATA TYPE** | **DATA RULES** |
| clientName | Text | * Required value: “XMLBusinessPartnerChannel” |
| userID | Text | * User name provided by RCI * Case sensitive |
| password | Text | * Password provided by RCI * Case sensitive |
| operationName | Text | * Required value: “retrieveActivity” |
| reservationToken | Group | * The reservationToken returned in the ReservationResponse from the confirmation |
| onHoldActivity | Empty | * If omitted, does not return held activity. * If present, the fetch attribute determines whether held activity will be retrieved. * If the fetch attribute is omitted, held activity is returned. |
| confirmedActivity | Empty | * If omitted, does not return confirmed activity. * If present, the fetch attribute determines whether confirmed activity will be retrieved * If the fetch attribute is omitted, confirmed activity is returned. |
| cancelledActivity | Empty | * If omitted, does not return cancelled activity. * If present, the fetch attribute determines whether cancelled activity will be retrieved * If the fetch attribute is omitted, cancelled activity is returned. |
| startDate | Date | * Valid date * mm/dd/yyyy format * This element should be combined with “endDate” and “dateFlag” |
| endDate | Date | * Valid date * mm/dd/yyyy format * This element should be combined with “startDate” and “dateFlag”. |
| dateFlag | Text | * Required value: “checkinDate” or “transactionDate” * This element should be combined with “startDate” and “endDate” * If the value of this element is ‘transactionDate’, it specifies to search for transactions that were commited within the start and end date * If the value of this element is ‘checkinDate’, it specifis to search for transactions, whose check-in dates are within the start and end date. |

### XML Example

<?xml version=“1.0”?>

<requestEnvelope>

<requestHeader>

<controlData>

<clientName>XMLBusinessPartnerChannel</clientName>

<userID>xmlweeksonly</userID>

<password>xmlweeksonly</password>

</controlData>

<processData>

<operationName>retrieveActivity</operationName>

</processData>

</requestHeader>

<ActivityRequest>

<onHoldActivity fetch="true"/>

<confirmedActivity fetch="false"/>

<cancelledActivity fetch="false"/>

<startDate>03/11/2008</startDate>

<endDate>08/22/2008</endDate>

<dateFlag>checkinDate</dateFlag>

<!--dateFlag>transactionDate</dateFlag-->

</ActivityRequest>

</requestEnvelope>

Search – “priceSearch”

### Purpose

Searches resort availability in a specified region within a specified price range.

### Additional Information/Restrictions

* After a hold or comfirmation is completed, the search results will become invalid. A new search, yielding a new session ID, would need to be done in order to hold or confirm an additional unit.

### Prerequisites

* None

### Expected Response

SearchResponse

### DTD

<?xml version=“1.0”?>

<!ELEMENT requestEnvelope (requestHeader, SearchRequest)>

<!ELEMENT requestHeader (controlData, processData)>

<!ELEMENT controlData (clientName, userID, password)>

<!ELEMENT clientName (#PCDATA)>

<!ELEMENT userID (#PCDATA)>

<!ELEMENT password (#PCDATA)>

<!ELEMENT processData (operationName)>

<!ELEMENT operationName (#PCDATA)>

<!ELEMENT SearchRequest (startDate, endDate, lengthOfStay, minUnitSize?, regionCode, minPrice, maxPrice, currency)>

<!ELEMENT startDate (#PCDATA)>

<!ELEMENT endDate (#PCDATA)>

<!ELEMENT lengthOfStay (#PCDATA)>

<!ELEMENT minUnitSize (#PCDATA)>

<!ELEMENT regionCode (#PCDATA)>

<!ELEMENT minPrice (#PCDATA)>

<!ELEMENT maxPrice (#PCDATA)>

<!ELEMENT currency (#PCDATA)>

### Noteworthy Elements

|  |  |  |
| --- | --- | --- |
| **ELEMENT NAME** | **DATA TYPE** | **DATA RULES** |
| clientName | Text | * Required value: “XMLBusinessPartnerChannel” |
| userID | Text | * User name provided by RCI * Case sensitive |
| password | Text | * Password provided by RCI * Case sensitive |
| operationName | Text | * Required value: “priceSearch” |
| startDate | Date | * Valid date * At least 2 days in the future * mm/dd/yyyy format * This is not necessarily the desired check-in date. This is the beginning of the check-in date range. |
| endDate | Date | * Valid date * Greater than or equal to startDate * No more than 10 weeks after startDate * mm/dd/yyyy format * This is not the desired check-out date. This is the end of the check-in date range. |
| lengthOfStay | Integer | * Number of nights the customer wants to stay. * For Weeks system inventory, this must be 7. * For Points system (nightly) inventory, this can be anything up to and including 6. * The wholesaler contract will specify whether the wholesaler has access to the non-Weeks inventory. |
| minUnitSize | Text | * Mandatory (default may be set to H) * The minimum size of unit for which the customer is searching. Must be one of:   + H – Hotel   + S – Studio   + 1 – 1 Bedroom   + 2 – 2 Bedroom   + 3 – 3 Bedroom   + 4 – 4 Bedroom * Note that the values for hotel and studio units are letters in the search request, but numbers in the SearchResponse |
| regionCode | Text | * Any valid 4-character RCI region code from the list provided by the RCI business contact |
| minPrice | Decimal | * The minimum amount the customer is willing to spend for the available inventory. In most situations this should be 0.00. * Include decimal point and 2 decimal places. |
| maxPrice | Decimal | * The maximum amount the customer is willing to spend for the available inventory. * Include decimal point and 2 decimal places. |
| currency | Text | * Currency in which the customer is searching. * As of this release, the value must be “USD”. As of the date this document was written, there are no plans to provide additional currencies. |

### XML Example

<?xml version=“1.0”?>

<requestEnvelope>

<requestHeader>

<controlData>

<clientName>XMLBusinessPartnerChannel</clientName>

<userID>xmlweeksonly</userID>

<password>xmlweeksonly</password>

</controlData>

<processData>

<operationName>priceSearch</operationName>

</processData>

</requestHeader>

<SearchRequest>

<startDate>10/01/2006</startDate>

<endDate>11/01/2006</endDate>

<lengthOfStay>7</lengthOfStay>

<minUnitSize>H</minUnitSize>

<regionCode>FLOR</regionCode>

<minPrice>0.00</minPrice>

<maxPrice>200.00</maxPrice>

<currency>USD</currency>

</SearchRequest>

</requestEnvelope>

Search – “quickSearch”

### Purpose

Searches resort availability in a region.

### Additional Information/Restrictions

* After a hold or comfirmation is completed, the search results will become invalid. A new search, yielding a new session ID, would need to be done in order to hold or confirm an additional unit.

### Prerequisites

* None

### Expected Response

SearchResponse

### DTD

<?xml version=“1.0”?>

<!ELEMENT requestEnvelope (requestHeader, SearchRequest)>

<!ELEMENT requestHeader (controlData, processData)>

<!ELEMENT controlData (clientName, userID, password)>

<!ELEMENT clientName (#PCDATA)>

<!ELEMENT userID (#PCDATA)>

<!ELEMENT password (#PCDATA)>

<!ELEMENT processData (operationName)>

<!ELEMENT operationName (#PCDATA)>

<!ELEMENT SearchRequest (startDate, endDate, lengthOfStay, minUnitSize?, regionCode)>

<!ELEMENT startDate (#PCDATA)>

<!ELEMENT endDate (#PCDATA)>

<!ELEMENT lengthOfStay (#PCDATA)>

<!ELEMENT minUnitSize (#PCDATA)>

<!ELEMENT regionCode (#PCDATA)>

### Noteworthy Elements

|  |  |  |
| --- | --- | --- |
| **ELEMENT NAME** | **DATA TYPE** | **DATA RULES** |
| clientName | Text | * Required value: “XMLBusinessPartnerChannel” |
| userID | Text | * User name provided by RCI * Case sensitive |
| password | Text | * Password provided by RCI * Case sensitive |
| operationName | Text | * Required value: “quickSearch” |
| startDate | Date | * Valid date * At least 2 days in the future * mm/dd/yyyy format * This is not necessarily the desired check-in date. This is the beginning of the check-in date range. |
| endDate | Date | * Valid date * Greater than or equal to startDate * No more than 10 weeks after startDate * mm/dd/yyyy format * This is not the desired check-out date. This is the end of the check-in date range. |
| lengthOfStay | Integer | * Number of nights the customer wants to stay. * For Weeks system inventory, this must be 7. * For Points system (nightly) inventory, this can be anything up to and including 6. * The wholesaler contract will specify whether the wholesaler has access to the non-Weeks inventory. |
| minUnitSize | Text | * Mandatory (default may be set to H) * The minimum size of unit for which the customer is searching. Must be one of:   + H – Hotel   + S – Studio   + 1 – 1 Bedroom   + 2 – 2 Bedroom   + 3 – 3 Bedroom   + 4 – 4 Bedroom * Note that the values for hotel and studio units are letters in the search request, but numbers in the SearchResponse |
| regionCode | Text | * Any valid 4-character RCI region code from the list provided by the RCI business contact |

### XML Example

<?xml version=“1.0”?>

<requestEnvelope>

<requestHeader>

<controlData>

<clientName>XMLBusinessPartnerChannel</clientName>

<userID>xmlweeksonly</userID>

<password>xmlweeksonly</password>

</controlData>

<processData>

<operationName>quickSearch</operationName>

</processData>

</requestHeader>

<SearchRequest>

<startDate>10/01/2006</startDate>

<endDate>11/01/2006</endDate>

<lengthOfStay>7</lengthOfStay>

<minUnitSize>H</minUnitSize>

<regionCode>FLOR</regionCode>

</SearchRequest>

</requestEnvelope>

Search – “resortIdSearch”

### Purpose

Searches availability of a specific resort(s).

### Additional Information/Restrictions

* After a hold or comfirmation is completed, the search results will become invalid. A new search, yielding a new session ID, would need to be done in order to hold or confirm an additional unit.

### Prerequisites

* None

### Expected Response

SearchResponse

### DTD

<?xml version=“1.0”?>

<!ELEMENT requestEnvelope (requestHeader, SearchRequest)>

<!ELEMENT requestHeader (controlData, processData)>

<!ELEMENT controlData (clientName, userID, password)>

<!ELEMENT clientName (#PCDATA)>

<!ELEMENT userID (#PCDATA)>

<!ELEMENT password (#PCDATA)>

<!ELEMENT processData (operationName)>

<!ELEMENT operationName (#PCDATA)>

<!ELEMENT SearchRequest (startDate, endDate, lengthOfStay, minUnitSize?, resortId+)>

<!ELEMENT startDate (#PCDATA)>

<!ELEMENT endDate (#PCDATA)>

<!ELEMENT lengthOfStay (#PCDATA)>

<!ELEMENT minUnitSize (#PCDATA)>

<!ELEMENT resortId (#PCDATA)>

### Noteworthy Elements

|  |  |  |
| --- | --- | --- |
| **ELEMENT NAME** | **DATA TYPE** | **DATA RULES** |
| clientName | Text | * Required value: “XMLBusinessPartnerChannel” |
| userID | Text | * User name provided by RCI * Case sensitive |
| password | Text | * Password provided by RCI * Case sensitive |
| operationName | Text | * Required value: “resortIdSearch” |
| startDate | Date | * Valid date * At least 2 days in the future * mm/dd/yyyy format * This is not necessarily the desired check-in date. This is the beginning of the check-in date range. |
| endDate | Date | * Valid date * Greater than or equal to startDate * No more than 10 weeks after startDate * mm/dd/yyyy format * This is not the desired check-out date. This is the end of the check-in date range. |
| lengthOfStay | Integer | * Number of nights the customer wants to stay. * For Weeks system inventory, this must be 7. * For Points system (nightly) inventory, this can be anything up to and including 6. * The wholesaler contract will specify whether the wholesaler has access to the non-Weeks inventory. |
| minUnitSize | Text | * Mandatory (default may be set to H) * The minimum size of unit for which the customer is searching. Must be one of:   + H – Hotel   + S – Studio   + 1 – 1 Bedroom   + 2 – 2 Bedroom   + 3 – 3 Bedroom   + 4 – 4 Bedroom * Note that the values for hotel and studio units are letters in the search request, but numbers in the SearchResponse |
| resortId | Text | * Any valid alphanumeric 4-character RCI resort ID * Most resort IDs are numeric (4-bytes with leading zeroes where appropriate), but alphanumeric resort IDs are also being used * Each resort ID must be listed in a separate element * There may be a maximum of 48 resorts listed |

### XML Example

<?xml version=“1.0”?>

<requestEnvelope>

<requestHeader>

<controlData>

<clientName>XMLBusinessPartnerChannel</clientName>

<userID>xmlweeksonly</userID>

<password>xmlweeksonly</password>

</controlData>

<processData>

<operationName>resortIdSearch</operationName>

</processData>

</requestHeader>

<SearchRequest>

<startDate>10/01/2006</startDate>

<endDate>11/01/2006</endDate>

<lengthOfStay>7</lengthOfStay>

<minUnitSize>H</minUnitSize>

<resortId>1788</resortId>

<resortId>AA99</resortId>

</SearchRequest>

</requestEnvelope>

# Chapter 5: XML Responses

## Purpose

This chapter describes each XML response, including the list of requests that generate them, DTD, descriptions of each XML element, and a sample XML response. It is geared towards the XML programmer responsible for processing the XML response data.

## Quick Reference to XML Responses

|  |  |
| --- | --- |
| **Response Name** | **Purpose** |
| ActivityResponse | Returns information about existing reservations |
| Error | Returns error information for a failed request |
| InquireTravelerResponse | Returns traveler information for a specific reservation |
| ReservationResponse | Returns reservation details for holds, confirmations, cancellations, and releases |
| SearchResponse | Returns detailed information of resort availability |
| Success | Acknowledgement of a successful traveler add request |

ActivityResponse

### Purpose

Returns information about existing reservations

### Response To

retrieveActivity

### DTD

<?xml version=“1.0”?>

<!ELEMENT responseEnvelope (responseHeader, ActivityResponse)>

<!ELEMENT responseHeader (statusData)>

<!ELEMENT statusData (completionCode, completionMessage)>

<!ELEMENT completionCode (#PCDATA)>

<!ELEMENT completionMessage (#PCDATA)>

<!ELEMENT ActivityResponse (onHoldActivity, confirmedActivity, cancelledActivity)>

<!ELEMENT onHoldActivity (activityItem\*)>

<!ELEMENT confirmedActivity (activityItem\*)>

<!ELEMENT cancelledActivity (activityItem\*)>

<!ELEMENT activityItem (reservationToken, status, accommodationItem, travelerInfo?)>

<!ELEMENT reservationToken (inventoryType, referenceNumber, unitNumber?)>

<!ELEMENT inventoryType (#PCDATA)>

<!ELEMENT referenceNumber (#PCDATA)>

<!ELEMENT unitNumber (#PCDATA)>

<!ELEMENT status (#PCDATA)>

<!ELEMENT accommodationItem (resortId, resortName, nbrOfBedrooms, kitchenType, maxOccupancy, bookingInfo)>

<!ELEMENT resortId (#PCDATA)>

<!ELEMENT resortName (#PCDATA)>

<!ELEMENT nbrOfBedrooms (#PCDATA)>

<!ELEMENT kitchenType (#PCDATA)>

<!ELEMENT maxOccupancy (#PCDATA)>

<!ELEMENT bookingInfo (checkInDate, checkOutDate, charge)>

<!ELEMENT checkInDate (#PCDATA)>

<!ELEMENT checkOutDate (#PCDATA)>

<!ELEMENT charge (value, currency)>

<!ELEMENT value (#PCDATA)>

<!ELEMENT currency (#PCDATA)>

<!ELEMENT travelerInfo (firstName, middleName?, lastName, addressline1, addressline2?, addressline3?, city, state, country, zipCode, homePhone, workPhone?, email)>

<!ELEMENT firstName (#PCDATA)>

<!ELEMENT middleName (#PCDATA)>

<!ELEMENT lastName (#PCDATA)>

<!ELEMENT addressline1 (#PCDATA)>

<!ELEMENT addressline2 (#PCDATA)>

<!ELEMENT addressline3 (#PCDATA)>

<!ELEMENT city (#PCDATA)>

<!ELEMENT state (#PCDATA)>

<!ELEMENT country (#PCDATA)>

<!ELEMENT zipCode (#PCDATA)>

<!ELEMENT homePhone (#PCDATA)>

<!ELEMENT workPhone (#PCDATA)>

<!ELEMENT email (#PCDATA)>

### Noteworthy Elements

|  |  |  |
| --- | --- | --- |
| **ELEMENT NAME** | **DATA TYPE** | **DATA RULES** |
| completionCode | Number | * 0 for success * See response code list for other codes |
| unitNumber | Text | * The unit number is not necessarily known at time of booking, and may be assigned at check-in. |
| status | Text | * Indicates the status of the reservation H – Held C – Confirmed X – Cancelled |
| middleName | Text | * This is the middle initial, not name |

### XML Example

<?xml version="1.0" ?>

<responseEnvelope>

<responseHeader>

<statusData>

<completionCode>0</completionCode>

<completionMessage>Request Completed Successfully.</completionMessage>

</statusData>

</responseHeader>

<ActivityResponse>

<onHoldActivity>

<activityItem>

<reservationToken>

<inventoryType>1</inventoryType>

<referenceNumber>70840200100019001</referenceNumber>

</reservationToken>

<status>H</status>

<accommodationItem>

<resortId>2750</resortId>

<resortName>Cypress Pointe Resort</resortName>

<nbrOfBedrooms>2</nbrOfBedrooms>

<kitchenType>Full Kitchen</kitchenType>

<maxOccupancy>6</maxOccupancy>

<bookingInfo>

<checkInDate>08/04/2006</checkInDate>

<checkOutDate>08/11/2006</checkOutDate>

<charge>

<value>330.0</value>

<currency>USD</currency>

</charge>

</bookingInfo>

</accommodationItem>

</activityItem>

<activityItem>

<reservationToken>

<inventoryType>1</inventoryType>

<referenceNumber>70840200100020001</referenceNumber>

</reservationToken>

<status>H</status>

<accommodationItem>

<resortId>2750</resortId>

<resortName>Cypress Pointe Resort</resortName>

<nbrOfBedrooms>2</nbrOfBedrooms>

<kitchenType>Full Kitchen</kitchenType>

<maxOccupancy>6</maxOccupancy>

<bookingInfo>

<checkInDate>08/04/2006</checkInDate>

<checkOutDate>08/11/2006</checkOutDate>

<charge>

<value>330.0</value>

<currency>USD</currency>

</charge>

</bookingInfo>

</accommodationItem>

<travelerInfo>

<firstName>MarcusD</firstName>

<middleName>D</middleName>

<lastName>McClellandD</lastName>

<addressline1>9999 N. Michigan Rd.</addressline1>

<addressline2>Line 2</addressline2>

<addressline3>Line 3</addressline3>

<city>Carmel</city>

<state>IN</state>

<country>USA</country>

<zipCode>46033</zipCode>

<homePhone>317 805-8001</homePhone>

<workPhone>317 805-8003</workPhone>

<email>indybuzz@yahoo.com</email>

</travelerInfo>

</activityItem>

</onHoldActivity>

<confirmedActivity />

<cancelledActivity />

</ActivityResponse>

</responseEnvelope>

Error

### Purpose

Returns error information for a failed request

### Response To

Any request

### DTD

<?xml version=“1.0”?>

<!ELEMENT responseEnvelope (responseHeader)>

<!ELEMENT responseHeader (errors)>

<!ELEMENT errors (error+)>

<!ELEMENT error (errorCode, errorMessage)>

<!ELEMENT errorCode (#PCDATA)>

<!ELEMENT errorMessage (#PCDATA)>

### Element Description

|  |  |  |
| --- | --- | --- |
| **ELEMENT NAME** | **DATA TYPE** | **DATA RULES** |
| errorCode | Number | * See response code list for other codes |

### XML Example

<?xml version=“1.0”?>

<responseEnvelope>

<responseHeader>

<errors>

<error>

<errorCode>5012</errorCode>

<errorMessage>ReferenceNumber is not specified.</errorMessage>

</error>

</errors>

</responseHeader>

</responseEnvelope>

InquireTravelerResponse

### Purpose

Returns traveler information for a specific reservation

### Response To

inquireTraveler

### DTD

<?xml version=“1.0”?>

<!ELEMENT responseEnvelope (responseHeader, InquireTravelerResponse)>

<!ELEMENT responseHeader (statusData)>

<!ELEMENT statusData (completionCode, completionMessage)>

<!ELEMENT completionCode (#PCDATA)>

<!ELEMENT completionMessage (#PCDATA)>

<!ELEMENT InquireTravelerResponse (reservationToken, travelerInfo)>

<!ELEMENT reservationToken (inventoryType, referenceNumber)>

<!ELEMENT inventoryType (#PCDATA)>

<!ELEMENT referenceNumber (#PCDATA)>

<!ELEMENT travelerInfo (firstName, middleName, lastName, addressline1, addressline2, addressline3, city, state, country, zipCode, homePhone, workPhone)>

<!ELEMENT firstName (#PCDATA)>

<!ELEMENT middleName (#PCDATA)>

<!ELEMENT lastName (#PCDATA)>

<!ELEMENT addressline1 (#PCDATA)>

<!ELEMENT addressline2 (#PCDATA)>

<!ELEMENT addressline3 (#PCDATA)>

<!ELEMENT city (#PCDATA)>

<!ELEMENT state (#PCDATA)>

<!ELEMENT country (#PCDATA)>

<!ELEMENT zipCode (#PCDATA)>

<!ELEMENT homePhone (#PCDATA)>

<!ELEMENT workPhone (#PCDATA)>

### Noteworthy Elements

|  |  |  |
| --- | --- | --- |
| **ELEMENT NAME** | **DATA TYPE** | **DATA RULES** |
| completionCode | Number | * 0 for success * See response code list for other codes |
| middleName | Text | * This is the middle initial, not name |

### XML Example

<?xml version=“1.0”?>

<responseEnvelope>

<responseHeader>

<statusData>

<completionCode>0</completionCode>

<completionMessage>Request Completed Successfully.</completionMessage>

</statusData>

</responseHeader>

<InquireTravelerResponse>

<reservationToken>

<inventoryType>1</inventoryType>

<referenceNumber>70840200100959001</referenceNumber>

</reservationToken>

<travelerInfo>

<firstName>JOHN</firstName>

<middleName />

<lastName>DOE</lastName>

<addressline1>9998 N. MICHIGAN RD.</addressline1>

<addressline2 />

<addressline3 />

<city>CARMEL</city>

<state>IN</state>

<country>USA</country>

<zipCode>46032</zipCode>

<homePhone>317 805-8000</homePhone>

<workPhone>317 805-8001</workPhone>

</travelerInfo>

</InquireTravelerResponse>

</responseEnvelope>

ReservationResponse

### Purpose

Returns reservation details for holds, confirmations, cancellations, and releases

### Response To

cancel

confirm

confirmHold

hold

release

### DTD

<?xml version=“1.0”?>

<!ELEMENT responseEnvelope (responseHeader, ReservationResponse)>

<!ELEMENT responseHeader (statusData)>

<!ELEMENT statusData (completionCode, completionMessage)>

<!ELEMENT completionCode (#PCDATA)>

<!ELEMENT completionMessage (#PCDATA)>

<!ELEMENT ReservationResponse (ReservationToken, status)>

<!ELEMENT ReservationToken (inventoryType, referenceNumber, unitNumber?)>

<!ELEMENT inventoryType (#PCDATA)>

<!ELEMENT referenceNumber (#PCDATA)>

<!ELEMENT unitNumber (#PCDATA)>

<!ELEMENT status (#PCDATA)>

### Noteworthy Elements

|  |  |  |
| --- | --- | --- |
| **ELEMENT NAME** | **DATA TYPE** | **DATA RULES** |
| completionCode | Number | * 0 for success * See response code list for other codes |
| unitNumber | Text | * The unit number is not necessarily known at time of booking, and may be assigned at check-in. * The unitNumber element is returned only for confirmations. |
| status | Text | * Indicates the status of the reservation H – Held C – Confirmed R – Released X – Cancelled |

### XML Example

The following is a sample response from a hold. Note that the reservation token does not include a unit number.

<?xml version=“1.0”?>

<responseEnvelope>

<responseHeader>

<statusData>

<completionCode>0</completionCode>

<completionMessage>Request Completed Successfully.</completionMessage>

</statusData>

</responseHeader>

<ReservationResponse>

<ReservationToken>

<inventoryType>1</inventoryType>

<referenceNumber>70840200100958001</referenceNumber>

</ReservationToken>

<status>H</status>

</ReservationResponse>

</responseEnvelope>

The following is a sample response from a confirmation. Note that the reservation token includes a unit number, since this resort assigns unit number at time of confirmation.

<?xml version=“1.0” ?>

<responseEnvelope>

<responseHeader>

<statusData>

<completionCode>0</completionCode>

<completionMessage>Request Completed Successfully.</completionMessage>

</statusData>

</responseHeader>

<ReservationResponse>

<ReservationToken>

<inventoryType>1</inventoryType>

<referenceNumber>70840200100959001</referenceNumber>

<unitNumber>3BEDLO</unitNumber>

</ReservationToken>

<status>C</status>

</ReservationResponse>

</responseEnvelope>

SearchResponse

### Purpose

Returns detailed information of resort availability

### Additional Information/Restrictions

* The same resort may be returned in multiple accommodationItem elements. However, the combination of resort ID, number of bedrooms, kitchen type, privacy occupancy, and maximum occupancy will be unique for each accommodationItem element.
* The accommodationItem elements and availabilityItem elements are in no particular order.

### Response To

priceSearch

quickSearch

resortIdSearch

### DTD

<?xml version=“1.0”?>

<!ELEMENT responseEnvelope (responseHeader, SearchResponse)>

<!ELEMENT responseHeader (statusData)>

<!ELEMENT statusData (completionCode, completionMessage)>

<!ELEMENT completionCode (#PCDATA)>

<!ELEMENT completionMessage (#PCDATA)>

<!ELEMENT SearchResponse (sessionToken?, accommodationItem+)>

<!ELEMENT sessionToken (sessionId)>

<!ELEMENT sessionId (#PCDATA)>

<!ELEMENT accommodationItem (resortId, resortName, nbrOfBedrooms, nbrOfBathrooms?, kitchenType?, privacyOccupancy?, maxOccupancy?, availabilityInfo+ )>

<!ELEMENT resortId (#PCDATA)>

<!ELEMENT resortName (#PCDATA)>

<!ELEMENT nbrOfBedrooms (#PCDATA)>

<!ELEMENT nbrOfBathrooms (#PCDATA)>

<!ELEMENT kitchenType (#PCDATA)>

<!ELEMENT privacyOccupancy (#PCDATA)>

<!ELEMENT maxOccupancy (#PCDATA)>

<!ELEMENT availabilityInfo (systemId, checkInDate, checkOutDate, unitCost?, unitCount?)>

<!ELEMENT systemId (inventoryType, inventoryId+)>

<!ELEMENT inventoryType (#PCDATA)>

<!ELEMENT inventoryId (#PCDATA)>

<!ELEMENT checkInDate (#PCDATA)>

<!ELEMENT checkOutDate (#PCDATA)>

<!ELEMENT unitCost (value, currency)>

<!ELEMENT value (#PCDATA)>

<!ELEMENT currency (#PCDATA)>

<!ELEMENT unitCount (#PCDATA)>

### Noteworthy Elements

|  |  |  |
| --- | --- | --- |
| **ELEMENT NAME** | **DATA TYPE** | **DATA RULES** |
| completionCode | Number | * 0 for success, even if there is no availability * See response code list for other codes |
| nbrOfBedrooms | Text | * The size of the unit. Will be one of:   + -1 – Hotel   + 0 – Studio   + 1 – 1 Bedroom   + 2 – 2 Bedroom   + 3 – 3 Bedroom   + 4 – 4 Bedroom * Note that the values for hotel and studio units are letters in the search request, but numbers in the SearchResponse. |
| nbrOfBathrooms | Decimal | * This element always returns 0.0 for Weeks system (inventory type 1) inventory, and should be ignored |
| kitchenType | Text | * Will be one of the following values:   + [empty tag] – No kitchen facilities are available in the unit   + Partial Kitchen – Not all major appliances area available in the unit   + Mini Kitchen – All major appliances are available in the unit, but some are mini size   + Full Kitchen – All major appliances are available in the unit, in full size |
| maxOccupancy | Number | * The maximum number of people (including children) this unit will accommodate in permanent sleeping fixtures. This is strictly enforced by some resorts. |
| privacyOccupancy | Number | * The number of adults who would occupy the unit if more privacy is desired. This is based on private sleeping areas (2 adults per sleeping area) and private lavatory access. |
| inventoryType | Number | * Identifies in which back-end system the inventory exists. This provided no value to the wholesaler/customer, but is required as input in several of the XML requests. |
| inventoryId | Text | * This is an encrypted value of the key to the inventory on the back end. Each available unit will have a different inventory ID. |
| checkInDate | Date | * The date that the unit is available for check-in. |
| checkOutDate | Date | * The date by which the customer must check out of the unit . |
| value | Decimal | * The price the wholesaler must pay RCI when confirming/booking this unit. This exact amount must be set as the “value” element when sending a “confirm” or “hold” XML request. |
| currency | Text | * As of this release, the value will be “USD”. As of the date this document was written, there are no plans to provide additional currencies. |
| unitCount | Number | * The number of like units available with the same unit configuration and check-in date. * The value returned is 1 in most cases, although more like units may be available. Due to a system limitation, there are no plans to return the correct number of units available. |

### XML Example

The following example returns 2 available units at resort 1788 (one for week 10/14/06 and one for week 10/21/06), and 1 available unit at resort AA99 (for week 10/7/06).

<?xml version=“1.0” ?>

<responseEnvelope>

<responseHeader>

<statusData>

<completionCode>0</completionCode>

<completionMessage>Request Completed Successfully.</completionMessage>

</statusData>

</responseHeader>

<SearchResponse>

<sessionToken>

<sessionId>093216570</sessionId>

</sessionToken>

<accommodationItem>

<resortId>1788</resortId>

<resortName>FRENCH LICK SPRINGS VILLAS</resortName>

<nbrOfBedrooms>2</nbrOfBedrooms>

<nbrOfBathrooms>0.0</nbrOfBathrooms>

<kitchenType>Full Kitchen</kitchenType>

<maxOccupancy>8</maxOccupancy>

<privacyOccupancy>6</privacyOccupancy>

<availabilityInfo>

<systemId>

<inventoryType>1</inventoryType>

<inventoryId>8XCxokL76z2HxcVKJrxIKYbtHlW3OKnl8+Hupr4UYoU+jGdq4blB1hRm9H1VnyAQ</inventoryId>

</systemId>

<checkInDate>10/21/2006</checkInDate>

<checkOutDate>10/28/2006</checkOutDate>

<unitCost>

<value>219.0</value>

<currency>USD</currency>

</unitCost>

<unitCount>1</unitCount>

</availabilityInfo>

<availabilityInfo>

<systemId>

<inventoryType>1</inventoryType>

<inventoryId>8XCxokL76z235JJR4PiSZ4btHlW3OKnl8+Hupr4UYoU+jGdq4blB1hRm9H1VnyAQ</inventoryId>

</systemId>

<checkInDate>10/14/2006</checkInDate>

<checkOutDate>10/21/2006</checkOutDate>

<unitCost>

<value>219.0</value>

<currency>USD</currency>

</unitCost>

<unitCount>1</unitCount>

</availabilityInfo>

</accommodationItem>

<accommodationItem>

<resortId>AA99</resortId>

<resortName>AA99-ALPHA RESORT TEST</resortName>

<nbrOfBedrooms>1</nbrOfBedrooms>

<nbrOfBathrooms>0.0</nbrOfBathrooms>

<kitchenType>Partial Kitchen</kitchenType>

<maxOccupancy>2</maxOccupancy>

<privacyOccupancy>2</privacyOccupancy>

<availabilityInfo>

<systemId>

<inventoryType>1</inventoryType>

<inventoryId>59Lm6yhPndp3RzJM2dM05CmkVSOEg57q1rx9QDrDQh9ZQtWwnKUcyBRm9H1VnyAQ</inventoryId>

</systemId>

<checkInDate>10/07/2006</checkInDate>

<checkOutDate>10/14/2006</checkOutDate>

<unitCost>

<value>179.0</value>

<currency>USD</currency>

</unitCost>

<unitCount>1</unitCount>

</availabilityInfo>

</accommodationItem>

</SearchResponse>

</responseEnvelope>

Success

### Purpose

Acknowledgement of a successful traveler add request

### Response To

addTraveler

### DTD

<?xml version=“1.0”?>

<!ELEMENT responseEnvelope (responseHeader)>

<!ELEMENT responseHeader (statusData)>

<!ELEMENT statusData (completionCode, completionMessage)>

<!ELEMENT completionCode (#PCDATA)>

<!ELEMENT completionMessage (#PCDATA)>

### Noteworthy Elements

|  |  |  |
| --- | --- | --- |
| **ELEMENT NAME** | **DATA TYPE** | **DATA RULES** |
| completionCode | Number | * 0 for success, even if there is no availability * See response code list for other codes |

### XML Example

<?xml version=“1.0”?>

<responseEnvelope>

<responseHeader>

<statusData>

<completionCode>0</completionCode>

<completionMessage>Request Completed Successfully.</completionMessage>

</statusData>

</responseHeader>

</responseEnvelope>

# Appendix A: Detailed Resort Info

For detailed specifications on accessing our resort content, please request the latest version of our **Resort Content Syndication Specification** document.

# Appendix B: Valid Countries

The following list provides the valid names/spellings of countries allowed for travelers (guest certificates)

Albania

Algeria

Andorra

Angola

Antigua

Argentina

Armenia

Australia

Azerbijan

Bahamas

Bahrain

Bangladesh

Barbados

Barbuda

Belarus

Belgium

Belize

Benin

Bermuda

Bolivia

Bosnia-Herzegovina

Botswana

Brazil

British Virgin Islands

Brunei

Bulgaria

Burkina Faso

Burma

Burundi

Cambodia

Cameroon

Canada

Canary Islands

Cayman Islands

Central African Republic

Chad

Channel Islands

Chile

China

Columbia

Congo

Cook Islands

Costa Rica

Croatia

Cyprus

Czech Republic

Denmark

Djibouti

Dominica

Dominican Republic

Ecuador

Egypt

El Salvador

England

Equatorial Guinea

Estonia

Ethiopia

Faeroe Islands

Falkland Islands

Fiji

Finland

Formosa

France

French Guiana

French Polynesia

Gabon

Gambia

Georgia

Germany

Ghana

Gibraltar

Greece

Greenland

Grenada

Guadeloupe

Guam

Guatemala

Guinea

Guinea - Bissau

Guyana

Haiti

Honduras

Hong Kong

Hungary

Iceland

India

Indonesia

Ireland

Israel

Italy

Ivory Coast

Jamaica

Japan

Jordan

Kazakhstan

Kenya

Korea, Rep of (North)

Korea, Rep of (South)

Kuwait

Kyrgyzstan

Laos

Latvia

Lebanon

Leeward Islands

Lesotho

Liberia

Liechtenstein

Lithuania

Luxembourg

Macao

Macedonia

Madagascar

Malawi

Malaysia

Maldives

Mali

Malta

Martinique

Mauritania

Mauritius

Mexico

Moldova

Monaco

Mongolia

Morocco

Mozambique

Namibia

Nepal

Netherland Antilles

Netherlands

New Caledonia

New Guinea

New Zealand

Nicaragua

Nigeria

Norfolk Island

North Ireland

Northern Cyprus,Turkish Republic of

Northern Ireland

Norway

Oman

Pakistan

Palau

Panama

Papua New Guinea

Paraguay

Peru

Philippines

Poland

Portugal

Puerto Rico

Qatar

Republic of Nauru

Reunion (Fr)

Romania

Russia

Rwanda

Saint Lucia

Saudi Arabia

Scotland

Senegal

Seychelles

Sierra Leone

Singapore

Slovakia

Slovenia

Solomon Islands

Somalia

South Africa

South Yemen, Republic of

Spain

Sri Lanka

St. Kitts, W.I.

St. Vincent and The Grenadines

Suriname

Swaziland

Sweden

Switzerland

Syria

Tahiti

Taiwan

Tajikistan

Tanzania

Thailand

Tobago

Togo

Trinidad and Tobago

Tunisia

Turkey

Turkmenistan

USA

Uganda

Ukraine

United Arab Emirates

Uruguay

Uzbekistan

Venezuela

Vietnam

Wales

Western Sahara

Western Samoa

Yemen

Zaire

Zambia

Zimbabwe

# Appendix C: Valid States

The following lists show the valid state codes for countries where a state code is required for the “addTraveler” request, as well as some zip/postal code rules. Only these four countries accept a state code; the state code should be blank for other countries.

## Australia

|  |  |
| --- | --- |
| **Territory** | **Code** |
| Australian Capital Territory | ACT |
| New South Wales | NSW |
| Northern Territory | NT |
| Queensland | QLD |
| Southern Australia | SA |
| Tasmania | TAS |
| Victoria | VIC |
| Western Australia | WA |

## Canada

|  |  |
| --- | --- |
| **Province** | **Code** |
| Alberta | AB |
| British Columbia | BC |
| Labrador | LB |
| Manitoba | MB |
| New Brunswick | NB |
| Newfoundland | NL |
| Northwest Territories | NT |
| Nova Scotia | NS |
| Nunavut | NU |
| Ontario | ON |
| Prince Edward Island | PE |
| Quebec | QC |
| Saskatchewan | SK |
| Yukon Territory | YT |

## Mexico

|  |  |
| --- | --- |
| **State** | **Code** |
| Aguascalientes | AGS |
| Baja California Norte | BCN |
| Baja California Sur | BCS |
| Campeche | CAMP |
| Chiapas | CHIS |
| Chihuahua | CHIH |
| Coahuila | COAH |
| Colima | COL |
| Distrito Federal (Capital) | DF |
| Durango | DGO |
| Guanajuato | GTO |
| Guerrero | GRO |
| Hidalgo | HGO |
| Jalisco | JAL |
| ? | MEX |
| Michoacan | MICH |
| Morelos | MOR |
| Nayarit | NAY |
| Nuevo Leon | NL |
| Oaxaca | OAX |
| Puebla | PUE |
| Queretaro | QRO |
| Quintana Roo | QROO |
| San Luis Potosi | SLP |
| Sinaloa | SIN |
| Sonora | SON |
| Tabasco | TAB |
| Tamaulipas | TAMP |
| Tlaxcala | TLAX |
| Veracruz | VER |
| Yucatan | YUC |
| Zacatecas | ZAC |

## US & US Territories

|  |  |
| --- | --- |
| **State/Territory** | **Code** |
| Apo/Fpo | AA, AE, AP |
| American Samoa | AS |
| Guam | GU |
| Puerto Rico | PR |
| Trust Territory | TT |
| Virgin Islands | VI |
| Alabama | AL |
| Alaska | AK |
| Arizona | AZ |
| Arkansas | AR |
| California | CA |
| Colorado | CO |
| Connecticut | CT |
| Delaware | DE |
| District Of Col | DC |
| Florida | FL |
| Georgia | GA |
| Hawaii | HI |
| Idaho | ID |
| Illinois | IL |
| Indiana | IN |
| Iowa | IA |
| Kansas | KS |
| Kentucky | KY |
| Louisiana | LA |
| Maine | ME |
| Maryland | MD |
| Massachusetts | MA |
| Michigan | MI |
| Minnesota | MN |
| Mississippi | MS |
| Missouri | MO |
| Montana | MT |
| Nebraska | NE |
| Nevada | NV |
| New Hampshire | NH |
| New Jersey | NJ |
| New Mexico | NM |
| New York | NY |
| North Carolina | NC |
| North Dakota | ND |
| Ohio | OH |
| Oklahoma | OK |
| Oregon | OR |
| Pennsylvania | PA |
| Rhode Island | RI |
| South Carolina | SC |
| South Dakota | SD |
| Tennessee | TN |
| Texas | TX |
| Utah | UT |
| Vermont | VT |
| Virginia | VA |
| Washington | WA |
| West Virginia | WV |
| Wisconsin | WI |
| Wyoming | WY |

# Appendix D: Zip/Postal Codes

The following list shows where the zip code should be placed in the address, for countries that use zip codes. All other countries should leave the zipCode field blank in XML requests.

## Location of zip/postal code in traveler addresses

For the following countries, place the zip/postal code in the zip code field:

Australia

Canada

England

Ireland

Japan

Mexico

Northern Ireland

Scotland

USA

Wales

For the following countries, place the zip/postal code in the city field *before* the city name:

Argentina

Austria

Belgium

Denmark

El Salvador

France

Germany

Greece

Israel

Italy

Luxembourg

Monaco

Nicaragua

Norway

South Africa

Spain

Sweden

Switzerland

Yugoslavia

For the following countries, place the zip/postal code in the city field *after* the city name:

Bahamas

Bolivia

Brazil

Colombia

Costa Rica

Czechoslovakia

Dominican Republic

Egypt

Hong Kong

India

Indonesia

Iran

Iraq

Lebanon

Libya

New Zealand

Pakistan

Panama

Philippines

Saudia Arabia

Singapore

Sudan

Thailand

Turkey

United Arab Emirates

Venezuela

Western Samoa

## Zip/Postal code formatting

### Canada

Canadian zip codes must be in the following format:

Letter Number Letter Space Number Letter Number

(e.g.: L4J 3B2)

For each of the following provinces, the first character of the zip code must be one of the letters specified

|  |  |
| --- | --- |
| **Province** | **First Letter** |
| Alberta | T |
| British Columbia | V |
| Labrador | A |
| Manitoba | R |
| New Brunswick | E |
| Newfoundland | A |
| Northwest Territories | X |
| Nova Scotia | B |
| Nunavut |  |
| Ontario | K,L,M,N,P |
| Prince Edward Island | C |
| Quebec | G,H,J |
| Saskatchewan | S |
| Yukon Territory | Y |

### USA

US zip codes must be 5 numeric characters. Zip+4 values are not accepted.

The following list shows the Zip code range for the “addTraveler” request

|  |  |  |
| --- | --- | --- |
| **Zip Code Range** | **State Code** | **State** |
| 00600-00999 | PR | Puerto Rico |
| 00800-00899 | VI | Virgin Islands |
| 01000-02799 | MA | Massachusetts |
| 02800-02999 | RI | Rhode Island |
| 03000-03899 | NH | New Hampshire |
| 03900-04999 | ME | Maine |
| 05000-05999 | VT | Vermont |
| 06000-06999 | CT | Connecticut |
| 07000-08999 | NJ | New Jersey |
| 09000-09899 | AE | Apo/Fpo |
| 09900-14999 | NY | New York |
| 15000-19699 | PA | Pennsylvania |
| 19700-19999 | DE | Delaware |
| 20000-20599 | DC | District Of Col |
| 20600-21999 | MD | Maryland |
| 22000-24699 | VA | Virginia |
| 24700-26899 | WV | West Virginia |
| 27000-28999 | NC | North Carolina |
| 29000-29999 | SC | South Carolina |
| 30000-31999 | GA | Georgia |
| 32000-34999 | FL | Florida |
| 35000-36999 | AL | Alabama |
| 37000-38599 | TN | Tennessee |
| 38600-39799 | MS | Mississippi |
| 40000-42799 | KY | Kentucky |
| 43000-45899 | OH | Ohio |
| 46000-47999 | IN | Indiana |
| 48000-49999 | MI | Michigan |
| 50000-52899 | IA | Iowa |
| 53000-54999 | WI | Wisconsin |
| 55000-56799 | MN | Minnesota |
| 57000-57799 | SD | South Dakota |
| 58000-58899 | ND | North Dakota |
| 59000-59999 | MT | Montana |
| 60000-62999 | IL | Illinois |
| 63000-65899 | MO | Missouri |
| 66000-67999 | KS | Kansas |
| 68000-69399 | NE | Nebraska |
| 70000-71499 | LA | Louisiana |
| 71600-72999 | AR | Arkansas |
| 73000-74999 | OK | Oklahoma |
| 75000-79999 | TX | Texas |
| 80000-81699 | CO | Colorado |
| 82000-83199 | WY | Wyoming |
| 83200-83899 | ID | Idaho |
| 84000-84799 | UT | Utah |
| 85000-86599 | AZ | Arizona |
| 87000-88499 | NM | New Mexico |
| 89000-89899 | NV | Nevada |
| 90000-96199 | CA | California |
| 96200-96699 | AP | Apo/Fpo |
| 96700-96799 | AS | American Samoa |
| 96700-96899 | HI | Hawaii |
| 96900-96999 | GU | Guam |
| 96900-96999 | MP | Mariana Islands |
| 96900-96999 | PW | Palau |
| 97000-97999 | OR | Oregon |
| 98000-99499 | WA | Washington |
| 99500-99999 | AK | Alaska |

Following is the list of exceptions for the Zip code range

|  |  |  |
| --- | --- | --- |
| **Zip Code** | **State Code** | **State** |
| 75502 | AR | Arkansas |
| 6379 | CT | Connecticut |
| 51510 | IA | Iowa |
| 45275 | KY | Kentucky |
| 71749 | LA | Louisiana |
| 3801 | ME | Maine |
| 20331 | MD | Maryland |
| 6390 | NY | New York |
| 73949 | TX | Texas |
| 20041 | VA | Virginia |
| 20370 | VA | Virginia |
| 20301 | VA | Virginia |
| 49936 | WI | Wisconsin |
| 39800-39899 | GA | Georgia |
| 34000-34099 | AA | APO/FPO |
| 20100-20199 | VA | Virginia |

### Mexico

The following list shows the Zip code range for Mexico.

For state code DF, first 2 digits of the Zip code should be between the numbers 01 to 16 and the next three digits can be any numeric value.

The valid 5 digit zip code for state DF can be:

              01456 or 12789 or 1643 or …..

|  |  |  |
| --- | --- | --- |
| **Zip code Range** | **State Code** | **State Name** |
| 01xxx-16xxx: | DF | Mexican Federal District (México, D.F.) |
| 20xxx-20xxx: | AGS | Aguascalientes |
| 21xxx-22xxx: | BCN | Baja California |
| 23xxx-23xxx: | BCS | Baja California Sur |
| 24xxx-24xxx: | CAMP | Campeche |
| 25xxx-27xxx: | COAH | Coahuila |
| 28xxx-28xxx: | COL | Colima |
| 29xxx-30xxx: | CHIS | Chiapas |
| 31xxx-33xxx: | CHIH | Chihuahua |
| 34xxx-35xxx: | DGO | Durango |
| 36xxx-38xxx: | GTO | Guanajuato |
| 39xxx-41xxx: | GRO | Guerrero |
| 42xxx-43xxx: | HGO | Hidalgo |
| 44xxx-49xxx: | JAL | Jalisco |
| 50xxx-57xxx: | MEX | México State (Estado de México) |
| 58xxx-61xxx: | MICH | Michoacán |
| 62xxx-62xxx: | MOR | Morelos |
| 63xxx-63xxx: | NAY | Nayarit |
| 64xxx-67xxx: | NL | Nuevo León |
| 68xxx-71xxx: | OAX | Oaxaca |
| 72xxx-75xxx: | PUE | Puebla |
| 76xxx-76xxx: | QRO | Querétaro |
| 77xxx-77xxx: | QROO | Quintana Roo |
| 78xxx-79xxx: | SLP | San Luis Potosí |
| 80xxx-82xxx: | SIN | Sinaloa |
| 83xxx-85xxx: | SON | Sonora |
| 86xxx-86xxx: | TAB | Tabasco |
| 87xxx-89xxx: | TAMP | Tamaulipas |
| 90xxx-90xxx: | TLAX | Tlaxcala |
| 91xxx-96xxx: | VER | Veracruz |
| 97xxx-97xxx: | YUC | Yucatán |
| 98xxx-99xxx: | ZAC | Zacatecas |