BOOK YOUR HOLIDAY

hostelsclub

Hostelsclub.com

Web Service Integration Guide for Affiliates

Version 1.2.3 (December 2012)

Index

1. About this document

This document is the webmaster's guide for the implementation of Hostelsclub.com's Web Service in a site or booking system. It will provide a basic documentation of the various XML and JSON-based requests and responses, a list of possible errors and error codes, as well as some suggestions on how to implement the system in your server.

1.1. Who should read this document

This document is meant for people who have technical experience of XML, JSON and programming in general. If you do not have this kind of experience, this document will probably provide you with some information on what can be achieved with our Web Service, but a good deal of it will look quite obscure.

1.2. Conventions used in this documentation

When we mention a tag in the XML messages, it will always be written like this: <tag>. When we mention a tag attribute, it will appear in *Italic*. Whenever a tag features a long series of attributes, they will be described in a table containing the attribute name in the left column, and the explanation on the right column.

Properties of JSON objects will appear in *Italic*.

1.2.1. Prices

All prices contained in the messages are output using a dot (.) for decimal places. Prices will have a maximum of two decimal places, regardless of the currency used. Prices might, in some instances, not have decimal places at all or have just one decimal place.

1.2.2. Capitalization

XML System: All the system, both for tag names and attribute values, is case-sensitive. This means that if you should send the <HotelReservations> tag and you send <hotelreservations> instead, our system will not recognize it. If an attribute value is "true" and you send "TRUE" or "True", it will not be recognized. If you send the *Target* attribute with a value of "test", instead of "Test", it will not be recognized and the system will switch to production mode.

Always use the same capitalization of the examples, copying and pasting from the documentation if necessary, or making reference to the OTA standards when in doubt.

JSON System: In general, we've resorted to camelCase for most variable and property names. All variable names are case sensitive.

2. What does this Web Service actually do?

This Web Service allows webmasters to make requests to our booking engine, receive an XML- or JSON-formatted reply, and display the information on availability and reservations directly in their site, with the preferred formatting.

The Web Service covers all of our standard reservation procedure, from the basic step of searching what bookable properties are present in a city to the final step of booking a reservation sending customer and credit card data.

The Web Service can be implemented at many levels, for example it can be used for the basic availability search, property and room choice, transferring the user to your personalized booking engine on our servers for the final payment procedure. It allows a great deal of flexibility on how you display search results, how you let users choose the property they want to book, and how you integrate this booking procedure with existing procedures in your system.

2.1. XML Web Service

Our legacy Web Service is based on XML and the Open Travel Alliance standards. It currently covers all features in our system, from hotel search to complete hotel reservation.

It is described from chapter 3 onwards.

2.2. JSON Web Service

As we've found out with time that the Open Travel Alliance standards can be demanding to implement, and that big XML messages have a significant overhead in terms of processing power needed to decode them (especially OTA-based messages), we decided to roll out a more lightweight, JSON-based system. This system currently covers the **affiliate function that is most frequently used by affiliates** (and most demanding in terms of performance)**, that is hotel search** in a city or area. The new system introduces new capabilities: searches for up to 30 days and new filters for things like room type, number of guests, facilities, in order to increase conversion by letting users find exactly what they need. The JSON Web Service replies directly with a list of properties with **the rooms that have availability matching the request**, allowing a quicker choice and checkout process.

Furthermore, since the JSON format is widely used in web applications, it will be easier for your application to directly pass data through to your Javascript frontend without having to process them at all.

The JSON Web Service is described from chapter 12 onwards.

3. *The XML web service - Open Travel Alliance Standards*

Our XML implementation follows the standards designed by the Open Travel Alliance[1] (OTA, http://www.opentravel.org/) standards 2006A.

This standard was developed by the OTA to make it easier for companies to implement Web Services and therefore to facilitate trade between companies. Refer to the web address above for more information on the OTA.

The OTA standards cover almost all possible business cases for each type of request, so our implementation is inevitably a subset of the complete standard, and won't feature all possible combination of elements and requests provided for by the OTA standard.

This guide you are reading should be sufficient to create a set of requests that are compatible with our implementation of the OTA standards. However, if you want to know more about the standards themselves, please visit the web address above.

4. *Documentation version*

As we add more features to our booking system, this documentation will be revised in order to illustrate changes and the new functionalities installed in our system. Please make sure you have the latest version. We will contact all affiliates who are using the system whenever there are changes to the available features, and provide them with the new documentation.

Every effort will be made to make changes to the system in a backwards-compatible way. If we will ever need to make changes that could disrupt the correct functioning of existing applications, we will warn you well in advance, so that you have time to make the appropriate changes to your system.

4.1. What's new in this version

This is a list that tracks changes made with every version of our system. If you have an out of date version of our documentation and receive a new one, here you will find all changes that took place since your version of the documentation.

Version 1.2.3 – added numerical overall rating to customer ratings in OTA_HotelDescriptiveInfoRS. Added Occupancy node to OTA_HotelAvailRS in the single hotel search mode.

Version 1.2.2 – added room breakdown in OTA_HotelAvailRQ: CustomerDataJSON tag

Version 1.2.1 - minor corrections to JSON Webservice documentation

Version 1.2.0 - new JSON-based Webservice for Availability Search

Version 1.1.2 - .the OTA_HotelResRQ now has the SeriesCode attribute, containing the CVV/CSV confirmation code for credit cards, which is compulsory for all card types except JCB.

Version 1.1.1 - SpecialRequests tag added in OTA_ResRetrieveRS and OTA_HotelResRS containing customer messages to the property.

Version 1.1 - There's a minor addition: many response messages now feature the geocoding coordinates of properties (when the data is available).

[1] All content related to the Open Travel Alliance is Copyright© Year 2006, OpenTravel™ Alliance, Inc.

5. Getting started

Our XML Web Service cannot be accessed by unauthorized requests. You will first need to contact our Affiliate department to get a test account activated before you can make test requests. Test requests will work exactly like normal requests, but they won't alter our database. This means that you'll be able to simulate a reservation or a reservation cancellation, but the actual reservation/cancellation will not take place.

After you've developed your implementation, a member of our staff will review it, and if your implementation passes the review, your account will be changed to a "production" account and you'll be able to produce actual reservations.

If your implementation does not pass the review, our staff will suggest the changes to be made to correct the problems encountered.

Your usage of our XML Web Service will be monitored by our staff: we reserve the right to close an XML account if its ratio of requests to reservations is not considered satisfactory. This is because the Service can consume a lot of resources, and if a big number of requests does not produce a reasonable number of reservations, then our resources are being wasted.

6. How to send requests

The addresses to which to send requests are the following:

http://www.hostelspoint.com/xml/xml.php

https://www.hostelspoint.com/xml/xml.php

The latter address (with https secure protocol) is required for requests that transmit sensitive data (customer data, reservation data, credit card data). More on this in the "Authentication" chapter below.

To receive a response, you must send your XML request as a POST variable called "OTA_request". This can be achieved through various programming methods, which will vary according to the programming language you're using. For example, in PHP the simplest way to send this kind of requests is using the CURL extension.

Make sure that you properly encode your XML request (URL encoding), otherwise ampersands (&s) which are present in your request will be interpreted by our system as the start of a new variable in POST and the XML will be considered invalid. If you don't have a tool that does this for you, you should basically substitute all ampersands with %26, which is the URL-equivalent of &. Make sure that all ampersands contained within the XML message are still XML-entity encoded, so they look like this: &amp; before the conversion, and %26amp; after the conversion.

Make sure that the content-type of your message is “application/x-www-form-urlencoded”, just like a normal HTML form would do.

Our system will return an XML file. If our system returns a blank page, without any XML, it is because it has found no data at all (the “OTA_request” has not been sent correctly).

7. Basic request structure

All requests must be sent in UTF-8 encoding.

A sample XML request will look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelDescriptiveInfoRQ EchoToken="1" Target="Production"
Version="1.002"
xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05
OTA_HotelDescriptiveInfoRQ.xsd"  PrimaryLangID="it">
      <POS>
            <Source>
                  <RequestorID ID="1"
MessagePassword="password1" />
            </Source>
      </POS>
      <HotelDescriptiveInfos>
            <HotelDescriptiveInfo HotelCode="1">
                  <HotelInfo SendData="true"/>
                  <FacilityInfo SendGuestRooms="true"/>
                  <Policies SendPolicies="true"/>
                  <ContactInfo SendData="true"/>
                  <MultimediaObjects SendData="true"/>
                  <ContentInfos>
                        <ContentInfo Name="CustomerRatings"/>
                        <ContentInfo Name="CustomerReviews"/>
                  </ContentInfos>
            </HotelDescriptiveInfo>
      </HotelDescriptiveInfos>
</OTA_HotelDescriptiveInfoRQ>
```

This is a sample request for descriptive information about hotel number 1 in our system (there's no hotel number 1 in our system, by the way). Do not worry about details of the request. At this step we will only look at the **structure** of the request itself.

In the first row, you see the XML declaration. Depending on how you will produce your XML, this might be generated automatically for you when you export the document tree.

The first element in the XML, the *root element*, must have a name of one of the supported request types, in this case the *OTA_HotelDescriptiveInfoRQ*. We will list all of these later on.

The root element may include several attributes: some of these are specific to one message type, others (such as xmlns) pertain to XML specifications, others are compulsory for all or most requests.

Here's a list of the main attributes supported by ALL calls to our system (all other optional, OTA-defined attributes that are not listed here and are not listed in specific request descriptions, such as AltLangID or SequenceNmbr, are ignored by our system – they can be sent anyway, though):

Attribute	***Notes***
EchoToken	This is valid for all messages. Although not compulsory, it is very important because it will be returned AS IT IS (as long as it does not exceed 128 chars length) by our system in an attribute of the same name in our response. It allows your system to pair your requests to our responses,

Attribute	***Notes***
	especially in the cases in which the same program instance is sending more than one request at a time. It can be a string of up to 128 characters.
Target	This attribute can contain the values "Production" or "Test". If this attribute is not sent, our system will default to "Production". However, if your account is not authorized for production requests, our system will switch to testing mode.
Version	This attribute refers to the OTA Version of the message being sent. Our system does not block requests that are earlier than the version we are supporting, and will try to reply to them if possible, but it will issue a warning about the fact that the version of the request is earlier than the one supported. If you're not familiar with OTA Message Versions, please use the ones in the sample requests contained in this document.
PrimaryLangID	This is expected to be a two-letter code identifying the language you want to receive customer-displayable data in. If the language is supported by our system, all text which should be displayed to the customer will be returned in that language, if available (for example, not all property descriptions are available in every language we support). All system error messages will still be issued in English. If the language you request is not supported or not recognized, the system will issue a warning and proceed returning data in English. Language codes follow ISO 639 standard (two-letters codes), but we also support the codes used on our site, which sometime differ from ISO 639 codes. For example, you can send *zh-cn* to identify simplified Chinese or *zh-tw* to identify traditional Chinese – but you can also send *cn* and *tw* only, since those two abbreviations are used on our public site for those two ways of writing the Chinese language. Refer to the Appendix for the complete code list – but you can easily spot the codes on our main site language menu, by looking at the urls (ex. index-es.html contains es for Spanish).
xmlns	This attribute is compulsory. It indicates the namespace of the xml document. It must be set exactly as you find it in each sample request in this document.

The inner content of each message (the tags contained in the root element) will vary from message to message, but authentication tags are common to most requests, so they will be discussed next.

8. Authentication

The only request that does not require authentication is, at the moment, the OTA_PingRQ, which is used for preliminary testing, or (where needed) to test whether our system is live and responding. All other requests require you to send authentication credentials using the POS tag as follows:

The POS tag (which must be positioned right after the root element) must contain a Source Tag, which in turn must contain a RequestorID tag. This last tag must **at least** contain an attribute ID containing the numeric ID of your affiliate **SITE**. Remember, this is your site ID, and not your affiliate ID. This is because we allow you to create multiple sites within an affiliate account, each of which can have a distinct name and a personalized booking interface. The site name will appear in confirmation e-mails to the customer, unless you decide to e-mail the customer yourself. Even in that case, it will still appear in our reporting system and in the establishments' back offices and confirmation e-mails.

The RequestorID can contain a MessagePassword tag, which must contain the password you will be assigned by us when we enable your XML account. **This is not your back office password**, but the password our staff will send you upon activation of your account. The password should be sent only through encrypted connections, for obvious reasons. This password will be checked only in https messages.

8.1. How does the authentication system work?

There are some requests that **require** an encrypted (https) connection. They are listed below. For all other requests, you are encouraged, if possible, to use an unencrypted connection. This is because unencrypted connections have a performance advantage, which is important for both parties (you and us). Basic searches, like city availability search, do not contain sensitive information and are not very different from what people will be able to see from our public sites.

In these cases, you should just send your site ID in the ID attribute of the RequestorID tag, and we will check that your site ID and the site's registered IP Address (you will be required to provide this when you ask for a test account) match.

The requests which require https are, at the moment, the following:

- **OTA_HotelResRQ**: the reservation request which passes customer data and credit card data
- **OTA_CancelRQ**: the reservation cancellation request
- **OTA_ReadRQ**: the reservation read request, which returns data about a reservation

All the requests above require you to send the password attribute and to use the https protocol. Otherwise, authentication will fail and the message will not be processed at all.

You can use https for all requests in the following cases:

- you are testing an implementation from a computer which has a variable IP connection
- your IP can vary from one call to another, for other reasons

9. Basic response structure

Before describing the various requests and response types, we will describe the various error levels, then the elements that are in common to all or most response types.

9.1. Low-level errors

The OTA specifications provide several error levels: the most basic errors are the following:

9.1.1. No data

If there is no POST variable called OTA_request, our system will reply with an empty page.

9.1.2. Malformed data

In case the XML you sent in the OTA_request variable is malformed, our system will reply with a general OTA_ErrorRS response, which will look like this:

```
<?xml version="1.0" encoding="UTF-8" ?><OTA_ErrorRS
xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/2006A/OTA_ErrorRS.xs
d"  TimeStamp="2007-01-09T11:15:31-05:00" Target="Production"
Version="1.004" PrimaryLangID="en" Status="NotProcessed"
ErrorCode="Malformed" ErrorMessage="Malformed XML - errors:
End tag : expected '>'
 - node name: EchoData - line 8 - column 1
Opening and ending tag mismatch: EchoData line 7 and Ech
 - node name: EchoData - line 8 - column 1
"></OTA_ErrorRS>
```

The ErrorCode attribute of the root element will contain the general error code, as per OTA specifications. The ErrorMessage will contain the error output of our parsing program, with details on errors in the received XML, in order to help you correct eventual errors.

9.1.3. Authentication Failure

In case your request could not be authenticated (either because the credentials are invalid, or you are trying to send a request that requires https through the http protocol), our system will reply with a general OTA_ErrorRS response, which will look like this:

```
<?xml version="1.0" encoding="UTF-8" ?><OTA_ErrorRS
xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/2006A/OTA_ErrorRS.xs
d"  EchoToken="1"  TimeStamp="2007-01-09T11:19:46-05:00"
Target="Production" Version="1.004" PrimaryLangID="tw"
Status="NotProcessed" ErrorCode="AuthFail"
ErrorMessage="Authentication failure"></OTA_ErrorRS>
```

The ErrorCode and ErrorMessage here are quite self-explanatory.

9.1.4. Message not supported

In case your request uses a root element (message type) which is incorrect, or not supported by our platform, the system will reply with a general OTA_ErrorRS response, which will look like this:

```
<?xml version="1.0" encoding="UTF-8" ?><OTA_ErrorRS
xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/2006A/OTA_ErrorRS.xs
d"  EchoToken="1"  TimeStamp="2007-01-09T11:21:26-05:00"
Target="Production" Version="1.004" PrimaryLangID="tw"
Status="NotProcessed" ErrorCode="UnrecognizedRoot"
ErrorMessage="REQUEST TYPE NOT RECOGNIZED OR NOT
SUPPORTED"></OTA_ErrorRS>
```

9.2. Success/Error tags and main response element

If your request has passed the low-level error checks above, the root element of the response will be that of the correct response type, corresponding to the request type you sent (for ex., *OTA_HotelAvailRS* for the *OTA_HotelAvailRQ* request type).

9.2.1. Root element attributes for the response message

The root element of the response message will always contain, beside *xmlns* and other standard attributes, the following attributes:

Attribute	***Notes***
EchoToken	This returns the content of the attribute of the same name in your request (see above). It allows your system to pair your requests to our responses, especially in the cases in which the same program instance is sending more than one request at a time.
Target	This attribute can contain the values "Production" or "Test". If this attribute has not been sent in your request, our system will default to "Production". However, if your account is not authorized for production requests, our system will switch to testing mode. In all cases which require some action on our databases, this attribute will tell you if your request has actually been processed: if this attribute is set to "Test" in our response, it means that no actual reservation or cancellation has taken place, but they have just been simulated for testing purposes. Remember that we must review your implementation before you can perform actual "Production" requests.
Version	This attribute refers to the OTA Version of the message being sent. Our system does not block requests that are earlier than the version we are supporting, and will try to reply to them if possible, but it will issue a warning about the fact that the version of the request is earlier than the one supported. If you're not familiar with OTA Message Versions, please use the ones in the sample requests contained in this document.
PrimaryLangID	If the system has recognized your language code, or part of it (for example, we generally parse only the first two letters of language codes like en-gb, with the exception of zh-tw and zh-cn for Traditional and Simplified Chinese), it will return it as you sent it. All text meant for consultation by the end user (establishment descriptions, reservation conditions, etc.) will be translated in that language, if a translation is available. Most error/warning messages will still be in English, however.
TimeStamp	An attribute containing the time stamp of message processing according to the ISO 8601 standard: YYYY-MM-DDThh:mm:ssZ with time values using the 24 hours (e.g. 20 January 2006, 2:58:12 pm UTC becomes 2006-01-20T14:58:12Z). The Z in the pattern above will be substituted with the time zone indication of the responding server, taking into account daylight saving time.

9.2.2. Request results

If your request has passed the low-level error checks, there are three possible results of our processing it:

1. The message **contains insufficient data to process the request**: for example, your request for availability does not contain a way of identifying a property ID, or

your request for reservation does not contain the required credit card data, or some data is not correctly formatted.

2. The message **contains the correct data**, but the **request itself failed**: for example, there are no establishments with availability in the city requested, or the rooms that you are trying to book are not available any more, or the credit card has been refused.
3. The message **contains the correct data**, and the **request is successful**: this is the case when everything goes as intended: your request is correct, we could process it and have a positive response.

The OTA Standards provide for two different ways of indicating that a message was not successful, one covering case 1. above, and the other covering case 2.

9.2.3. The <Errors> tag - insufficient data

In the first case, the response message contains, as the first child of the root element, an <Errors> tag containing one or more <Error> tags indicating the code for each error encountered and a brief textual explanation of the error itself. Each <Error> tag will have two attributes, *Code* indicating the actual error code according to OTA standards, and *Type* indicating the type of error (for ex. Required field missing, Authentication, Business Rules) according to OTA standards. The <Error> tag will have a text content which explains exactly what error took place.

You will find a list of error codes (and warning codes) used in the appendix of this document, with explanation of what they mean.

The response XML will look like this:

```
<?xml version="1.0" encoding="utf-8"?>
<OTA_HotelAvailRS xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05
OTA_HotelAvailRS.xsd" TimeStamp="2007-01-09T12:11:25-05:00"
Version="1.006" PrimaryLangID="de" Target="Test">
  <Errors>
    <Error Code="321" Type="3"> No means of identifying the city
requested found. You can pass a Hclub city id in the HotelRef tag
- HotelCityCode attribute or a city name with country and
optionally state in the Address tag</Error>
  </Errors>
</OTA_HotelAvailRS>
```

As you can see, here the root element is *OTA_HotelAvailRS* (which is the response for a request for availability, as opposed to the *OTA_ErrorRS* above). But the message's infor was not sufficient to proceed, therefore the response only contaiins <Error> tags.

9.2.4. The <Success/> and <Warnings> tags

The second case is a bit trickier. The message has been successfully parsed, and so you will find a <Success/> tag (with no attributes or content) as first child of the root element. This is the indicator of the fact that the message was successfully processed.

However, this does not guarantee that there is data to respond to your request. This might be for any reason listed above at point 2. In this case, there will be a <Warnings> tag containing as many warnings as the number of "errors" encountered in processing your request. In the case of a **failed request**, the <Warnings> tag will follow the <Success/>

tag, and will not be followed by other tags, as in the following example:

```
<?xml version="1.0" encoding="utf-8"?>
<OTA_HotelAvailRS xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05
OTA_HotelAvailRS.xsd" TimeStamp="2007-01-09T12:16:14-05:00"
Version="1.006" PrimaryLangID="de" Target="Test">
  <Success/>
  <Warnings>
    <Warning Code="357" Type="10"> City not found in
database</Warning>
  </Warnings>
</OTA_HotelAvailRS>
```

In the case above, there is one single <Warning> tag, which indicates that although you passed a sufficient amount of data to proceed with a city search, the search itself has returned no results. <Warning> tags also have a *Code* and *Type* attribute and a textual content, exactly like <Error> tags. The codes follow OTA standards.

9.2.5. The main response element

If a request is successful, it can still return warnings, usually "trivial" ones that are meant to point out small problems concerning your request, which could be ignored or corrected by our application.

The way to test that a request was *really successful*, you have to look for the *main response element*. According to the OTA Standards, every message has one or more main response elements (although our implementation usually uses just one per message use case), depending on the type of request.

Let's look at a simple example, using the *OTA_PingRQ*, which basically just tests that the system is live and that it correctly returns data:

```
<?xml version="1.0" encoding="utf-8"?>
<OTA_PingRS xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05
OTA_PingRS.xsd" TimeStamp="2007-01-09T12:23:07-05:00"
Version="1.001" PrimaryLangID="en" Target="Test">
  <Success/>
  <Warnings>
    <Warning Code="184" Type="3"> Request language not supported -
switching to English</Warning>
    <Warning Code="999" Type="2"> Your request's version is
earlier than our supported version. We tried to process your
request in case the versions are compatible. We support version
1.001 for the OTA_PingRQ call.</Warning>
  </Warnings>
  <EchoData>Hello</EchoData>
</OTA_PingRS>
```

The main response element for this request type is the <EchoData> tag, which reflects exactly the content of the corresponding tag in the request message (apart from the fact that we will truncate its content if it is too long).

The response above tells that the message was successfully parsed, but two warnings were issued: one concerning the language of the request, which was not recognized (in these cases our system switches to English), and another concerning the fact that the message version of the request was lower than 1.001, which is the minimum request version for our implementation (but our system will try to parse it anyway in case the messages are compatible).

Besides that, the request was successful and the response returns the "Hello" content in the <EchoData> tag.

So, when you want to check that the request was successful, we suggest you to check if the expected main response node (or nodes) for your requests is actually present in the response. In that case, you should parse its contents and proceed with displaying data to the user. Otherwise, you should check Warnings first and Errors next. Warnings and Errors exclude themselves mutually: you will not find a response containing both Errors and Warnings.

9.2.6. Errors vs. Warnings

Sometimes the difference between Errors and Warnings in the programming logic is a bit blurry: what must be considered an error and what must be considered a warning? We always followed the logic mentioned above: if the data is sufficient, we try to process the request and issue warnings if it fails. If the data is insufficient or incorrect in format, we issue an error. In any case, the text in the Error/Warning description will make clear what kind of problem was encountered.

10. Business logic

The OTA standards allow to implement almost any business logic with their messages: we (inevitably) adapted them to our business logic, trying to use the <TPA_Extensions> tag as little as possible; the <TPA_Extensions> tag is used to include XML data that cannot fit the standard request models. In some cases (both at request and response level) this could not be avoided.

Our implementation follows the business logic of our public site: the booking process is divided into 4 steps:

1. **Search** – in which the customer searches for properties in a determined city with have availability on certain dates
2. **Select** - in which the customer selects rooms from a property with availability on those dates
3. **Book** – in which the customer is assigned the final room choice and inserts his/her personal data and credit card data
4. **Confirmed** – in which the customer receives a response on whether the reservation was successful

Since it is possible that you desire to implement just a part of the booking procedure (for example, if you want to "keep" users on your site until the booking procedure, which requires https and a bit tougher to implement), we will provide you with instructions on how to "exit" from steps 1 and 2 above, and direct customers to our public site www.hostesclub.com OR to your personalized booking interface running under domain www.hostelspoint.com/res (depending on your preferences).

We also implemented the following procedures:

1. **Search for properties in a city** regardless of availability (useful for building city pages in your site)
2. Request of various kind of **info about a property** (description, pictures, customer ratings, booking requirements, etc.)
3. Request for **details on a reservation**
4. Reservation **cancellation**
5. Ping – a way to tell if our system is live (and a basic message useful for initial testing of your calling system)

The list of hotels and most details about a single property should be cached on your system, since they will not change that frequently. We strongly encourage you to cache that info on a daily or weekly basis so that you don't have to perform repeated calls to our servers, always asking for the same information.

Most of the above procedures have a separate request/response message, but some of them (due to the way the OTA Standards have been designed) will share the same request/response messages, with different implementation of the same message to perform the desired request.

Here's the correspondence of the above procedures to the request/response message couples:

Procedure	***Request message***	***Response message***
Search – in which the customer searches for properties in a determined city with have availability on certain dates	OTA_HotelAvailRQ	OTA_HotelAvailRS
Select - in which the customer selects rooms from a property with availability on those dates	OTA_HotelAvailRQ	OTA_HotelAvailRS
Book – in which the customer is assigned the final room selection and inserts his/her personal data and credit card data	OTA_HotelAvailRQ	OTA_HotelAvailRS
Confirmed – in which the customer receives a response on whether the reservation was successful	OTA_HotelResRQ	OTA_HotelResRS
Search for properties in a city regardless of availability (useful for building city pages in your site)	OTA_HotelSearchRQ	OTA_HotelSearchRS
Request of various kind of **info about a property** (description, pictures, customer ratings, booking	OTA_HotelDescriptiveInfoRQ	OTA_HotelDescriptiveInfoRS

Procedure	***Request message***	***Response message***
requirements, etc.)		
Request for **details on a reservation**	OTA_ReadRQ	OTA_ResRetrieveRS
Reservation **cancellation**	OTA_CancelRQ	OTA_CancelRS
Ping	OTA_PingRQ	OTA_PingRS

As you see, the first three steps are covered by the same message pairs. We will describe in detail how to differentiate between the three uses of the message.

11. Messages

We'll start describing the usage of the simpler message types, then proceed to those that concern themselves with the reservation procedure.

In general, we won't describe details of messages that are self-explanatory.

Most of the request/reply messages you will find in this guide will have *Target* attributes set to "Test", not "Production". You must make sure that you request has the appropriate value, depending on whether you're making a test or a production request. Always check the value of the *Target* attribute in our response to make sure you've proceeded with the intended modality.

11.1. The OTA_PingRQ/RS

This message pair is the simplest of them all, and it is meant for applications to check if the other end is live and able to receive messages. This could easily be done by sending any other message and seeing if it gets a reply, but we've decided to implement this message because it's useful if you want to start implementing your application using the most basic of messages first, then proceeding to the more sophisticated ones.

The OTA_PingRQ looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_PingRQ xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05
OTA_PingRQ.xsd" TimeStamp="2003-03-17T11:09:47-05:00"
Target="Production" Version="1.001" PrimaryLangID="en"
EchoToken="testtoken12">
<EchoData>Hello</EchoData>
</OTA_PingRQ>
```

This message contains just an <EchoData> tag with a string content. The reply will look like this:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<OTA_PingRS xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05
OTA_PingRS.xsd" TimeStamp="2007-01-10T10:46:31-05:00"
Version="1.001" PrimaryLangID="en" EchoToken="testtoken12"
Target="Test">
  <Success/>
  <EchoData>Hello</EchoData>
</OTA_PingRS>
```

As you can see, the system returned the <EchoData> tag as it was, as well as the EchoToken contained in the request.

The OTA_PingRQ message is the only message that does not require authentication data to be passed along in the XML.

11.2. The OTA_HotelSearchRQ/RS

This messages are used to perform "static" search of properties within a city. This means that you are not passing along any dates for availability, and the system will return all properties that correspond to the search criteria you indicated.

11.2.1. The OTA_HotelSearchRQ message

The basic request will look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelSearchRQ xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05
OTA_HotelSearchRQ.xsd" EchoToken="HL"
Target="Production" Version="1.003" PrimaryLangID="en"
MaxResponses="2">
<POS>
<Source>
<RequestorID ID="1" MessagePassword="password" />
</Source>
</POS>
<Criteria AvailableOnlyIndicator="true">
<Criterion>
<Address>
      <CountryName Code="US">United States</CountryName>
      <CityName>New York</CityName>
      <StateProv StateCode="NY">New York</StateProv>
</Address>
<HotelRef HotelCityCode="194" SegmentCategoryCode="2"
PropertyClassCode="20" />
<TPA_Extensions>
      <OrderBy Value="rating"/>
</TPA_Extensions>
</Criterion>
</Criteria>
</OTA_HotelSearchRQ>
```

Please note that the message above contains a duplicate way of identifying the city to look in: we've shown both ways in the message so you know where the tags/attributes need to be located.

11.2.1.1. Root element attributes

This message contains only one message-specific root element attribute: *MaxResponses*, which tells our programming how many responses to send. If it is omitted or left empty, we will return a maximum of 100 properties for each request.

11.2.1.2. The main request element

This message has one main request element, the <Criteria> tag.

The <Criteria> tag can contain an attribute, *AvailableOnlyIndicator*. If it is set to "true" our system will return only establishments that have availability in the next future. Some establishments might not have availability at the moment, maybe because they're closed for season, or they're under renovation, etc. If this attribute is not present, we will return all establishments that match the search Criteria.

11.2.1.3. The <Criterion> tag

The <Criterion> tag is contained in the <Criteria> tag, and contains the actual information for your hotel search: city, ordering and sorting requirements.

Please note that the OTA Specs allow for more than one Criterion tag to be sent with each request. For performance reasons, we will consider only the first Criterion tag you sent. You can make multiple searches by performing multiple calls with different criteria.

11.2.1.4. Identifying the city

The city you want to search properties for can be indicated in two ways:

1. In the <HotelRef> tag, using the *HotelCityCode* attribute. This code is our system's city code, which can be found in the affiliate XML data (see Appendix). It is the simplest way of referencing a city, and should be preferably used instead of the other one below. This method always has a precedence to the one below, meaning that if you send it it will be looked at first, and all other city-related information will be disregarded.

2. By using the <Address> tag: you can send the following data in this tag:

```
<CountryName Code="US">USA</CountryName>
<CityName>New York</CityName>
<StateProv StateCode="NY">New York</StateProv>
```

The <CountryName> *Code* attribute is the standard ISO Country Code (we've provided them in the XML data): if this is sent, then the content of the tag is disregarded. If this is not sent, we will look at the content of the tag (in the case above, USA). It is preferable to use the ISO Country Code, as it is univocal. You must send either the *Code* attribute or the name of the country in the tag content.

The <CityName> tag, with the city name in the content part, is compulsory.

The <StateProv> tag is optional, and it is strongly recommended for US states, since the US have very many cities with the same name in different states (think of Springfield, which exists in at least 34 states) As for with the country code, you can send the *StateCode* attribute, which will be used instead of the content of the <StateProv> if it is present.

In this case, our system will try and search a city that matches the data above. To maximize results, the system will search the names of Countries, Cities and States in our databases **both** in the message language and in English.

If no city is found, or more than one city is found, the system will return a warning without results, as we have no way of determining what actual city you meant.

We'd like to point out again that the <Address> tag information will be ignored if you passed a *HotelCityCode* attribute in the <HotelRef> tag.

11.2.1.5. Other search criteria

You can choose to receive results only on some types of properties, which are selected using the *SegmentCategoryCode* and *PropertyClassCode* attributes of the <HotelRef> tag. The relevant codes are in the Appendix of this document.

The *SegmentCategoryCode* is used to select only properties which fall into one **price-level categories**, regardless of the type of property. For example, if you make a search using *SegmentCategoryCode* 2 which is the OTA Code for "Budget", we will return Hostels, Guest Houses, B&Bs, Campings and 1 Star Hotels.

If you send code 17 for "Midscale", we will return 2 Star Hotels and 3 Star Hotels, and so on.

The *PropertyClassCode* can be used when you want just properties of a specific **type of property**. If you send 20 for "Hotel", we will only return hotels, excluding hostels and B&Bs. The relevant codes are in the Appendix of this document.

If you send both attributes, the system will try to satisfy both requirements. The sample request, searching for "Budget" *SegmentCategoryCode* and "Hotel" *PropertyClassCode* will only return 1 Star Hotels. If there's no overlap of the two categories (ex. Luxury Segment and Hostels), the system won't return properties.

11.2.1.6. Ordering results

We have added a custom way of pre-ordering results. Since our ordering functions are very quick, you can easily pass on the <TPA_Extensions> tag with an <OrderBy> tag in it:

```
<TPA_Extensions>
     <OrderBy Value="rating"/>
</TPA_Extensions>
```

The accepted value for the *Value* attribute of the <OrderBy> tag are "rating" and "price".

The "rating" value will show properties with a higher customer rating first – customer rating is the rating provided by our customers through our rating system, and is a proof of customer satisfaction levels. Your customers will feel encouraged to book properties with a higher fellow customer appreciation.

The "price" value will show properties with a lower minimum price first. The price range for our properties is calculated regularly based on actual availability.

11.2.1.7. Criteria summary

With the options above, you can decide, for example, to get just the 10 budget hotels with the best customer rating. Or to get the 20 budget properties with the lowest prices.

The OTA Specs allow for more criteria, which might be implemented in future versions of our Service.

11.2.2. The OTA_HotelSearchRS message

The reply will look like this (we've removed part of the establishment descriptions, which are usually longer than one paragraph):

```
<?xml version="1.0" encoding="utf-8"?>
<OTA_HotelSearchRS xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05
OTA_HotelSearchRS.xsd" TimeStamp="2007-01-10T12:15:01-05:00"
Version="1.006" PrimaryLangID="it" EchoToken="HL" Target="Test">
  <Success/>
  <Properties>
    <Property HotelCode="5255" HotelCityCode="194"
HotelName="Locanda Ca' Foscari" HotelCodeContext="HCL"
HotelSegmentCategoryCode="2">
      <VendorMessages>
        <VendorMessage InfoType="1" Language="en"
Title="Descrizione">
          <SubSection>
            <Paragraph>
              <Text Language="en">The hotel is located between
Campo Santa Tomà and Campo Santa Margherita, one of the most
interesting areas of Venice.Bars, restaurants and a lot of shops
are in the area,the night life in Venice is just 2 minutes walk
from   our Hotel.  The rooms are well cared for and kept up on a
regular maintenance schedule.Breakfast,taxes are included
</Text>
            </Paragraph>
          </SubSection>
        </VendorMessage>
      </VendorMessages>
      <Position Latitude="45.435291798966"
Longitude="12.3464012146"/>
      <Address>
        <AddressLine>Calle della Frescada- Dorsoduro
3887/B</AddressLine>
        <CityName>Venezia</CityName>
        <CountryName Code="IT">Italia</CountryName>
      </Address>
      <Amenities>
        <Amenity RoomAmenity="20">TV</Amenity>
        <Amenity RoomAmenity="88">Frigo</Amenity>
        <Amenity PropertyAmenityType="242">Riscaldamento</Amenity>
        <Amenity PropertyAmenityType="218">Accettiamo
bambini</Amenity>
        <Amenity PropertyAmenityType="224">Accettiamo
animali</Amenity>
        <Amenity PropertyAmenityType="77">Servizio in
camera</Amenity>
        <Amenity PropertyAmenityType="223">Accesso
internet</Amenity>
        <Amenity RoomAmenity="189">Colazione</Amenity>
        <Amenity PropertyAmenityType="78">Cassette di
sicurezza</Amenity>
      </Amenities>
      <RateRange MinRate="40" MaxRate="120" CurrencyCode="EUR"/>
      <TPA_Extensions>
        <ThumbPicture Url="www.hostelspoint.com/photos/th-005255-
```

```
1167406466.jpg"/>
        <FullSizePicture Url="www.hostelspoint.com/photos/005255-
1167406466.jpg"/>
        <CustomerRatings>
          <Rating Type="overall" Value="90" NumberRatings="4"
Base="100" Description="totale"/>
          <Rating Type="cleanliness" Value="99" Base="100"
Description="pulizia"/>
          <Rating Type="staff" Value="85" Base="100"
Description="staff"/>
          <Rating Type="position" Value="90" Base="100"
Description="posizione"/>
          <Rating Type="fun_factor" Value="80" Base="100"
Description="divertimento"/>
          <Rating Type="personality" Value="90" Base="100"
Description="personalit&#xE0;"/>
        </CustomerRatings>
        <HotelCategory Code="20"/>
        <TextCategory>Albergo *</TextCategory>
      </TPA_Extensions>
    </Property>
    <Property HotelCode="1661" HotelCityCode="194"
HotelName="Hotel Caneva" HotelCodeContext="HCL"
HotelSegmentCategoryCode="2">
      <VendorMessages>
        <VendorMessage InfoType="1" Language="en"
Title="Descrizione">
          <SubSection>
            <Paragraph>
              <Text Language="en">As the hotel also has an
entrance on one of Venice's best known canals, it is possible to
reach it directly by water taxi or gondola. Some rooms are
overlooking the canal and others face Faccanon palace where also
the famous Giacomo Casanova once lived.[...]</Text>
            </Paragraph>
          </SubSection>
        </VendorMessage>
      </VendorMessages>
      <Address>
        <AddressLine>Campo della Fava  5515</AddressLine>
        <CityName>Venezia</CityName>
        <CountryName Code="IT">Italia</CountryName>
      </Address>
      <RateRange MinRate="50" MaxRate="150" CurrencyCode="EUR"/>
      <TPA_Extensions>
        <ThumbPicture Url="www.hostelspoint.com/photos/th-001661-
1143530274.jpg"/>
        <FullSizePicture Url="www.hostelspoint.com/photos/001661-
1143530274.jpg"/>
        <CustomerRatings>
          <Rating Type="overall" Value="84" NumberRatings="53"
Base="100" Description="totale"/>
          <Rating Type="cleanliness" Value="89" Base="100"
Description="pulizia"/>
          <Rating Type="staff" Value="88" Base="100"
Description="staff"/>
          <Rating Type="position" Value="92" Base="100"
Description="posizione"/>
          <Rating Type="fun_factor" Value="74" Base="100"
```

```
Description="divertimento"/>
          <Rating Type="personality" Value="80" Base="100"
Description="personalit&#xE0;"/>
        </CustomerRatings>
        <HotelCategory Code="20"/>
        <TextCategory>Albergo *</TextCategory>
      </TPA_Extensions>
    </Property>
  </Properties>
</OTA_HotelSearchRS>
```

11.2.2.1. The main response node

The main response node for this message is the <Properties> tag. If present, it will contain a series of individual <Property> tags, each containing the description of a property.

The <Property> tag has the following attributes:

Attribute	***Value***
HotelCode	Numeric Hostelsclub code for property
HotelCityCode	Numeric Hostelsclub code for city
HotelName	Property name
HotelCodeContext	Always the HCL value, to indicate that we use our internal code for the HotelCode attribute
HotelSegmentCategoryCode	The SegmentCategoryCode – see the criteria above. In case you're wondering, the PropertyClassCode, according to current OTA Specs, cannot be put here, so you'll find it in the Extensions

11.2.2.2. <Property> tag subnodes

The first subnode is the <VendorMessages> tag, which contains (enclosed in a few more tags) the establishment description. If possible, the description is provided in the language request. If the description has not been translated yet, it will be provided in English. The description will never contain HTML tags, but only UTF-8 plain text with line breaks. You will see from the example above that the two-letter language code of the description is present in two attributes above. The "Descrizione" value for the *Title* attribute of the <Vendor Message> tag is Italian for "Description" (remember, the call was made with "it" – for Italian – as a Primary Language). This way systems which automatically output all vendor messages can output what the actual content of the message is, in the language of the customer viewing the page.

The <Positon> tag contains the geocoding coordinates of the property (data is available for a growing number of properties – if no geocoding data is available, the tag is omitted).

The <Address> tag contains the address of the property.

The <RateRange> tag contains the maximum and minimum price in EUR. Note that this may reflect, at the two opposites, the minimum price for a bed in a shared room (dormitory) and the maximum price of a 4-bed private room. At this level, prices will always be in EUR and won't be converted in other currencies by our system.

The <TPA_Extensions> tag contains several custom tags:

- the <ThumbPicture> and <FullSizePicture> tags, with links to thumbnail-size and full-size versions of the establishment's main picture (without http:// or https:// protocol in the url – you need to add the one you need to use in that particular context)
- the <Customer Ratings>, containing the various customer ratings for the establishment, as in our public site. Each rating has a textual description of the characteristic rated in the Primary Language of the request (so you can output it). The *NumberRatings* attribute of the overall rating is the number of customers who have rated this establishment.
- The <HotelCategory> tag contains the PropertyClassCode of the property, while the <TextCategory> contains the property category in textual value, in the Primary Language of the request.

11.3. The OTA_HotelDescriptiveInfoRQ/RS

This message pair is used to retrieve general info about a property. It is flexible, so it allows you to retrieve only the type of info you need at the specific moment.

The info you can request currently includes:

- general property information (name, category, description, services offered by the property)
- property rooms (though this will only return rooms inserted in our booking system)
- property address
- booking policies and booking restrictions
- multimedia objects (images)
- ratings and reviews

11.3.1. The Ota_HotelDescriptiveInfoRQ

The request message looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelDescriptiveInfoRQ EchoToken="1" Target="Production"
Version="1.006"
xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05
OTA_HotelDescriptiveInfoRQ.xsd"  PrimaryLangID="en">
<POS>
<Source>
<RequestorID ID="1" MessagePassword="password" />
</Source>
</POS>
<HotelDescriptiveInfos>
<HotelDescriptiveInfo HotelCode="958">
<HotelInfo SendData="true"/>
<FacilityInfo SendGuestRooms="true" />
<Policies SendPolicies="true"/>
<ContactInfo SendData="true"/>
<MultimediaObjects SendData="true"/>
<ContentInfos>
```

```
<ContentInfo Name="CustomerRatings"/>
<ContentInfo Name="CustomerReviews"/>
</ContentInfos>
</HotelDescriptiveInfo>
</HotelDescriptiveInfos>
</OTA_HotelDescriptiveInfoRQ>
```

11.3.1.1. The main request element

The main request element for this message is <HotelDescriptiveInfos>. The OTA Standard allows to request info about multiple properties using multiple <HotelDescriptiveInfo> tags contained in the main request element. We will reply to just the first tag found. If you need info on multiple properties, you should perform multiple calls.

To identify the property you're requesting info for, you must pass the Hostelsclub property ID (retrieved from static XML files or from Search/Availability requests on a city) as an attribute *HotelCode* of the <HotelDescriptiveInfo> tag.

The above example contains a request for all information on an establishment that is currently available:

```
<HotelInfo SendData="true"/>
```

requests the establishment's general information.

```
<FacilityInfo SendGuestRooms="true" />
```

requests information about the establishment's rooms

```
<Policies SendPolicies="true"/>
```

requests information about booking policies and special booking requirements: some of our establishments have limitations on weekend reservations (for example, a customer must book Friday and Saturday together), the minimum or maximum number of guests per reservation, and have special limitations for some periods of the year (for example, minimum of 4 nights for reservation including New Year's Eve), etc. We will look at those in detail when we will analyse the response.

```
<ContactInfo SendData="true"/>
```

requests property address information. We will provide only street address, city etc., not telephone or e-mail.

```
<MultimediaObjects SendData="true"/>
```

will return the property's image gallery, sending both addresses for thumbnail and full-size images.

```
<ContentInfos>
<ContentInfo Name="CustomerRatings"/>
<ContentInfo Name="CustomerReviews"/>
</ContentInfos>
```

the <ContentInfo> tags (you can put either or both in the <ContentInfos> container tag) will return the property's ratings and the latest customer reviews. Please note that reviews can be in languages other than the one you've requested.

11.3.2. The OTA_ HotelDescriptiveInfoRS

The tags content of this message will depend on what you've requested using the options above. Here's a sample response message for an establishment which has got a farily complete data set.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<OTA_HotelDescriptiveInfoRS
xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05
OTA_HotelDescriptiveInfoRS.xsd" TimeStamp="2007-01-12T10:22:28-
05:00" Version="1.006" PrimaryLangID="en" EchoToken="1"
Target="Test">
<Success/>
<HotelDescriptiveContents HotelCode="958" HotelCodeContext="HCL"
HotelName="Amstel Inn Tourist Accommodation" HotelCityCode="224">
<HotelDescriptiveContent HotelCode="958" HotelCodeContext="HCL"
HotelName="Amstel Inn Tourist Accommodation" HotelCityCode="224"
CurrencyCode="EUR">
<HotelInfo>
<HotelName>Amstel Inn Tourist Accommodation</HotelName>
<CategoryCodes>
<SegmentCategory Code="2"/>
<HotelCategory Code="16" CodeDetail="Guest House"/>
<GuestRoomInfo>
<RateRanges>
<RateRange MinRate="38.00" MaxRate="120.00" CurrencyCode="EUR"
DecimalPlaces="2"/>
</RateRanges>
</GuestRoomInfo>
</CategoryCodes>
<Descriptions>
<Description ContentCode="5" Language="en">
<Text Language="en">We have one small and one grande room. The
rate of the small room is lower then the grande.The rooms have
shared facilities.

At walking distance you have Rembrandt Square, The Famous
shoppingstreet Kalverstraat, Albertcuyp market,  Flower market and
more.

All the rooms are non-smoking.

The rooms could be double or twin.

</Text>
</Description>
<Description ContentCode="4" ContentFormatCode="5">
<URL>www.hostelspoint.com/photos/th-000958-1145361229.jpg</URL>
</Description>
</Descriptions>
<Position Latitude="45.435291798966" Longitude="12.3464012146"/>
<Services>
<Service Code="242">
<Description Language="en">
<Text Language="en">Heating</Text>
</Description>
</Service>
<Service Code="218">
<Description Language="en">
<Text Language="en">Children accepted</Text>
</Description>
</Service>
</Services>
</HotelInfo>
```

```
<FacilityInfo>
<GuestRooms>
<GuestRoom Code="1_4" CodeContext="HCL" Quantity="3"
RoomTypeName="private" MaxOccupancy="2">
<TypeRoom Name="2 bed/s private " RoomType="1_4"
StandardNumBeds="2" RoomClassificationCode="19"/>
</GuestRoom>
</GuestRooms>
</FacilityInfo>
<Policies>
<Policy DefaultValidBookingMinOffset="1">
<CancelPolicy>
<CancelPenalty>
<Deadline OffsetDropTime="BeforeArrival" OffsetTimeUnit="Day"
OffsetUnitMultiplier="2"/>
<AmountPercent NmbrOfNights="1"/>
</CancelPenalty>
</CancelPolicy>
<GuaranteePaymentPolicy>
<GuaranteePayment GuaranteeType="PrePay" NameInd="1"
PaymentCode="5" NonRefundableIndicator="1">
<AmountPercent FeesInclusive="false" Percent="10"/>
</GuaranteePayment>
</GuaranteePaymentPolicy>
<StayRequirements>
<StayRequirement Fri="1" Sat="1" Sun="1">
<Description Language="en">
<Text>You must always book &lt;strong&gt;friday, saturday and
sunday&lt;/strong&gt; nights together</Text>
</Description>
</StayRequirement>
</StayRequirements>
<FeePolicies>
<FeePolicy Amount="2" CurrencyCode="EUR" ChargeFrequency="26"
Code="14" MandatoryIndicator="true"/>
</FeePolicies>
</Policy>
<Policy DefaultValidBookingMinOffset="1" Start="2007-04-01"
End="2007-04-05">
<CancelPolicy>
<CancelPenalty>
<Deadline OffsetDropTime="BeforeArrival" OffsetTimeUnit="Day"
OffsetUnitMultiplier="2"/>
<AmountPercent NmbrOfNights="1"/>
</CancelPenalty>
</CancelPolicy>
<GuaranteePaymentPolicy>
<GuaranteePayment GuaranteeType="PrePay" NameInd="1"
PaymentCode="5" NonRefundableIndicator="1">
<AmountPercent FeesInclusive="false" Percent="10"/>
</GuaranteePayment>
</GuaranteePaymentPolicy>
<StayRequirements>
<StayRequirement MinLOS="2">
<Description Language="en">
<Text>you must book at least 2 nights</Text>
</Description>
</StayRequirement>
</StayRequirements>
```

```
<FeePolicies>
<FeePolicy Amount="2" CurrencyCode="EUR" ChargeFrequency="26"
Code="14" MandatoryIndicator="true"/>
</FeePolicies>
</Policy>
</Policies>
<MultimediaObjects>
<MultimediaObject ContentCode="4" ContentFormatCode="5"
MultimediaObjectHeight="80" MultimediaObjectWidth="80"
Version="Thumbnail">
<URL>www.hostelspoint.com/photos/th-000958-1145361229.jpg</URL>
</MultimediaObject>
<MultimediaObject ContentCode="4" ContentFormatCode="5"
Version="Fullsize">
<URL>www.hostelspoint.com/photos/000958-1145361229.jpg</URL>
</MultimediaObject>
</MultimediaObjects>
<ContactInfos>
<ContactInfo Location="6">
<Addresses>
<Address>
<AddressLine>Hemonystraat 7</AddressLine>
<CityName>Amsterdam</CityName>
<PostalCode>1018 EP</PostalCode>
<StateProv StateCode="">North Holland</StateProv>
<CountryName Code="NL">Netherlands</CountryName>
</Address>
</Addresses>
</ContactInfo>
</ContactInfos>
<TPA_Extensions>
<CustomerRatings>

          <Rating Type="overall" Value="95" NumberRatings="9"
Base="100" Description="overall rating"/>
          <Rating Type="cleanliness" Value="97" Base="100"
Description="cleanliness"/>
          <Rating Type="staff" Value="97" Base="100"
Description="staff"/>
          <Rating Type="position" Value="91" Base="100"
Description="position"/>
          <Rating Type="fun_factor" Value="94" Base="100"
Description="Value"/>
          <Rating Type="personality" Value="94" Base="100"
Description="Recommended"/>
</CustomerRatings>
<CustomerReviews>
<Review CustomerName="Raphael" CustomerNationality="France"
Date="2010-01-08T13:16:51-05:00" Rating="100">Excellente adresse
bien située au centre de Cagliari. Le personnel, parlant anglais,
s'est révélé précieux par son attention et ses informations. Le
Corso Vittorio Emmanuel, rue sur laquelle donnent certaines
chambres, peut se révéler bruyante jusque tard dans la nuit, en
particulier pendant les périodes festives (jour de l'an pour nous)
mais ce défaut devrait être corrigé dans l'année par des travaux
prévus sur les fenêtres. A part ce détail, bientôt corrigé, cette
adresse est à recommander. Meilleurs voeux / Auguri</Review>
</CustomerReviews>
</TPA_Extensions>
```

```
</HotelDescriptiveContent>
</HotelDescriptiveContents>
</OTA_HotelDescriptiveInfoRS>
```

Let's look at the details. The response contains a <HotelDescriptiveContents> tag that contains a <HotelDescriptiveContent> tag with the info on the property. Both tags have a similar set of attributes (they have been duplicated since OTA Specs allow them to be on both tags):

Attribute	***Value***
HotelCode	The hotel code in our system
HotelCodeContext	Always HCL – for Hostelsclub
HotelName	The name of the establishment
HotelCityCode	The city code for the establishment's city
CurrencyCode	The currency the establishment uses internally (only present on the HotelDescriptiveContent> tag)

11.3.2.1. Subnodes: the <HotelInfo> tag

This tag contains the information about the hotel, in the following tags:

- <HotelName> - the name of the hotel again. Here it is in the tag content
- <CategoryCodes> - the categories to which the property belongs. There are two subtags: one is the <SegmentCategory> tag, whose *Code* attribute contains the numerical code for the establishment's segment; the other is the <HotelCategory> tag, which contains a *Code* attribute with the property code, and a *CodeDetail* attribute containing the textual property type (for ex. Hotel, B&B etc.) in the Primary Language of the response.
- The <GuestRoomInfo> tag contains a <RateRanges> tag which contains a single <RateRange> tag with minimum and maximum prices and currency for the prices. These are price taken from actual availability; remember that the minimum price can be the price for one bed in a shared room, and the maximum price can be the price for a 4-bed private room.
- The <Descriptions> tag contains, nested in another couple of tags, the establishment description – there are language attributes to figure out what language the text is actually in. There is usually another <Description> tag containing the main thumbnail image for the establishment.
- The <Position> tag contains the geocoding coordinates of the property (data is available for a growing number of properties – if no geocoding data is available, the tag is omitted).
- The <Services> tag contains a list of Services offered in the establishment, with an OTA Code and a textual description in the desired language.

11.3.2.2. The <FacilityInfo> tag

This tag contains the rooms inserted in our system: they are contained in a <GuestRooms> tag, and each room type is contained in a <GuestRoom> tag with the following attributes:

Attribute	***Value***
Code	The Hostelsclub code for rooms. It consist in a number (1 or 2), followed by an underscore and another number. The 1 stands for private room, 2 for shared room (dormitory). Please check the Appendix for a list of codes.
CodeContext	HCL indicating that the previous code is our internal code
Quantity	The number of rooms present in our database
RoomTypeName	Whether the room is private or shared, in the request language
MaxOccupancy	Maximum number beds in the room

The <GuestRoom> contains a <TypeRoom> tag, which contains the following attributes:

Attribute	***Value***
Name	Complete room name in the request language
RoomType	Our internal Room Code (See *Code*) above
StandardNumBeds	The number of beds in the room
RoomClassificationCode	The type of room according to the OTA Code List Guest Room Info (GRI). (See Appendix for codes)

11.3.2.3. The <Policies> tag

This returns the booking policies of the establishment. Please remember that the deposit, fee and cancellation policy for our system are always the same for each establishment at the moment:

We require a deposit of 10% on the total cost of the reservation.

There is a 2€ booking fee for each reservation.

The cancellation must take place 24h before the reservation starts (which is in fact two days before the reservation), otherwise the establishment is authorized to charge one night's stay per person.

Each <Policy> tag contains that information (repeated in each tag for the convenience of systems which already parse all policy tags and output separated policy descriptions), plus the following:

- All <Policy> tags contain the *DefaultValidBookingMinOffset* attribute, which indicates the minimum number of days one must book in advance. This can be a value of 1 or higher. Our system currently doesn't permit same-day reservations.
- The **first** policy tag can contain the **general** <StayRequirements> for an establishments: some properties have specific stay requirements concerning weekends or minimum length of stay. In the first case, you will find a <StayRequirement> tag with attributes for each day of the weekend that must be booked together, all set to 1. This can be *Fri* and *Sat*, *Sat* and *Sun*, or *Fri*, *Sat* and *Sun*. In the latter case, the <StayRequirement> tag will have a *MinLOS* (Minimum Length of Stay) set to the minimum number of nights one must book. Both tags also

contain a text description of the limitation in the requested language, included within a <Description> and a <Text> tag.

- The **second and following tags** (the example has one of them) are for period-related restrictions, and carry a *Start* and *End* attributes, both in the YYYY-MM-DD format: these are the start and end date of the restriction, and the *MinLOS* attribute of the single <StayRequirement> tag present will contain the minimum length of stay in nights. The restriction can coincide with the whole period from *Start* to *End*, or just require that a minimum number of days must be booked during that period. In the first case, the *MinLOS* attribute will have a value which is equal to the number of nights from *Start* to *End*, otherwise it will be a number lower than that. The <Description> of the restriction will contain a different message: either "You must book the whole period" or "You must book a minimum of X nights".

The restrictions are enforced at the room choice level in our system. If the request for rooms does not match the requirements, the system produces an error message and refuses to go on with the reservation. It is recommended to make a Ota_HotelDescriptiveInfoRQ containing <Policies SendPolicies="true"/> every time your users are at the "choose rooms in an establishment" step and output the descriptions of the restrictions (and the dates, for period- related restrictions) before the room choice system. The OTA message which outputs Room Availability does not allow to show this level of detail for restrictions.

11.3.2.4. The <MultimediaObjects> tag

This tag contains pictures of the establishment, always in couples: one for the Thumbnail of the image, and one for the Full-size version. All of our system's thumbnails are 80x80 pixels, while size of the Full-size version will vary.

11.3.2.5. The <ContactInfos> tag

This tag always contains the address of the establishment, divided in street, city, country and (where applicable) <StateProv>, which will contain the state (for US and other federal states) or the Province/Region (depending on country) where the city is located, complete with State/Province/Region code if available.

11.3.2.6. The <TPA_Extensions> tag

This will contain, if you have requested them, the <CustomerRatings> (see OTA_HotelSearchRS above) for the establishments, and the 10 most recent reviews (if available) in the <CustomerReviews> tag. Each review is contained in a <Review> tag with the following attributes:

Attribute	***Value***
CustomerName	The first name of the customer
CustomerNationality	The nationality of the customer: this is a text attribute, and it's always in English
Date	This is the date the review was inserted
Rating	The overall rating (with a maximum score of 100) given with this review.

The review is in the content of the tag. The review language can be any language, while most of them are in English.

11.4. The OTA_HotelAvailRQ/RS

As we've previously noted, this message is used for three purposes in our implementation:

1. **Search** – in which the customer searches for properties in a determined city with have availability on certain dates
2. **Select** - in which the customer selects rooms from a property with availability on those dates
3. **Book** – in which the customer is assigned the final room choice and inserts his/her personal data and credit card data

The Ota_HotelAvailRQ adds a specific attribute on the root element: the *RequestedCurrency* attribute, which must contain a standard three-letter ISO 4217 currency abbreviation. See Appendix for code details. All prices will be converted in the currency requested, while deposits and fees will still be requested and charged in EUR, with and indication of the equivalent amount in the request currency. If the requested currency is not supported by our system, it will issue a warning and switch to EUR. We support most world currencies, however.

We will examine how to differentiate between the three request types:

11.4.1. Property availability search: the Search OTA_HotelAvailRQ

This request will return a list of properties available in a city for the requested dates. Here's a sample request:

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelAvailRQ xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05
OTA_HotelAvailRQ.xsd" Version="1.006" RequestedCurrency="USD"
PrimaryLangID="de" SortOrder="rating" AllowPartialAvail="false">
<POS>
<Source>
<RequestorID ID="1" MessagePassword="pwd" />
</Source>
</POS>
<AvailRequestSegments>
<AvailRequestSegment>
<HotelSearchCriteria>
<Criterion>
<StayDateRange Start="2007-02-12" End="2007-02-16" />
<Address>
      <CountryName Code="IT"></CountryName>
      <CityName>Venice</CityName>
</Address>
<HotelRef  SegmentCategoryCode="2" PropertyClassCode="20" />
</Criterion>
</HotelSearchCriteria>
</AvailRequestSegment>
</AvailRequestSegments>
</OTA_HotelAvailRQ>
```

You can see immediately that it has some similarities with the OTA_HotelSearchRQ.

You need to send a <Criterion> tag with the search criteria desired. In this case, the <Criterion> tag is enclosed in a <HotelSearchCriteria>, which is contained in a <AvailRequestSegment> tag, which is contained in a <AvailRequestSegments> tag. As for the OTA_ HotelSearchRQ, we will only process the first <Criterion> tag we receive.

As in the OTA_HotelSearchRQ, you need to send proper means of identifying the city you want to find properties in, either by sending the Hostelsclub city code in the *HotelCityCode* of the <HotelRef> tag, or by sending city name and country data in the <Address> tag.

The ordering functions, which in the OTA_HotelSearchRQ had to be passed via the <TPA_Extensions>, can be passed here using the OTA standard *SortOrder* attribute of the root element. We accept "rating" and "price" as values.

You can pass on the *SegmentCategoryCode* and the *PropertyClassCode* attributes in the <HotelRef> tag to select types of properties, just like in the OTA_HotelSearchRQ.

Some properties might not have complete availability for the period requested: you can avoid receiving this properties in the response by setting the root element *AllowPartialAvail* attribute to "false", as in the example. This way, all properties in the response will have full availability for the period requested. If this attribute is absent or set to "true", then you the response will contain all properties which have availability on the days requested. Properties with full availability will be listed first, if you haven't chosen an ordering method.

11.4.1.1. The <StayDateRange> tag

This tag is required in all uses of the OTA_HotelAvailRQ tag. It is used to indicate the period in which you want to perform a search.

It must contain a *Start* attribute with the start date of the search in ISO format, which is yyyy-mm-ddThh:mmZ - we will only take into consideration the date part of the data, so you can omit everything from the T onwards, although our system will parse the full date format correctly.

The **End** of the period can be indicated in two ways: either by sending an end (departure /check-out date) date in the *End* attribute of the <StayDateRange> tag, or by sending the number of nights the customer wants to stay in the *Duration* attribute of the <StayDateRange> tag. According to the OTA Standards, the duration must be expressed as follows: P([0-9]{1,3})N, which means that for a duration of 3 nights you must send P3N.

If you send the *End* attribute, remember that our system considers that the **departure/check-out date**, so a request from Feb 12th to Feb 16th will search for availability of a room from the nights of the 12th to the night of the 15th (a total of 4 nights).

If you send both the *Duration* and the *End* attributes, the *End* attribute will have precedence over the *Duration* attribute. So if you send a tag that looks like this:

```
<StayDateRange Start="2007-02-12" End="2007-02-16"
Duration="P7N"/>
```

the system will consider it a search for 4 nights (from the 12th to the 16th excluded).

Important: currently, searches and reservation in our system are restricted to **a maximum of 7 nights per reservation**. Any search for a longer period **will issue an error**.

11.4.2. Property availability search: the Search OTA_HotelAvailRS

Here's a sample response for a Property Availability Search:

```
<?xml version="1.0" encoding="utf-8"?>
```

```xml
<OTA_HotelAvailRS xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05
OTA_HotelAvailRS.xsd" TimeStamp="2007-01-15T10:23:27-05:00"
Version="1.006" PrimaryLangID="de" EchoToken="" Target="Test">
<Success/>
<RoomStays>
<RoomStay>
<RoomRates>
<RoomRate>
<Rates>
<Rate RateTimeUnit="Day">
<Base AmountBeforeTax="45.14" CurrencyCode="USD"/>
<RateDescription>
<Text Language="de">Privatzimmer ab 45.14 USD (pro Zimmer)</Text>
</RateDescription>
</Rate>
</Rates>
</RoomRate>
<RoomRate>
<Rates>
<Rate RateTimeUnit="Day">
<Base AmountBeforeTax="25.80" CurrencyCode="USD"/>
<RateDescription>
<Text Language="de">Mehrbettzimmer ab 25.80 USD (pro
Person)</Text>
</RateDescription>
</Rate>
</Rates>
</RoomRate>
</RoomRates>
<BasicPropertyInfo HotelCode="1174" HotelCityCode="194"
HotelName="CaSa Linger" HotelCodeContext="HCL"
HotelSegmentCategoryCode="2">
<VendorMessages>
<VendorMessage InfoType="1" Language="en" Title="Beschreibung">
<SubSection>
<Paragraph>
<Text Language="en">Ca'Sa Linger is situated in the Heart of the
City Center, just minutes away from the beautiful San Marco
Square. We offer Perfect rooms at perfect prices! Some rooms are
furnished in Venetian Style, we offer rooms with private WC and
without. We are minutes away from all the important monuments of
the City and there is a Big Supermarket meters away with
everything you will need.... Staff are very friendly and will help
you spend a relaxing time in the beautiful city of Venice. We
guarantee you a Good Time!</Text>
</Paragraph>
</SubSection>
</VendorMessage>
</VendorMessages>
<Position Latitude="45.435291798966" Longitude="12.3464012146"/>
<Address>
<AddressLine>Salizada Sant' Antonin Castello 3541</AddressLine>
<CityName>Venedig</CityName>
<CountryName Code="IT">Italien</CountryName>
</Address>
</BasicPropertyInfo>
<TPA_Extensions>
```

```
<ThumbPicture Url="www.hostelspoint.com/photos/th-001174-
1098877319.jpg"/>
<FullSizePicture Url="www.hostelspoint.com/photos/001174-
1098877319.jpg"/>
<CustomerRatings>
<Rating Type="overall" Value="75" NumberRatings="230" Base="100"
Description="Allgemein"/>
<Rating Type="cleanliness" Value="76" Base="100"
Description="Sauberkeit"/>
<Rating Type="staff" Value="75" Base="100"
Description="Personal"/>
<Rating Type="position" Value="87" Base="100" Description="Lage"/>
<Rating Type="fun_factor" Value="66" Base="100"
Description="Spa&#xDF;faktor"/>
<Rating Type="personality" Value="69" Base="100"
Description="Charakter"/>
</CustomerRatings>
<HotelCategory Code="20"/>
<TextCategory>Hotel *</TextCategory>
</TPA_Extensions>
</RoomStay>
<RoomStay>
<RoomRates>
<RoomRate>
<Rates>
<Rate RateTimeUnit="Day">
<Base AmountBeforeTax="38.69" CurrencyCode="USD"/>
<RateDescription>
<Text Language="de">Privatzimmer ab 38.69 USD (pro Zimmer)</Text>
</RateDescription>
</Rate>
</Rates>
</RoomRate>
<RoomRate>
<Rates>
<Rate RateTimeUnit="Day">
<Base AmountBeforeTax="25.80" CurrencyCode="USD"/>
<RateDescription>
<Text Language="de">Mehrbettzimmer ab 25.80 USD (pro
Person)</Text>
</RateDescription>
</Rate>
</Rates>
</RoomRate>
</RoomRates>
<BasicPropertyInfo HotelCode="3114" HotelCityCode="194"
HotelName="Hotel Giovannina" HotelCodeContext="HCL"
HotelSegmentCategoryCode="2">
<VendorMessages>
<VendorMessage InfoType="1" Language="de" Title="Beschreibung">
<SubSection>
<Paragraph>
<Text Language="de">Das Hotel Giovanna ist ein Ein-Strene-Hotel
und ist in der Naehe des Bahnhofs von Mestre-Venedig ( zwei
Minuten zu Fuss) gelegen. Wir bieten saubere, einfache Zimmer
einige mit privatem  Bad usw…Insgesamt haben wir 30 Zimmer, die
alle in gutter Verfassung und sehr sauber sind. Fuer Raucher haben
wir eine eigene Terasse.
</Text>
```

```
</Paragraph>
</SubSection>
</VendorMessage>
</VendorMessages>
<Position Latitude="45.435291798966" Longitude="12.3464012146"/>
<Address>
<AddressLine>Via Dante 113</AddressLine>
<CityName>Venedig</CityName>
<CountryName Code="IT">Italien</CountryName>
</Address>
</BasicPropertyInfo>
<TPA_Extensions>
<ThumbPicture Url="www.hostelspoint.com/photos/th-003114-
1121419399.png"/>
<FullSizePicture Url="www.hostelspoint.com/photos/003114-
1121419399.png"/>
<CustomerRatings>
<Rating Type="overall" Value="70" NumberRatings="114" Base="100"
Description="Allgemein"/>
<Rating Type="cleanliness" Value="74" Base="100"
Description="Sauberkeit"/>
<Rating Type="staff" Value="68" Base="100"
Description="Personal"/>
<Rating Type="position" Value="71" Base="100" Description="Lage"/>
<Rating Type="fun_factor" Value="59" Base="100"
Description="Spa&#xDF;faktor"/>
<Rating Type="personality" Value="60" Base="100"
Description="Charakter"/>
</CustomerRatings>
<HotelCategory Code="20"/>
<TextCategory>Hotel *</TextCategory>
</TPA_Extensions>
</RoomStay>
</RoomStays>
</OTA_HotelAvailRS>
```

This is a response for a request for Venice, Italy. This sample response has been trimmed to include two properties only. Establishments descriptions have been trimmed to one paragraph.

11.4.2.1. The main response node

This message has the <RoomStays> as a main response node. This contains as many <RoomStay> children tag as there are properties matching the request. The <RoomStay> tag is very similar to the <Property> tag in a OTA_HotelSearchRS, only that instead of the <RateRange> tag it features a <RoomRates> tag, containing one or two <RoomRate> tags. The rates correspond to Private and Shared (Dorm) rooms. The rates look like this:

```
<RoomRate>
<Rates>
<Rate RateTimeUnit="Day">
<Base AmountBeforeTax="38.69" CurrencyCode="USD"/>
<RateDescription>
<Text Language="de">Privatzimmer ab 38.69 USD (pro Zimmer)</Text>
</RateDescription>
</Rate>
</Rates>
```

```
</RoomRate>
<RoomRate>
<Rates>
<Rate RateTimeUnit="Day">
<Base AmountBeforeTax="25.80" CurrencyCode="USD"/>
<RateDescription>
<Text Language="de">Mehrbettzimmer ab 25.80 USD (pro
Person)</Text>
</RateDescription>
</Rate>
</Rates>
</RoomRate>
```

There is an XML-formatted set of data (the <Base> tag), and a formatted text in the <RateDescription> tag, which can be displayed directly to the end customer. The prices are indicative, and could not correspond to the actual prices found in the next step, for performance reasons.

11.4.2.2. How to proceed sending the customer to your personalized booking interface on our server

If you want to "exit" at this point, and let the customer proceed with the room choice using your personalized booking interface on our server, you must direct the customer to the following URL:

http://www.hostelspoint.com/res/*your_site_code_here*/step02_str.php

Where *your_site_code_here* stands for your site code (not your affiliate code).

If you want to direct users to our public Hostelsclub site, you must direct the customer to the following URL:

http://www.hostelsclub.com/step02_str.php?aff_ID=*your_affiliate_id_here*

Where *your_affiliate_id_here* stands for your affiliate code (not your site id). Please note that the variable name is aff_ID, with capital ID.

You also need to pass the following variables in GET:

Variable	**Value**
str	The propery ID, which you can find in the *HotelCode* attribute of the <BasicPropertyInfo> tag.
lang	The language for the interface. It must be our two-letter language code – see the language list in the Appendix
giorno	The day of the month, in numbers. Examples: 1, 17, 23. You can also send 01 instead of 1.
mese	The number of the month. Examples: 1, 2, 12. You can also send 01 instead of 1.
anno	The year in 4-figure format (ex. 2007)
notti	The number of nights (ex. 2)
currency	The customer chosen currency – you must use our internal currency ID. See the appendix below for currency codes and IDs.

A sample query string will look like this:

str=7153&lang=de&giorno=17&mese=1&anno=2007&notti=3&currency=1

11.4.3. Room selection in a property: the Select OTA_HotelAvailRQ

Once a customer has selected a property, you should display him/her the rooms available in a property. This must be done with a OTA_HotelAvailRQ which has a slightly different format than the previous one. The request will look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelAvailRQ xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05
OTA_HotelAvailRQ.xsd" Version="1.006" RequestedCurrency="USD"
PrimaryLangID="it" EchoToken="1" Target="production" >
<POS>
<Source>
<RequestorID ID="1" MessagePassword="pwd" />
</Source>
</POS>
<AvailRequestSegments>
<AvailRequestSegment>
<HotelSearchCriteria>
<Criterion>
<StayDateRange Start="2007-02-12" End="2007-02-15" />
<HotelRef HotelCode="637"/>
</Criterion>
</HotelSearchCriteria>
</AvailRequestSegment>
</AvailRequestSegments>
</OTA_HotelAvailRQ>
```

The root element is still the <AvailRequestSegments> tag, containing an <AvailRequestSegment> tag, a <HotelSearchCriteria> tag and a <Criterion> tag. Again, we will process only the first <Criterion> tag and ignore the others, if there are any.

You can indicate the currency to get room prices in using the *RequestedCurrency* attribute in the root node. The dates the customer wants to stay should be indicated using the <StayDateRange> as in the previous message (see above). You can decide to send the End date of the stay in the End attribute, or the number of nights in a *Duration* attribute (using the format in the example above).

You must identify what property you want information for by inserting a <HotelRef> tag and its *HotelCode* attribute. The Hotel Code must be our internal code you were provided in the previous step, in the *HotelCode* attribute of the <BasicPropertyInfo> tag.

Since the standard OTA_HotelAvailRS message cannot accommodate exhaustive property information, we will return only the details of the rooms available; you should provide more Property information using the information cached on your system (file cache or database), or by making a OTA_HotelDescriptiveInfo request to request information about the property. We strongly encourage you to save Property Info on your local server instead of requesting it each time you need it, since this info does not change very often. Remember that some establishments have reservation restrictions which can be obtained with the <Policies> tags of the OTA_HotelDescriptiveInfo, and that these policies are enforced after the room selection level, when the system assigns a specific set of rooms.

So it would be useful to display the restrictions before the customer selects the rooms.

11.4.4. Room selection in a property: the Select OTA_HotelAvailRS

The response message will look like this:

```
<?xml version="1.0" encoding="utf-8"?>
<OTA_HotelAvailRS xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05
OTA_HotelAvailRS.xsd" TimeStamp="2007-01-15T12:29:13-05:00"
Version="1.006" PrimaryLangID="en" EchoToken="1" Target="Test">
<Success/>
<RoomStays>
<RoomStay>
<RoomTypes>
<RoomType RoomTypeCode="2_13" NumberOfUnits="4">
<RoomDescription Language="en">
<Text Language="en">4 beds mixed with WC</Text>
</RoomDescription>
<Occupancy MaxOccupancy="4" MinOccupancy="4"/>
</RoomType>
</RoomTypes>
<RoomRates>
<RoomRate>
<Rates>
<Rate RateTimeUnit="Day">
<Base AmountBeforeTax="24.51" CurrencyCode="USD"/>
<RateDescription>
<Text>Shared Rooms - Beds available (prices are per bed)</Text>
</RateDescription>
</Rate>
</Rates>
</RoomRate>
</RoomRates>
<TimeSpan Start="2007-02-01" End="2007-02-02"/>
<BasicPropertyInfo HotelCode="637" HotelName="Westend City Hostel"
HotelCityCode="65"/>
</RoomStay>
<RoomStay>
<RoomTypes>
<RoomType RoomTypeCode="2_29" NumberOfUnits="8">
<RoomDescription Language="en">
<Text Language="en">8 beds mixed with WC</Text>
</RoomDescription>
<Occupancy MaxOccupancy="8" MinOccupancy="8"/>
</RoomType>
</RoomTypes>
<RoomRates>
<RoomRate>
<Rates>
<Rate RateTimeUnit="Day">
<Base AmountBeforeTax="22.31" CurrencyCode="USD"/>
<RateDescription>
<Text>Shared Rooms - Beds available (prices are per bed)</Text>
</RateDescription>
</Rate>
</Rates>
</RoomRate>
```

```
</RoomRates>
<BasicPropertyInfo HotelCode="637" HotelName="Westend City Hostel"
HotelCityCode="65"/>
</RoomStay>
<RoomStay>
<RoomTypes>
<RoomType RoomTypeCode="2_31" NumberOfUnits="4">
<RoomDescription Language="en">
<Occupancy MaxOccupancy="12" MinOccupancy="12"/>
<Text Language="en">12 beds mixed with WC</Text>
</RoomDescription>
</RoomType>
</RoomTypes>
<RoomRates>
<RoomRate>
<Rates>
<Rate RateTimeUnit="Day">
<Base AmountBeforeTax="20.89" CurrencyCode="USD"/>
<RateDescription>
<Text>Shared Rooms - Beds available (prices are per bed)</Text>
</RateDescription>
</Rate>
</Rates>
</RoomRate>
</RoomRates>
<BasicPropertyInfo HotelCode="637" HotelName="Westend City Hostel"
HotelCityCode="65"/>
</RoomStay>
</RoomStays>
</OTA_HotelAvailRS>
```

As you can see, this response returned 3 available rates. The root element is still the <RoomStays> tag, which contains a <RoomStay> tag for each of the available room types.

With the price info for each room, you will need to build some way for the customer to select the number of rooms or beds he/she wants to choose, and send the relevant data to the next step in the process.

The <RoomStay> tag contains three or four child elements: a <RoomTypes> tag, a <RoomRates> tag, a <TimeSpan> tag (**only** if the Room is not available for the whole period requested) and a <BasicPropertyInfo> tag.

11.4.4.1. The <RoomTypes> tag

The <RoomTypes> tag contains one <RoomType> child tag. The <RoomType> tag contains two attributes:

Attribute	***Value***
RoomTypeCode	This attribute indicates the room type used in our system. It is always a code that looks like this: number_number. The first number can be either a 1 (indicating Private Room) or a 2 (indicating Shared Room). The second number is the actual room type, which is our internal numeric code. Examples: 2_31, 1_2 For the room type codes, see the Appendix.
NumberOfUnits	The number of units available for the period

Attribute	*Value*
	requested. In the case of a shared room it is the number of beds available, in the case of private rooms it is the number of rooms available.

The <RoomType> tag contains a <RoomDescription> child tag, which in turn contains a <Text> child tag which contains the textual description of the room type, in the requested language.

The <RoomType> tag also contains an <Occupancy> child tag, which contains the maximum and minimum number of beds in the room, respectively in the *MaxOccupancy* and the *MinOccupancy* attribute. Currently these always have the same value, but we reserve the right to change this to reflect variable room size (extra beds, etc.). Please, for the moment, always use *MinOccupancy* if you need to display the number of beds to a user.

11.4.4.2. The <RoomRates> tag

In our system, a room will always have one rate only. Whether the rate is per room or per bed depends on the type of room (see above). Prices will be per bed/per room and per night.

Each <RoomRate> tag contains a <Rates> tag, which contains a <Rate> tag: the <Rate> tag has a *RateTimeUnit* attribute, which will always have value "Day", indicating that the rate refers to the cost of one day (in fact, one night) of stay.

The <Rate> tag contains a <Base> tag, which contains the minimum price of the room/bed. The tag has two attributes:

Attribute	*Value*
AmountBeforeTax	Contains the minimum price (per bed/room, per night) of this room type
CurrencyCode	The three-letter currency code for the currency used in the price (which will be that of the request)

It must be noted that the price indicated in the *AmountBeforeTax* attribute is the minimum price for the period. This is because most establishments charge more for weekend days, and might also (this is especially valid for hostels) have different rooms which fall in the same room type, but have different prices because one room is slightly better than the other. The actual room, complete with day-by-day prices, will be assigned in the next step, when our system verifies the request and assigns all rooms requested if they are still available.

So you must make the user aware that this are minimum prices, and that they are either per room/night or per bed/night. The <RateDescription> tag will hold a <Text> tag with the price description in the request language.

11.4.4.3. The <TimeSpan> tag

The <TimeSpan> tag will only appear in rooms that are not available for the whole period. Our system will output the *Start* and *End* attributes indicating the period in which this room is actually available. The Start/End logic of the period is always that the *Start* attribute contains the arrival date and the *End* attribute contains the departure date.

In the example above, only the first room has a <TimeSpan> tag, which looks like this:

```
<TimeSpan Start="2007-02-01" End="2007-02-02"/>
```

This shows that the first of the three rooms has availability only for the first day of the request range.

11.4.4.4. The <BasicPropertyInfo> tag

This tag will contain the *HotelCode* attribute (with the establishment's numeric code), the *HotelName* attribute and the *HotelCityCode* attribute (containing our internal city code).

```
<BasicPropertyInfo HotelCode="637" HotelName="Westend City Hostel"
HotelCityCode="65"/>
```

11.4.4.5. How to proceed sending the customer to your personalized booking interface on our server

If you want to "exit" at this point, and let the customer proceed with the payment using your personalized booking interface on our server, you must direct the customer to the following page (please note the HTTPS protocol):

https://www.hostelspoint.com/res/*your_site_code_here*/step03_book.php

Where *your_site_code_here* stands for your site code (not your affiliate code).

If you want to direct users to our public Hostelsclub site, you must direct the customer to the following URL:

https://www.hostelsclub.com/step03_book.php?aff_ID=*your_affiliate_id_here*

Where *your_affiliate_id_here* stands for your affiliate code (not your site id). Please note that the variable name is aff_ID, with capital ID.

You also need to **pass the following variables in GET**:

Variable	**Value**
str	The propery ID, which you can find in the *HotelCode* attribute of the <BasicPropertyInfo> tag.
lang	The language for the interface. It must be our two-letter language code – see the language list in the Appendix
giorno	The day of the month, in numbers. Examples: 1, 17, 23. You can also send 01 instead of 1.
mese	The number of the month. Examples: 1, 2, 12. You can also send 01 instead of 1.
anno	The year in 4-figure format (ex. 2007)
notti	The number of nights (ex. 2)
currency	The customer chosen currency – you must use our internal currency ID. See the appendix below for currency codes and IDs

A sample query string will look like this:

str=7153&lang=de&giorno=17&mese=1&anno=2007&notti=3&currency=1

You will also need to **send the following variables in POST**:

- For each room types selected by the user, a variable called like the RoomCode with a value equal to the quantity (of rooms or beds, depending on the room type) chosen by the user. For example, if for the establishment results above the customer wants to book 4 beds in the 12 beds room, you need to send a variable called 2_31 and assign it a value of 4.
- A variable called *formtype* with a value of "standard". This is an internal requirement of our system.

11.4.5. Property room assignment and payment: the Book OTA_HotelAvailRQ

To get the rooms/beds final prices, you will need to send a third OTA_HotelAvailRQ, which will look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelAvailRQ xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05
OTA_HotelAvailRQ.xsd" Version="1.006" RequestedCurrency="USD"
PrimaryLangID="en">
<POS>
<Source>
<RequestorID ID="1" MessagePassword="pwd" />
</Source>
</POS>
<AvailRequestSegments>
<AvailRequestSegment>
<HotelSearchCriteria>
<Criterion>
<StayDateRange Start="2007-02-01" End="2007-02-04" />
<HotelRef HotelCode="637"/>
<RoomStayCandidates>
      <RoomStayCandidate  RoomTypeCode="2_29"  Quantity="4"  />
</RoomStayCandidates>
</Criterion>
</HotelSearchCriteria>
</AvailRequestSegment>
</AvailRequestSegments>
</OTA_HotelAvailRQ>
```

The only difference between the previous request and this one is the fact that it containes a <RoomStayCandidates> node, which contains a series of <RoomStayCandidate> children, one for each room type.

The <RoomStayCandidate> has the following attributes:

Attribute	***Value***
RoomTypeCode	The room type code provided in the previous response
Quantity	The quantity of rooms/beds (depending whether it is a private or shared rooms) the customer wishes to book

11.4.6. Property room assignment and payment: the Book OTA_HotelAvailRS

The response message will look like this:

```
<?xml version="1.0" encoding="utf-8"?>
<OTA_HotelAvailRS xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05
OTA_HotelAvailRS.xsd" TimeStamp="2007-01-16T12:26:26-05:00"
Version="1.006" PrimaryLangID="en" EchoToken="" Target="Test">
<Success/>
<RoomStays>
<RoomTypes>
<RoomType RoomTypeCode="2_29" NumberOfUnits="4">
<RoomDescription Language="en">8 bed/s   mixed  shared with WC
</RoomDescription>
</RoomType>
</RoomTypes>
<RoomRates>
<RoomRate>
<Rate>
<Total RoomTypeCode="2_29" AmountBeforeTax="268.55"
CurrencyCode="USD"/>
</Rate>
</RoomRate>
</RoomRates>
<TimeSpan Start="2007-02-01" Duration="P3N" End="2007-02-04"/>
<DepositPayments>
<GuaranteePayment GuaranteeType="PrePay" NameInd="1"
PaymentCode="5" NonRefundableIndicator="1">
<AmountPercent FeesInclusive="true" Amount="22.76"
CurrencyCode="EUR"/>
<Description Language="en">
<Text Formatted="true" Language="en">Deposit - 10% of total price:
20.76 EUR&lt;br /&gt; (Equivalent to:  26.78 USD) &lt;br
/&gt;Booking fee: 2 EUR &lt;br /&gt;You pay now: 22.76
EUR&lt;br /&gt; (Equivalent to: 29.36 USD)&lt;br /&gt;</Text>
</Description>
</GuaranteePayment>
</DepositPayments>
<CancelPenalties>
<CancelPenalty>
<Deadline OffsetDropTime="BeforeArrival" OffsetTimeUnit="Day"
OffsetUnitMultiplier="2"/>
<AmountPercent NmbrOfNights="1"/>
</CancelPenalty>
</CancelPenalties>
<Total AmountBeforeTax="270.34" CurrencyCode="USD"/>
<BasicPropertyInfo HotelCode="637" HotelName="Westend City Hostel"
HotelCityCode="65"/>
<TPA_Extensions>
<HotelPayment AmountBeforeTax="186.84" CurrencyCode="EUR"
HotelCode="637" />
</TPA_Extensions>
</RoomStays>
<TPA_Extensions>
<ConfirmData sharedtotal="3" privatetotal="0" currency="2"
totale="207.6" p_dal="2007-02-01" p_al="2007-02-03" p_notti="3"
tot_notti="3" id_str="637" chiave="2871895-7GQp8cj40s"
```

```
sh_0="2_2007-02-01_2309_4" sh_1="2_2007-02-02_2309_4"
sh_2="2_2007-02-03_2309_4"/>
<CustomerTable>&lt;table  class=&quot;hc-hostelbook&quot;&gt;
&lt;tr&gt;&lt;td colspan=&quot;4&quot; class=&quot;hc-
nopad&quot;&gt;&lt;p class=&quot;hc-title&quot;&gt;You are booking
in: Westend City Hostel,  Vienna, Austria&lt;/p&gt;
&lt;p class=&quot;hc-arrivaldate2&quot;&gt;arrival date Feb 1, 07
&amp;nbsp;&amp;nbsp;number of nights &amp;gt;
3&lt;/p&gt;&lt;/td&gt;&lt;/tr&gt;
&lt;tr&gt;&lt;td class=&quot;hc-
greybg&quot;&gt;Date:&lt;/td&gt;&lt;td class=&quot;hc-
greybg&quot;&gt;Room type:&lt;/td&gt;&lt;td class=&quot;hc-
greybg&quot;&gt;Beds:&lt;/td&gt;&lt;td class=&quot;hc-
greybg&quot;&gt;Price:&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td
rowspan=&quot;1&quot; class=&quot;hc-lightgrey&quot;&gt;Feb 1,
07&lt;br/&gt;[total people:4]&lt;/td&gt;&lt;td class=&quot;hc-
lightgrey&quot;&gt;8 bed/s   mixed  shared with WC
&lt;/td&gt;&lt;td class=&quot;hc-
bedsbook&quot;&gt;4&lt;/td&gt;&lt;td class=&quot;hc-
pricegrey&quot;&gt;89.25 USD&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td
rowspan=&quot;1&quot; class=&quot;hc-lightgrey&quot;&gt;Feb 2,
07&lt;br/&gt;[total people:4]&lt;/td&gt;&lt;td class=&quot;hc-
lightgrey&quot;&gt;8 bed/s   mixed  shared with WC
&lt;/td&gt;&lt;td class=&quot;hc-
bedsbook&quot;&gt;4&lt;/td&gt;&lt;td class=&quot;hc-
pricegrey&quot;&gt;89.25 USD&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td
rowspan=&quot;1&quot; class=&quot;hc-lightgrey&quot;&gt;Feb 3,
07&lt;br/&gt;[total people:4]&lt;/td&gt;&lt;td class=&quot;hc-
lightgrey&quot;&gt;8 bed/s   mixed  shared with WC
&lt;/td&gt;&lt;td class=&quot;hc-
bedsbook&quot;&gt;4&lt;/td&gt;&lt;td class=&quot;hc-
pricegrey&quot;&gt;89.25 USD&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td
colspan=&quot;3&quot;  class=&quot;hc-booktd&quot;&gt;Total
price:&lt;/td&gt;&lt;td class=&quot;hc-price&quot;&gt;267.76
USD&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td colspan=&quot;3&quot;
class=&quot;hc-booktd&quot; align=&quot;right&quot; &gt;Equivalent
to:&lt;/td&gt;&lt;td class=&quot;hc-price&quot;&gt;207.60
EUR&lt;/td&gt;&lt;/tr&gt;
&lt;tr&gt;
&lt;td colspan=&quot;4&quot;&gt;&amp;nbsp;&lt;/td&gt;&lt;/tr&gt;


&lt;tr&gt;&lt;td colspan=&quot;4&quot; class=&quot;hc-
nopad&quot;&gt;&lt;p class=&quot;hc-floatinggrey&quot;&gt;DEPOSIT
TO PAY NOW&lt;/p&gt;
&lt;p class=&quot;hc-pad4&quot;&gt;The amount below is what you
pay now to confirm your
reservation.&lt;/p&gt;&lt;/td&gt;&lt;/tr&gt;
&lt;tr&gt;&lt;td colspan=&quot;3&quot; class=&quot;hc-
greybg&quot;&gt;Description:&lt;/td&gt;&lt;td class=&quot;hc-
greybg&quot;&gt;Price:&lt;/td&gt;&lt;/tr&gt;
&lt;tr&gt;&lt;td colspan=&quot;3&quot;  class=&quot;hc-
lightgrey&quot;&gt;Deposit - 10% of total price:&lt;/td&gt;&lt;td
class=&quot;hc-pricegrey&quot;&gt;26.78
USD&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td colspan=&quot;3&quot;
class=&quot;hc-lightgrey&quot;&gt;Equivalent to:&lt;/td&gt;&lt;td
class=&quot;hc-pricegrey&quot;&gt; 20.76
EUR&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td colspan=&quot;3&quot;
class=&quot;hc-lightgrey&quot;&gt;Booking fee:&lt;/td&gt;&lt;td
```

```
class=&quot;hc-pricegrey&quot;&gt;2.00 EUR&lt;/td&gt;&lt;/tr&gt;
&lt;tr&gt;&lt;td colspan=&quot;3&quot;  class=&quot;hc-
booktd&quot; &gt;You pay now:&lt;/td&gt;&lt;td class=&quot;hc-
price&quot;&gt;22.76 EUR&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td
colspan=&quot;3&quot;&gt;
Equivalent to:&lt;/td&gt;
&lt;td class=&quot;hc-price&quot;&gt;29.36
USD&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td
colspan=&quot;4&quot;&gt;You will pay the difference of  186.84
Euro (240.99 USD)  when you'll
arrive&lt;/td&gt;&lt;/tr&gt;&lt;/table&gt;</CustomerTable>
<CustomerDataJSON>{"2012-08-02":{"stringDate":"Aug 2, 12","rooms":
[{"type":"2 bed\/s private with
WC","guests":"2","price":"57.58","price_local":"43.30"},{"type":"3
bed\/s mixed dorm shared
WC","guests":"1","price":"15.82","price_local":"11.90"}]},"2012-
08-03":{"stringDate":"Aug 3, 12","rooms":[{"type":"2 bed\/s
private with
WC","guests":"2","price":"57.58","price_local":"43.30"},{"type":"3
bed\/s mixed dorm shared
WC","guests":"1","price":"15.82","price_local":"11.90"}]},"2012-
08-04":{"stringDate":"Aug 4, 12","rooms":[{"type":"2 bed\/s
private with
WC","guests":"2","price":"57.58","price_local":"43.30"},{"type":"3
bed\/s mixed dorm shared
WC","guests":"1","price":"15.82","price_local":"11.90"}]}}</Custom
erDataJSON>
</TPA_Extensions>
</OTA_HotelAvailRS>
```

11.4.6.1. The main response element: the <RoomStays> node

This element contains the following tags:

11.4.6.1.1. The <RoomTypes> tag:

```
<RoomTypes>
<RoomType RoomTypeCode="2_29" NumberOfUnits="4">
<RoomDescription Language="en">8 bed/s   mixed  shared with WC
</RoomDescription>
</RoomType>
</RoomTypes>
```

This node contains one <RoomType> tag for each room type in the request (obviously, if there is availability for all rooms requested). The <RoomType> tag has the *RoomTypeCode* attribute set to the room code, and the *NumberOfUnits* attribute set to the quantity requested. The <RoomType> tag contains a <RoomDescription> tag, with a textual description of the room in the language of the request.

11.4.6.1.2. The <RoomRates> tag:

```
<RoomRates>
<RoomRate>
<Rate>
<Total RoomTypeCode="2_29" AmountBeforeTax="207.6"
CurrencyCode="USD"/>
</Rate>
</RoomRate>
```

```
</RoomRates>
```

The </RoomRates> tag contains a <RoomRate> tag for each room type in the request, which contains a <Rate> tag containing a <Total> tag, which has 3 attributes:

Attribute	***Value***
RoomTypeCode	The code for the room type
AmountBeforeTax	The price for the total number of rooms/beds of this type, for the whole stay, not including service fees
CurrencyCode	The currency code in three-letter ISO code

If you want to display a room breakdown with prices to the customer, you can use our pre-prepared HTML data (see below), otherwise you can pair each Rate with the Room Type in the previous set of tags, and display a total price per room type for the whole period.

11.4.6.1.3. The <TimeSpan> tag:

```
<TimeSpan Start="2007-02-01" Duration="P3N" End="2007-02-04"/>
```

This tag contains the start, end and duration of the period requested.

11.4.6.1.4. The <DepositPayments> tag:

```
<DepositPayments>
<GuaranteePayment GuaranteeType="PrePay" NameInd="1"
PaymentCode="5" NonRefundableIndicator="1">
<AmountPercent FeesInclusive="true" Amount="22.76"
CurrencyCode="EUR"/>
<Description Language="en">
<Text Formatted="true" Language="en">Deposit - 10% of total price:
20.76 EUR&lt;br /&gt; (Equivalent to:  26.78 USD) &lt;br
/&gt;Booking fee: 2 EUR &lt;br /&gt;You pay now: 22.76
EUR&lt;br /&gt; (Equivalent to: 29.36 USD)&lt;br /&gt;</Text>
</Description>
</GuaranteePayment>
</DepositPayments>
```

This tag describes, both in structured OTA-compatible data, as well as in plain text, the deposit and booking fee required to confirm the payment.

It is advisable to output the content of the <Description> <Text> tag to the customer: it contains the details in the request language, in entity-escaped XHTML.

Since the payment must be made in EUR, this field also contains the equivalent value in the request currency, which cannot be properly described using standard OTA code.

11.4.6.1.5. The <CancelPenalties> tag:

```
<CancelPenalties>
<CancelPenalty>
<Deadline OffsetDropTime="BeforeArrival" OffsetTimeUnit="Day"
OffsetUnitMultiplier="2"/>
<AmountPercent NmbrOfNights="1"/>
</CancelPenalty>
</CancelPenalties>
```

This is the OTA-compatible description of our standard cancellation policy: the reservation must be cancelled two days in advance (until midnight, server time, of the day before the day before the reservation starts), otherwise the establishment will be allowed to charge one room/night on the credit card provided for the reservation.

11.4.6.1.6. The <Total> tag:

```
<Total AmountBeforeTax="270.34" CurrencyCode="USD"/>
```

This tags contains the TOTAL cost of the reservation (total rooms costs + booking fee), in the requested currency.

11.4.6.1.7. The <BasicPropertyInfo> tag:

```
<BasicPropertyInfo HotelCode="637" HotelName="Westend City Hostel"
HotelCityCode="65"/>
```

This tag contains the usual basic property information.

11.4.6.1.8. The first <TPA_Extensions> tag:

```
<TPA_Extensions>
<HotelPayment AmountBeforeTax="186.84" CurrencyCode="EUR"
HotelCode="637" />
</TPA_Extensions>
```

This tag is contained in the <RoomStays> tag, and contains the amount that must be paid at the property upon arrival. Since currently our system collects just a 10% deposit (plus booking fees), the remaining amount must be paid to the property. The property has a working currency, and this amount is in this currency (which can be different from the request currency and the Euro used for the deposit), so that it correspond exactly to what the property expects to receive.

11.4.6.2. The final <TPA_Extensions> node

After the main response node, we output another <TPA_Extensions> tag, which contains two important pieces of information: the <ConfirmData> tag and the <CustomerTable> tag.

11.4.6.2.1. The <ConfirmData> tag

This tag contains data which must be passed on **as it is** to the next message, together with customer data (name, credit card, etc.) to finalize the reservation. It contains a summary of the reservation data and the individual room codes for the actual rooms assigned by the system. It also contains a unique key (the *chiave* attribute). You will see its importance in the next step.

```
<ConfirmData sharedtotal="3" privatetotal="0" currency="2"
totale="207.6" p_dal="2007-02-01" p_al="2007-02-03" p_notti="3"
tot_notti="3" id_str="637" chiave="2871895-7GQp8cj40s"
sh_0="2_2007-02-01_2309_4" sh_1="2_2007-02-02_2309_4"
sh_2="2_2007-02-03_2309_4"/>
```

11.4.6.2.2. The <CustomerTable> tag

This has been included for your convenience. It is an XHTML <table> (entity-escaped), containing all the data about the reservation (cancellation policy excluded), conveniently preformatted and in the request language. You can output this to the end customer to show a summary of room costs, payment data, deposit, totals, etc. It will contain exactly the same info that is shown on our site at the same step. This tag will contain a day-by-day breakdown of rooms, including each room's separate price and the total beds booked per day. It also includes a conversion of the data in the <HotelPayment> tag in the customer's preferred currency. The table contains css class attributes so you can apply styles to it.

11.4.6.2.3. The <CustomerDataJSON> tag

This has been included for your convenience as well. It contains the room breakdown, day by day. It is the same information as in the CustomerTable above, in JSON Object format (we've chosen JSON for this data for performance reasons). Every day in the reservation period has an entry in the object, with the date in db-friendly format (YYYY-MM-DD) as the property name, and an object with the following values:

- *stringDate* contains the date formatted in the request language
- *rooms* contains an array of rooms objects, each with a *type* (room description in request language), number of *guests* booked in that room type, *price* in the currency of the request and *price_local* which is the price in the property's own currency.

11.5. The OTA_HotelResRQ/RS message

This is the message which confirms the reservation. You need to send the <ConfirmData> from the previous call, as well as information about the customer and his/her credit card data. You should preferably collect this data on an https-protected page. It is best to use https as early as possible (on the page that collects the credit card data) to maximize the customers' perception of security.

This message requires an https connection.

You will need to collect the following information from the customer:

Data	***Format***	***Comment***
Email	Standard e-mail format	A correctly-formatted email address is required for sending the confirmation to the customer
First Name	Text	
Last Name	Text	
Nationality	ISO country code	You will need to collect the nationality of the user, and send an ISO country code to our system.
Gender	Either "Male" or "Female"	Whether the customer is male or female. If not present, it will default to "Male" without issuing warnings.
Estimated arrival time	Hours and minutes: examples are 09:00, 23:00. We will only consider the hour value, not the minutes (for example, 09:30 will be converted to 09:00)	The customer's estimated arrival time, which must be between 9AM and 24 (midnight).
Telephone number	A text value (can include symbols like + beside numbers)	This data is **optional.**
Comment	Text – maximum length is 255 characters.	**Optional**. A comment/request the customer would like to send to the property together with their reservation. The customer should be advised that the comment should be in a language the

Data	*Format*	*Comment*
		property could reasonably understand (for example, no messages in Chinese to Italian properties unless they specify that they speak Chinese in the description...)
Credit Card number	Credit card number: all-figures (no spaces in between numbers)	
Credit Card Expiry date	The credit card expiry date (month and year in the mmyy format). Must be 4 numbers.	Examples are: 0108 (Jan 2008) 1209 (Dec 2009)
Card Type	A two-letter code for the card type. See Appendix for accepted credit card codes.	We've added some codes of our own to those provided by OTA Standards
Series Code (Card Security Code)	The card's security code. See http://en.wikipedia.org/wiki/Card_Security_Code for details.	

You are required to set up some means of pre-checking of the completeness and correctness of the data on your side (Client-side (Javascript), as well as Server-side). This will reduce useless calls to our servers. We will obviously check the data we receive, but we require you to do it as well.

All this information needs to be passed through the OTA_HotelResRQ message.

11.5.1. The OTA_HotelResRQ message

A sample message will look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelResRQ xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05
OTA_HotelResRQ.xsd" Version="1.003" EchoToken="sasaf2"
Target="Production" PrimaryLangID="en">
<POS>
<Source>
<RequestorID ID="1" MessagePassword="pwd" />
</Source>
</POS>
<HotelReservations>
<HotelReservation>
<RoomStays>
<RoomStay>
<Guarantee>
<GuaranteesAccepted>
<GuaranteeAccepted>
<PaymentCard CardCode="VI" CardNumber="***************"
ExpireDate="0208" SeriesCode="***">
<CardHolderName>SURNAME NAME</CardHolderName>
</PaymentCard>
</GuaranteeAccepted>
</GuaranteesAccepted>
</Guarantee>
```

```
</RoomStay>
</RoomStays>
<ResGuests>
<ResGuest ArrivalTime="10:00">
<Profiles>
<ProfileInfo>
<Profile>
<Customer Gender="Male">
<PersonName>
<GivenName>NAME</GivenName>
<Surname>SURNAME</Surname>
</PersonName>
<Telephone PhoneNumber="+1111111111"/>
<Email>email@example.org</Email>
<CitizenCountryName  Code="US" />
</Customer>
</Profile>
</ProfileInfo>
</Profiles>
<Comments>
<Text>COMMENT COMMENT COMMENT COMMENT COMMENT </Text>
</Comments>
</ResGuest>
</ResGuests>
<TPA_Extensions>
<ConfirmData sharedtotal="0" privatetotal="1" currency="2"
totale="1" p_dal="2009-01-17" p_al="2009-01-17" p_notti="1"
tot_notti="1" id_str="2530" chiave="2878309-zDMbF0tvg4"
pr_0="1_2009-01-17_33762"/>
</TPA_Extensions>
</HotelReservation>
</HotelReservations>
</OTA_HotelResRQ>
```

As with previous messages, although the OTA standards allow for the sending of multiple reservation requests in the same message, we will parse and process the first one only, ignoring other requests.

It is quite evident what piece of data goes where in the XML of the message. The <ConfirmData> tag obtained from the previous call will have to be extracted and inserted verbatim as a child node of the <TPA_ Extensions> tag. There is no need to insert data about the property or the rooms requested in any other tags. This tag's attributes summarizes all the room data we require to proceed with reservation.

Important: the attribute “SeriesCode” in the <PaymentCard> tag must contain the CVV2/CVC2/4DBC verification code which is found on Visa and Mastercards on the back of the card, and on the front for American Express Cards (see http://en.wikipedia.org/wiki/Card_Security_Code) for more info. It is compulsory for all cards **EXCEPT JCB cards** which do not have that code.

11.5.2. About e-mails to the customer and to the affiliate

Our system by default will send a confirmation e-mail to the customer, including cancellation information and an URL for cancelling the reservation. The URL will be that of your personalized booking interface's customer area. We will also provide the customer with means to login in the customer area and cancel the reservation, if they wish to do so, or retrieve reservation data and print an on-line receipt (although the printed e-mail itself is considered a sufficient proof of reservation).

If you wish to send your own confirmation e-mail, you must make a request to our tech staff. We will disable the sending of the e-mail to the customer, but in that case you **must** provide the customer with a way of cancelling the reservation through your site, using the OTA_ReadRQ and the OTA_CancelRQ messages described below. The e-mail, when used as a proof of receipt, must contain a reference to your site's name **as inserted in your affiliate back office**, as that will be the name seen by the properties in their back office. You must also mention "Hostelsclub" as the booking system used, to help properties identify the source of the reservation.

You can also ask us to send you an e-mail with each reservation that takes place using your affiliate account. Although this will probably not be necessary if you have implemented the full XML procedure, you can request it. You will receive a copy of the *establishment e-mail*, that is a copy of the email sent to the establishment, not the one sent to the customer.

11.5.3. About Test and Production usage of this message

If you send the *Target* attribute in the root element and assign it a value of "Test", or if your account has not been enabled to send Production calls, the system will just simulate the reservation, without processing the Credit Card. If the reservation is successful, you will receive a copy of the e-mail that would have been sent to the establishment. The customer e-mail you sent in your message will also receive a sample e-mail confirmation message, marked "- TESTING RESERVATION" in the mail subject. The system will return a fake reservation ID with a value of "9999999999".

The response message will contain the same checks as if a standard reservation was taking place, so that if you receive a positive response you can be sure that your call was structurally correct and contained all the data required for a Production reservation. The Credit Card number will not be checked for credit, and it will not be saved anywhere. You can use fake numbers if you prefer.

11.5.4. About the unique key (chiave) and multiple requests for the same room/s

If our system processes a reservation request, the message is correctly formatted and all information required for a reservation was available, it will try to check if all availability is correct and then it will try to charge the credit card provided and, if successful, it will issue a response containing the details of the processed reservation.

However, if the reservation fails at this step, it can be for a variety of reasons. Some are:

- The credit card is not valid
- The credit card has no credit left
- The rooms aren't available any more (they have been booked by someone else or availability has been removed by the property)
- Prices have changed – if the property has increased room prices exactly in the interval between when the customer is inserting his/her data in the web form and when the customer actually sends the data, the reservation process is blocked, since the customer cannot be charged more or less than what he/she was told (at the previous step) was the price to be paid.

In all the above cases, we will require you to **make another request for room availability** (step *Book* above, using the HotelAvailRQ) and obtain a new set of <ConfirmData> values.

This will try and find other rooms of the same type if the specific rooms are not available any more, or maybe reassign exactly the same rooms if there was a credit card error.

This is because we issue a unique key for each booking request, and this key "expires" once the customer has tried to use it for a reservation. This is for both security and general performance reasons. If the customer tries to resend the same request using the same set of data (for ex., by using the "back" button of the browser, inserting data in the same form preserved in their browser history), our system will issue an error.

11.5.5. The OTA_HotelResRSmessage

This message contains the details of the reservation, if the reservation was successful, or the errors/warnings that indicate the reasons why it has not succeeded. If the reservation has been successful, the message will contain the main response element, the <HotelReservations> tag, otherwise it will only contain <Errors> or <Warning> tags (see chapter 9.2 above).

A sample message for a successful reservation will look like this:

```
<?xml version="1.0" encoding="utf-8"?>
<OTA_HotelResRS xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05
OTA_HotelResRS.xsd" TimeStamp="2007-01-17T10:54:46-05:00"
Version="1.003" PrimaryLangID="en" EchoToken="sasaf2"
Target="Production">
<Success/>
<HotelReservations>
<HotelReservation ResStatus="Reserved">
<RoomStays>
<RoomStay>
<RoomTypes>
<RoomType RoomTypeCode="1_66" NumberOfUnits="1">
<RoomDescription Language="en">
<Text Language="en">Double bed private with WC</Text>
</RoomDescription>
</RoomType>
</RoomTypes>
<BasicPropertyInfo HotelCode="2530" HotelName="Test establishment"
HotelCityCode="1193">
<Address>
<AddressLine>Test address</AddressLine>
<CityName>Zafferana Etnea</CityName>
<CountryName Code="IT">Italy</CountryName>
</Address>
<ContactNumbers>
<ContactNumber PhoneNumber="24422121421"/>
</ContactNumbers>
</BasicPropertyInfo>
<TPA_Extensions>
<HotelEmail>hotelemail@example.org</HotelEmail>
<Directions Language="en">
<Text Language="en">Test directions Test directions Test
directions Test directions Test directions Test directions Test
directions Test directions Test directions Test directions Test
directions Test directions </Text>
</Directions>
```

```
</TPA_Extensions>
</RoomStay>
</RoomStays>
<ResGuests>
<ResGuest>
<Profiles>
<ProfileInfo>
<Profile>
<Customer Gender="Male">
<PersonName>
<GivenName>Name</GivenName>
<Surname>Surname</Surname>
</PersonName>
<Telephone PhoneNumber=" 111111111111"/>
<Email>email@example.org</Email>
<CitizenCountryName Code="US"/>
</Customer>
</Profile>
</ProfileInfo>
</Profiles>
</ResGuest>
</ResGuests>
<ResGlobalInfo>
<TimeSpan Start="2009-01-17" End="2009-01-18"/>
<SpecialRequests>
<SpecialRequest>
<Text>We are coming with our colleagues, so it would be nice that
we are sleeping all together.</Text>
</SpecialRequest>
</SpecialRequests>
<DepositPayments>
<GuaranteePayment GuaranteeType="PrePay" NameInd="1"
PaymentCode="5" NonRefundableIndicator="true">
<AmountPercent FeesInclusive="true" Amount="12.00"
CurrencyCode="EUR"/>
<Description Language="en">
<Text Language="en">Deposit - 10% of total price</Text>
</Description>
</GuaranteePayment>
</DepositPayments>
<CancelPenalties>
<CancelPenalty>
<Deadline OffsetDropTime="BeforeArrival" OffsetTimeUnit="Day"
OffsetUnitMultiplier="2"/><
AmountPercent NmbrOfNights="1"/>
</CancelPenalty>
</CancelPenalties>
<Fees>
<Fee ChargeFrequency="26" Amount="2" Code="14" CurrencyCode="EUR"
MandatoryIndicator="true" TaxInclusive="true" Type="Inclusive">
<Description Language="en">
<Text Language="en">Booking fee</Text>
</Description>
</Fee>
</Fees>
<Total AmountBeforeTax="102.00" CurrencyCode="EUR"/>
<HotelReservationIDs>
<HotelReservationID ResID_Value="249358" ResID_Type="14"
ForGuest="true"/>
```

```
</HotelReservationIDs>
</ResGlobalInfo>
<TPA_Extensions>
<HotelPayment AmountBeforeTax="90.00" CurrencyCode="EUR"/>
<CancelCode Code="20fKVDBaa14hf"/>
</TPA_Extensions>
</HotelReservation>
</HotelReservations>
</OTA_HotelResRS>
```

This message's main response node is the <HotelReservations> tag, which in our case contains just one <HotelReservation> tag, for the current reservation.

11.5.5.1. The <HotelReservation> tag

The <HotelReservation> tag contains the following information:

11.5.5.1.1. The <RoomStays> tag

```
<RoomStays>
<RoomStay>
<RoomTypes>
<RoomType RoomTypeCode="1_66" NumberOfUnits="1">
<RoomDescription Language="en">
<Text Language="en">Double bed private with WC</Text>
</RoomDescription>
</RoomType>
</RoomTypes>
<BasicPropertyInfo HotelCode="2530" HotelName="Test establishment"
HotelCityCode="1193">
<Address>
<AddressLine>Test address</AddressLine>
<CityName>Zafferana Etnea</CityName>
<CountryName Code="IT">Italy</CountryName>
</Address>
<ContactNumbers>
<ContactNumber PhoneNumber="24422121421"/>
</ContactNumbers>
</BasicPropertyInfo>
<TPA_Extensions>
<HotelEmail>hotelemail@example.org</HotelEmail>
<Directions Language="en">
<Text Language="en">Test directions Test directions Test
directions Test directions Test directions Test directions Test
directions Test directions Test directions Test directions Test
directions Test directions </Text>
</Directions>
</TPA_Extensions>
</RoomStay>
</RoomStays>
```

This tag contains several subtags.

The <RoomTypes> tag contains the global room breakdown: for each room type booked, there will be a <RoomType> node, containing the room code, the number of units booked (beds or rooms, depending on room type), and the room description in the request language.

The <BasicPropertyInfo> tag:

```
<BasicPropertyInfo HotelCode="2530" HotelName="Test establishment"
HotelCityCode="1193">
<Address>
<AddressLine>Test address</AddressLine>
<CityName>Zafferana Etnea</CityName>
<CountryName Code="IT">Italy</CountryName>
</Address>
<ContactNumbers>
<ContactNumber PhoneNumber="24422121421"/>
</ContactNumbers>
</BasicPropertyInfo>
```

This tag contains all information about the property. Name, code and address, as well as the property's phone number.

It will be followed by a <TPA_Extensions> tag:

```
<TPA_Extensions>
<HotelEmail>hotelemail@example.org</HotelEmail>
<Directions Language="en">
<Text Language="en">Test directions Test directions Test
directions Test directions Test directions Test directions Test
directions Test directions Test directions Test directions Test
directions Test directions </Text>
</Directions>
</TPA_Extensions>
```

This tag contains further info about the property: the email, and the text directions on how to get to the property. The text directions will be in the request language, if available, otherwise they will be in English. The *Language* attribute of the <Directions> and <Text> tag will tell you what language the information is in.

11.5.5.1.2. The <ResGuests> tag

```
<ResGuests>
<ResGuest>
<Profiles>
<ProfileInfo>
<Profile>
<Customer Gender="Male">
<PersonName>
<GivenName>Name</GivenName>
<Surname>Surname</Surname>
</PersonName>
<Telephone PhoneNumber=" 111111111111"/>
<Email>email@example.org</Email>
<CitizenCountryName Code="US"/>
</Customer>
</Profile>
</ProfileInfo>
</Profiles>
</ResGuest>
</ResGuests>
```

This tag will contain one <ResGuest> tag, with all the information about the customer: gender, name, telephone number (if provided at the time of booking), email and nationality code. Our system will always return one <ResGuest> tag, because we always request the information for one of the customers booking the room/s, not for all of them.

11.5.5.1.3. The <ResGlobalInfo> tag

```
<ResGlobalInfo>
<TimeSpan Start="2009-01-17" End="2009-01-18"/>
<SpecialRequests>
<SpecialRequest>
<Text>We are coming with our colleagues, so it would be nice that
we are sleeping all together.</Text>
</SpecialRequest>
</SpecialRequests>
<DepositPayments>
<GuaranteePayment GuaranteeType="PrePay" NameInd="1"
PaymentCode="5" NonRefundableIndicator="true">
<AmountPercent FeesInclusive="false" Amount="10.00"
CurrencyCode="EUR"/>
<Description Language="en">
<Text Language="en">Deposit - 10% of total price</Text>
</Description>
</GuaranteePayment>
</DepositPayments>
<CancelPenalties>
<CancelPenalty>
<Deadline OffsetDropTime="BeforeArrival" OffsetTimeUnit="Day"
OffsetUnitMultiplier="2"/><
AmountPercent NmbrOfNights="1"/>
</CancelPenalty>
</CancelPenalties>
<Fees>
<Fee ChargeFrequency="26" Amount="2" Code="14" CurrencyCode="EUR"
MandatoryIndicator="true" TaxInclusive="true" Type="Inclusive">
<Description Language="en">
<Text Language="en">Booking fee</Text>
</Description>
</Fee>
</Fees>
<Total AmountBeforeTax="102.00" CurrencyCode="EUR"/>
<HotelReservationIDs>
<HotelReservationID ResID_Value="249358" ResID_Type="14"
ForGuest="true" />
</HotelReservationIDs>
</ResGlobalInfo>
```

This tag contains the general information about the reservation. Start and End dates (remember that *End* is the check-out date), special messages from the user, the type of deposit and fees paid, cancel penalties, and the <Total> tag containing the total amount of the reservation, fees included (this is the total amount of all rooms plus booking fee, not the sum of the deposit plus booking fee).

There is a <HotelReservationIDs> tag containing a <HotelReservationID> tag. This has the attribute *ResID_Value* which contains the reservation number, which is the Confirmation number (as indicated by the *ResID_Type* set to 14, the OTA code for Reservation). The *ForGuest* attribute indicates that this number should be shown to the end customer. The same reservation number goes to the property, as we do not use separate reservation IDs for internal and public use.

Please note: if you're doing a test reservation, the *ResID_Value* attribute will always have a value of "9999999999", indicating that this is a test reservation and hasn't been inserted in the database.

11.5.5.1.4. The final <TPA_Extensions> tag

There is another <TPA_Extensions> tag, which is a direct child of <HotelReservation> (instead of being a child of the <RoomStay> tag like the previous one):

```
<TPA_Extensions>
<HotelPayment AmountBeforeTax="90.00" CurrencyCode="EUR"/>
<CancelCode Code="20fKVDBaa14hf"/>
</TPA_Extensions>
```

It contains two tags: one is the <HotelPayment> tag, which contains the amount to be paid to the establishment upon arrival, with the corresponding currency code (which can be different from both the request currency and Euro).

The second tag is the <CancelCode> tag: this code should be shown to the customer if you implement a cancellation system on your server, and especially if you are sending the customer the confirmation e-mail yourselves. This code must be sent to be able to cancel the reservation.

11.6. The OTA_ReadRQ/OTA_ResRetrieveRS

This message is used to retrieve data about an existing reservation. Note that the two messages (request and response) have a different base name. The request is called OTA_Read, and the response is OTA_ResRetrieve. If you implement a system that automatically recognizes the root element, it is advisable that you make pairs association request->response, instead of assuming that the response message will have a root element that is identical to the RQ message root element, with RS instead of RQ.

This message requires an https connection.

11.6.1. The OTA_ReadRQmessage

The message looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_ReadRQ xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05
OTA_ReadRQ.xsd" EchoToken="5553429" TimeStamp="2003-03-
17T09:30:47-05:00" Target="Production"
Version="1.006" SequenceNmbr="1" PrimaryLangID="it">
<POS>
<Source>
<RequestorID ID="1" MessagePassword="pwd" />
</Source>
</POS>
<ReadRequests>
<ReadRequest>
<UniqueID Type="14" ID="249695" />
</ReadRequest>
</ReadRequests>
</OTA_ReadRQ>
```

This message contains just a <ReadRequests> tag, containing a <ReadRequest> tag, which in turn contains a <UniqueID> tag, with attributes *Type* set to 14 to indicate Reservation Confirmation Number, and *ID* containing our reservation ID, which is the one you received from the confirmation message.

Our system will allow you to read all requests made using the site identified by the <POS>

node. It is up to you to use a means of identification to check that the customer is actually the one that made that reservation. If this is too much of a burden for your level of implementation, we advise to let us send the customer e-mail, so that he/she will have means of reviewing/cancelling the reservation by using our the hostelspoint.com customer area.

As usual, our system will accept one request at a time. If you want to retrieve multiple reservations, please make multiple calls.

11.6.2. The <OTA_ResRetrieveRS> message

The response data is identical to the <HotelReservation> tag returned by the OTA_HotelResRS message we've seen above, only the tag is contained in a <ReservationsList> tag. The only difference is the fact that the *ResStatus* attribute of the <HotelReservation> tag can have two values: "Reserved" and "Cancelled", if the reservation has been cancelled. If the reservation has been cancelled, the room breakdown data is not present any more. Note that the <CancelCode> tag is present here, so it can be retrieved with this message if a customer needs to cancel a reservation.

```
<?xml version="1.0" encoding="utf-8"?>
<OTA_ResRetrieveRS xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05
OTA_ResRetrieveRS.xsd" TimeStamp="2007-01-18T09:10:31-05:00"
Version="1.006" PrimaryLangID="it" EchoToken="5553429"
Target="Test">
<Success/>
<ReservationsList>
<HotelReservation ResStatus="Reserved">
<RoomStays>
<RoomStay>
<RoomTypes>
<RoomType RoomTypeCode="1_66" NumberOfUnits="1">
<RoomDescription Language="it">
<Text Language="it">letto matrimoniale  stanza privata con WC
</Text>
</RoomDescription>
</RoomType>
</RoomTypes>
<BasicPropertyInfo HotelCode="2530" HotelName="Paolo test2"
HotelCityCode="1193">
<Address>
<AddressLine>Test address</AddressLine>
<CityName>Zafferana Etnea</CityName>
<CountryName Code="IT">Italia</CountryName>
</Address>
<ContactNumbers>
<ContactNumber PhoneNumber=""/>
</ContactNumbers>
</BasicPropertyInfo>
<TPA_Extensions>
<HotelEmail>pmioni@hce.it</HotelEmail>
<Directions Language="en">
<Text Language="en">Test directions Test directions Test
directions Test directions Test directions Test directions Test
directions Test directions Test directions Test directions Test
directions Test directions </Text>
</Directions>
```

```
</TPA_Extensions>
</RoomStay>
</RoomStays>
<ResGuests>
<ResGuest>
<Profiles>
<ProfileInfo>
<Profile>
<Customer Gender="Male">
<PersonName>
<GivenName>Paolo</GivenName>
<Surname>Mioni</Surname>
</PersonName>
<Telephone PhoneNumber=" 393493821635"/>
<Email>paolo@hostelsclub.com</Email>
<CitizenCountryName Code="IT"/>
</Customer>
</Profile>
</ProfileInfo>
</Profiles>
</ResGuest>
</ResGuests>
<ResGlobalInfo>
<TimeSpan Start="2009-01-19" End="2009-01-20"/>
<SpecialRequests>
<SpecialRequest>
<Text>We are coming with our colleages,so it would be nice that we
are sleeping all together.</Text>
</SpecialRequest>
</SpecialRequests>
<DepositPayments>
<GuaranteePayment GuaranteeType="PrePay" NameInd="1"
PaymentCode="5" NonRefundableIndicator="true">
<AmountPercent FeesInclusive="false" Amount="10"
CurrencyCode="EUR"/>
<Description Language="it">
<Text Language="it">Anticipo - 10% del prezzo totale</Text>
</Description>
</GuaranteePayment>
</DepositPayments>
<CancelPenalties>
<CancelPenalty>
<Deadline OffsetDropTime="BeforeArrival" OffsetTimeUnit="Day"
OffsetUnitMultiplier="2"/>
<AmountPercent NmbrOfNights="1"/>
</CancelPenalty>
</CancelPenalties>
<Fees>
<Fee ChargeFrequency="26" Amount="2" Code="14" CurrencyCode="EUR"
MandatoryIndicator="true" TaxInclusive="true" Type="Inclusive">
<Description Language="en">
<Text Language="en">Booking fee</Text>
</Description>
</Fee>
<Total AmountBeforeTax="102" CurrencyCode="EUR"/>
<HotelReservationIDs>
<HotelReservationID ResID_Value="249695" ResID_Type="14"/>
</HotelReservationIDs>
</ResGlobalInfo>
```

```
<TPA_Extensions>
<HotelPayment AmountBeforeTax="90" CurrencyCode="EUR"/>
<CancelCode Code="y0N7p9tjqRm1"/>
</TPA_Extensions>
</HotelReservation>
</ReservationsList>
</OTA_ResRetrieveRS>
```

11.7. The <OTA_CancelRQ/RS> message

This message is used to cancel an existing reservation.

This message requires an https connection.

11.7.1. The <OTA_CancelRQ> message

This message looks like this.

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_CancelRQ xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05
OTA_CancelRQ.xsd" EchoToken="892345" TimeStamp="2003-03-
17T09:30:47-05:00" Target="Production"
Version="1.001" CancelType="Cancel">
<POS>
<Source>
<RequestorID ID="1" MessagePassword="pwd" />
</Source>
</POS>
<UniqueID Type="14" ID="1202510552" />
<Verification>
      <TPA_Extensions><CancelCode Code="rft0fpHVQJaa1"
/></TPA_Extensions>
</Verification>
</OTA_CancelRQ>
```

Please note that even if the OTA specs require the use of a *CancelType* attribute at the root element, with different values indicating the various possible options (Initiate | Ignore | Commit | Cancel), we default to "Cancel". Any <OTA_CancelRQ> message with appropriate credentials and verification data will proceed to cancel a reservation, unless you set the *Target* value to "Test". In this latter case, the system will reply as if it had cancelled the reservation, but it won't really cancel it.

This message has two main element nodes. One is the <UniqueID> tag, containing the reservation ID in the usual format.

The second is the <Verification> tag. To cancel a reservation in our system, you need to insert a <TPA_extensions> tag, which must contain a <CancelCode> tag, with a *Code* attribute containing the Cancel Code which has been provided by either the <OTA_HotelResRS> or the <OTA_ResRetrieveRS>.

11.7.2. The <OTA_CancelRS>

This message is a bit different from the others, in that its success/failure is indicated in a root element attribute, the *Status* attribute.

An unsuccessful cancellation will look like this:

```
<?xml version="1.0" encoding="utf-8"?>
<OTA_CancelRS xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05
OTA_CancelRS.xsd" TimeStamp="2007-01-18T10:15:58-05:00"
Version="1.001" PrimaryLangID="en" EchoToken="892345"
Target="Test" Status="Unsuccessful">
<Success/>
<Warnings>
<Warning Code="193" Type="11"> This reservation cannot be
cancelled</Warning>
</Warnings>
<UniqueID Type="14" ID="227430"/>
</OTA_CancelRS>
```

A successful cancellation will look like this:

```
<?xml version="1.0" encoding="utf-8"?>
<OTA_CancelRS xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05
OTA_CancelRS.xsd" TimeStamp="2007-01-18T10:38:26-05:00"
Version="1.001" PrimaryLangID="en" EchoToken="892345"
Target="Test" Status="Cancelled">
<Success/>
<UniqueID Type="14" ID="249695"/>
</OTA_CancelRS>
```

Both messages will carry the same main response element, <UniqueID> with the reservation code. The fact that the reservation has been successfully cancelled can be seen from the *Status* attribute, which is "Cancelled" if the reservation could be cancelled, and "Unsuccessful" if it could not be cancelled. The <Warnings> tags will give you the reason why the reservation could not be cancelled.

Possible reasons are:

- The CancelCode is invalid
- The Reservation could not be found: the ID is wrong, or the reservation has not been made with the calling site
- The Reservation has already been cancelled, or it is beyond cancellation deadline, or it is taking place/has already taken place.

Our system will send its own e-mail to the customers to confirm that the reservation has been cancelled.

12. The JSON-based Web Service

Currently the JSON Web Service covers just one request type, the request for availability in a city. If you have specific needs for it to manage other request types, please contact our support team explaining your requirements and we will evaluate your request.

12.1 Making calls to the service

The service can be accessed with a simple GET call to an URL. It will reply with a JSON file. JSON files are easily parsed in most programming languages, and most of all they can be forwarded to your application's Javascript frontend without need for changes, since JSON is natively understood by all Javascript clients.

12.1.1. Authentication

Authentication is based on your site's ID and IP – your site (which you can create from within your Affiliate administration) needs to be enabled for web service: contact our support team to have it enabled, indicating the IP or IP range your request will come from. All requests must come from the IP ranges that you have provided us for that site. Please note that your application will have to query our server through its own server, it cannot send direct requests to our server from the customer's browser, as they won't be authenticated.

12.1.2. Address

The web service resides at the following address:

http://www.hostelspoint.com/webservices/affiliates/json.php

12.1.3. GET parameters

You will need to send some of the following parameters in GET:

Parameters	*Possible Values*	*Compulsory*	*Notes*
IDSite	The ID of your site	Yes	Must be a valid site ID for a web service-enabled site
requestType	AvailSearchCity	Yes	
day	Day of month for search – two numbers (examples: 01, 02, 03, 10, 30)	Yes	
year_month	Year and month for search in YYYY-MM format (2010-03, 2011-11, 2012-05)	Yes	
nights	1-30	Yes	
currency	Hostelsclub Currency code (see appendix)	No, defaults to EUR	
cat	0: All property types	No, defaults to 0	Property category: only

Parameters	***Possible Values***	***Compulsory***	***Notes***
	2: Hotel * 3: Hotel ** 4: Hotel *** 8: Hotel **** 9: Hotel ***** 1: Hostel 5: Guest House 6: Bed & Breakfast 7: Camping 10: Apartment 11: All hotel types		property matching this category will be returned
guests	From 1 to 20	Yes	
city	Hostelsclub City ID	No	If not filled in, you must use the parameters below.
cityName	Name of city in same language of request	See notes	If city is not present, the system will use these textual values to make the match. ISOcountry is checked before countryName. Parameters stateCode and stateName refer to US states, like California etc.
countryName		See notes	
ISOCountry		See notes	
stateCode		See notes	
stateName		See notes	
lang	See appendix		Language of requests
order	0: Most Availability 1: Price (Lowest) 2: Rating 3: Price (Highest) 4: Alphabetical (A-Z) 5: Alphabetical (Z-A)	No, defaults to 0	
room_type	0: Any room type 1: Private rooms 2: Shared rooms (dorms)	No, defaults to 0	
identifier	Any string	No	It will be returned in the response, enabling you to match requests and responses exactly

A sample request will look like this:

```
http://www.hostelspoint.com/webservices/affiliates/json.php?
IDSite=1&requestType=AvailSearchCity&day=05&year_month=2011-
03&nights=10&currency=1&cat=11&guests=2&city=190
```

12.1.4. Error responses

An error response will look like this:

```
{
      "dateTime":"2011-03-04 18:30:12 +0100",
      "status":false,
      "errors":[{"code":202,"text":"Authentication failure"}]
}
```

every error response will have the date it has been created, the status set to false and an array of one or more error objects containing a textual error message.

Current error codes are the following:

Code	Meaning
101	Server issues – our system cannot reply due to internal problems
201	You did not supply authentication credentials
202	Your credentials are invalid or the IP you are making calls from is not registered for our Web Service
301	Wrong request type, the request type you sent is not supported
401	The city could not be found
402	No availability in the city
403	Your request has a bad date (it is invalid or in the past)

12.1.5. Successful response

These responses will have a status set to true.

The following request:

```
http://www.hostelspoint.com/webservices/affiliates/json.php?
IDSite=33&requestType=AvailSearchCity&day=05&year_month=2011-
03&nights=10&currency=1&cat=1&guests=2&city=190&identifier=testIDe
ntifier
```

will produce a reply like the following (but with more hostels, of course):

```
{
    "response": {
        "currency": 1,
        "currencyCode": "EUR",
        "language": "en",
        "properties": {
            "2219": {
                "name": "Ostello Casale dei Monaci",
                "IDCategory": "1",
                "zip": "00043",
                "numberRatings": "22",
                "categoryName": "Hostel",
                "cityName": "Rome",
                "countryName": "Italy",
                "address": "Via Melvin Jones s.n.c.",
                "latitude": "41.7839850718694",
                "longitude": "12.6208019256592",
                "city_ID": 190,
                "rating": "60",
                "rating1": "67",
                "rating2": "65",
                "rating3": "55",
                "rating4": "47",
                "rating5": "65",
                "photoThumb60":
"http:\/\/www.hostelspoint.com\/pics\/2219\/60_002219-
1104930289.jpg",
                "photoThumb80":
"http:\/\/www.hostelspoint.com\/photos\/th-002219-1104930289.jpg",
                "photo250":
"http:\/\/www.hostelspoint.com\/pics\/2219\/250_002219-
1104930289.jpg",
                "photoBig":
"http:\/\/www.hostelspoint.com\/photos\/002219-1104930289.jpg",
                "minPrice": "20.00",
                "currency": "EUR",
                "rooms": [
                    {
                        "type": "1",
                        "roomCode": "1_1",
                        "name": "1 bed\/s private ",
                        "wc": true,
                        "availability": 1,
                        "price": "52.79",
                        "beds": "1"
                    },
                    {
                        "type": "2",
                        "roomCode": "2_229",
                        "name": "standard 6 bed\/s female dorm ",
                        "wc": false,
                        "availability": 18,
                        "price": "26.40",
                        "beds": "6",
                        "gender": "2"
                    },
                    {
                        "type": "2",
                        "roomCode": "2_266",
                        "name": "standard 4 bed\/s male dorm ",
                        "wc": true,
                        "availability": 8,
                        "price": "26.40",
                        "beds": "4",
                        "gender": "1"
                    }
                ]
            }
        }
    },
    "identifier": "testIDentifier",
    "dateTime": "2011-03-04 18:42:07 +0100",
    "status": true,
    "warnings": []
}
```

The root level of the reply has the following elements:

identifier	The identifier sent in the request, if present.	
dateTime	The server date and time when the response was created	
status	The Status of the response: if it is false, then there will be a set of error parameters.	
warnings	We are reserving a name for a “warnings” property, to be used for minor problems that do not prevent our system from responding but that nonetheless must be issued. The current implementation does not require them, but future implementations might so we encourage you to check for them just in case they come up in the future.	
response	The actual data of the response (see below)	

The response has the following root-level elements:

Element	***Value/s***	***Notes***
currency	The numerical code of the currency in our internal system	
currencyCode	The three-letter ISO code for the currency	
language	Two-letter code for the response language (Hostelsclub format, see Appendix)	
properties	An array with the list of properties matching the request, see below	

12.1.5.1. The properties array

The properties array holds as many object elements as there are properties matching the request. Each array has a key corresponding to the property ID (2219 in the example). Each object element has the following properties:

Element	***Value/s***	***Notes***
name	Property name	
IDCategory	The ID of the property's category	
zip	The property's zip or postal code	
address	The property's street address	

numberRatings	The number of ratings the property has received from our customers	
categoryName	The property's category in words, in the response's chosen language	
cityName		Please note that the request might return properties in nearby cities. Our system, especially for cities that are very near to each other, will include in the response proeprties that are within a certain range, which varies from city to city depending on its urban area conformation. If you do not wish to include these just filter only properties that have a city ID or name matching your request.
countryName		
latitude		
longitude		
city_ID	The ID of the property's city	
rating – rating1, etc.	The average customer rating and the individual values for our 5 rating categories	The 5 categories are: 1: cleanliness 2: staff 3: position 4: fun factor 5: atmosphere
photoThumb60	Link to picture: thumbnail in 60x60 pixels format	
photoThumb80	In 80x80 pixels format	
photo250	In 250x250 format	
photoBig	The biggest format available for the main picture	
minPrice	The minimum price for this property	This is the price per person
currency	The currency of the prices shown in the rooms list	
rooms	A list of the rooms that have availability	

The rooms list contains the following elements:

Element	***Value/s***	***Notes***
type	1 for private, 2 for dorm/shared room	
roomCode	Our internal room code	
name	Textual room type in request language	
wc	True, false	If the room has its own bathroom (true) or if it has shared facilities (false)
availability	How many rooms (for private rooms) or beds (for dorms) are available.	

Element	***Value/s***	***Notes***
price	Price per person	
beds	Number of beds in the room	
kingbed	true/false	Only for private rooms, if it is true the room has one big bed for two people instead of two separate beds. It is a Twin room, not a Double.
Gender	0=>mixed, 1=>male only, 2=>female only	Only for shared rooms (dorms), whether it is a mixed or single-gender room.

12.1.5.2. How to send customers to our booking system

If your interface allows customers to choose rooms on the spot, you can see the following chapter:

#11.4.4.5.How to proceed sending the customer to your personalized booking interface on our server

If it does not, please read the following chapter:

#11.4.2.2.How to proceed sending the customer to your personalized booking interface on our server

In case you have further questions, please contact us.

If you need further property details in this message or you wish to propose some message type that would suit your requirements, please contact us and we'll analyse your requirements.

13. *Appendix*

This part of the document contain a series of tables with codes for various parts of the system. The longer tables have been put in a separate spreadsheet file for your convenience. The spreadsheet file is distributed with this document. If you haven't received it, please contact our staff to get a copy.

13.1. Error Types

Both <Error> and <Warning> tags have both a *Code* attribute, which refers to the specific error code, and a *Type* attribute, that indicates in what broad category the error falls in. We've had to add a code to use for errors that do not fall into any of the existing categories.

```
<Warning Code="193" Type="11"> This reservation cannot be
cancelled</Warning>
```

The OTA allows for the following error types (table EWT - Error Warning Type) :

Error Type	***Short OTA description***	***Full OTA Description***	***Notes***
1	Unknown	Indicates an unknown error.	
2	No implementation	Indicates that the target business system has no implementation for the intended request.	
3	Biz rule	Indicates that the XML message has passed a low-level validation check, but that the business rules for the request message were not met.	
4	Authentication	Indicates the message lacks adequate security credentials	
5	Authentication timeout	Indicates that the security credentials in the message have expired	
6	Authorization	Indicates the message lacks adequate security credentials	
7	Protocol violation	Indicates that a request was sent within a message exchange that does not align to the message	
8	Transaction model	Indicates that the target business system does not support the intended transaction-oriented operation	
9	Authentical model	Indicates the type of authentication requested is not recognized	

Error Type	***Short OTA description***	***Full OTA Description***	***Notes***
10	Required field missing	Indicates that an element or attribute that is required in by the schema (or required by agreement between trading partners) is missing from the message	
11	-	-	Other errors/warnings (especially warnings), which do not fall into any of the above categories.

13.2. Error Codes

The list of error codes is provided in a separate Spreadsheet file, for ease of export and usage. If you have not received it, please contact us.

It must be noted that we had to introduce an error code (999) for situations that were not covered by the OTA codes. They are all cases of business rules/structural xml issues.

13.3. Currency codes

The list of currency codes is provided in a separate Spreadsheet file, for ease of export and usage. If you have not received it, please contact us.

13.4. Hotel Category codes

13.4.1. Property Class Type

These are the categories we use from the Property Class Type (PCT) OTA codes. They are used for *PropertyClassCode* attributes.

Code	***Description***
3	Apartment
4	Bed and breakfast
6	Campground
16	Guest house limited service
19	Hostel
20	Hotel

13.4.2. Segment Category Code

These are the Segment Category Code (SEG) OTA codes used for *SegmentCategoryCode* attributes: these are not all the OTA attributes we support during request (we try to support all those which apply to Hotels), but are the ones we return (and ones you can certainly know we will understand in your requests.

Code	***Description***
2	Budget

Code	*Description*
17	Midscale
14	Upscale
8	Luxury

When you make a request for properties falling in a certain segment, we will return properties which according to the data we have will match your criteria.

13.5. OTA Room type codes

These are used in some cases in output descriptions – it is a selection of codes from OTA GRI - Guest Room Info

Code	*Description*
10	double bedrooms
19	twin bedrooms
37	Single-bedded accommodations
36	Family/oversized accommodations

13.6. Hostelsclub Room type codes

Due to the variable nature of hostel rooms, we have a big number of room types in our system. More keep being added automatically as owners insert new room types (for ex. 27 bed dorm with shared toilet).

The codes are provided in a separate Spreadsheet file, for ease of export and usage. If you have not received it, please contact us.

In the descriptions, all rooms "with WC" contain a private WC in the room. Others have shared toilet facilities. A "shared" room is a room where people can book one or more beds, and share the room with strangers (typical of hostels). Shared rooms can be mixed (with people of all sexes allowed), only for female and only for male occupants.

13.7. Languages

These are the languages supported by our system at the time of writing. More keep being added – in general, if a language is supported in our public site, it is always supported in the XML. We try to use two-letter codes which are ISO – compatible, but when this is not possible our XML will also recognize ISO codes. Please see the section above about language recognition for details about Chinese codes (zh-tw and zh-cn).

The codes will be the same as those used in the URL of our public site. So if a new language is added and you would like to know its code, check our site. You can infer the code from our URLs. For example, the home page of our site in French is http://www.hostelsclub.com/index-fr.html, where "fr" stands for French.

The codes are provided in a separate Spreadsheet file, for ease of export and usage. If you have not received it, please contact us.

13.8. Credit Card Codes

Some credit card codes are included in the OTA specifications, but we had to add some. If a credit card type is not in this list, it means it is not accepted by our system at the moment.

Code	***Card Name***
AX	Amex
DN	Diners
JC	JCB
MC	Mastercard
VI	Visa
VE	Visa Electron
AU	Aura

13.9. City and Country Codes

City and country codes can be found downloading XML files on our site.

The address of the files is

https://www.hostelspoint.com/xml_aff/cities_LANGUAGE_CODE_HERE.xml

For example, the English version of the file has the following address:

https://www.hostelspoint.com/xml_aff/cities_en.xml

These files are updated in real time.

Please note that we have some countries which are "duplicated" in our system.

The United Kingdom is also present as Great Britain (minus Northern Ireland), and in its component countries of Northern Ireland, Wales, England and Scotland. Use the ISO code GB for search in the whole of United Kingdom.

Holland is a duplicate of the Netherlands. This has been duplicated because in many European languages people will refer to the Netherlands using the (less correct) term "Holland".