

HOTELSTON.COM

Hotelston.com API Specification

Version 1.41

Saulius Pačekajus

2015-04-23

Table of Contents

Introduction	2
Methods	3
Generic Types	4
Method Request Type	4
Price Method Request Type	4
Currency and Rates	5
Method Response Type	6
IdNameBean Type	6
Locales	7
Client Nationality	7
Integration Scenario	8
Static Data Import	10
Method getDestinationList	10
Method getHotelList	11
Method getHotelDetails	13
Hotel Search	17
Availability Check	21
Booking Terms	26
Booking	30
Cancel Booking	34
Questions?	35

Introduction

This is a specification for Hotelston.com API version 1.04.

Hotelston.com API is a standard SOAP Web Service (WS) available at URL <https://www.hotelston.com/ws/HotelService?wsdl>

Integration developers may consider using a free version of software like SoapUI (<http://www.soapui.org>) to test Hotelston.com API. Given URL to WSDL file it can build stub for WS and help to explore how all methods look like and even test them with some test data.

In order to get login details you must sign up as an agent at Hotelston.com: <https://www.hotelston.com/registerCompany.jsf>.

In case you have any questions, please contact Hotelston API Support by api@hotelston.com.

Methods

Hotelston API has 6 main methods any integration must use:

Method	Description
getDestinationList	Provides a full list of Hotelston destinations
getHotelList	Provides a list of hotels
getHotelDetails	Provides full information of a particular hotel
searchHotels	Searches for hotel offers given criteria. Can search by city or by hotel.
getBookingTerms	Provides booking terms (cancellation policy, important remarks) for a particular room choice
bookHotel	Makes a booking given criteria
cancelBooking	Cancels booking given booking reference

Full information about each method is available in further chapters.

Generic Types

Every method is defined by request and response. Every request is derived from generic type *MethodRequest* and every response is derived from a generic type *MethodResponse* (see WSDL for type declaration). This chapter reviews all generic request/ response types. Descriptions for inherited elements are skipped further in specifications of particular request/response types.

Method Request Type

Every request is derived from type *MethodRequest* and (see WSDL for type declaration).

Example of a generic request:

```
<MethodRequest>
  <locale>en</locale>
  <loginDetails email="test@hotelston.com" password="test19"/>
</MethodRequest>
```

Parameter values:

Name	Type	Description
locale	Enumeration	Locale/language in which information like room/board/distance types, or hotel descriptions is returned. Available values: <i>en, lt</i> . See section „Locales“ for more details
loginDetails	Complex	
email	String	Hotelston.com user login email
password	String	Hotelston.com user login password
transport	Deprecated	Deprecated

Price Method Request Type

Some methods like **searchHotels**, **getBookingTerms**, **bookHotel**, **cancelBooking** need to consider prices, thus require some price specific information. Hence *MethodRequest* type has been extended to *PriceMethodRequest* type.

Example of a generic price request:

```
<PriceMethodRequest>
  <locale>en</locale>
  <loginDetails email="test@hotelston.com" password="test19"/>
  <currency>EUR</currency>
  <netRates>true</netRates>
</PriceMethodRequest>
```

Parameter values:

Name	Type	Description
Same elements as in <i>MethodRequest</i> type; see section „Method Request“		
currency	Enumeration	Currency to use in response. Available values: <i>EUR, LTL</i> . See section „Currency and Rates“
netRates	Boolean	<i>true</i> if prices are NET or false if <i>prices</i> include commission. See section „Currency and Rates“
agentCompany	Element	Used only on special agreement

Currency and Rates

Every price method (namely *searchHotels*, *getBookingTerms*, *bookHotel*, *cancelBooking*) has 2 price related elements: *currency* and *netRates*. While *currency* is pretty much obvious, *netRates* represent an important property of the market – whether prices are NET (agency will be billed with that particular price) or it includes commission rate (agency will be billed with the particular price minus agreed commission rate).

Both elements represent properties for all prices met in request and response of a particular method call. For example if *currency = EUR* and *netRates = true* in *searchHotels* request, then all prices will be NET and in EUR in response. If *currency = LTL* and *netRates = false* in *bookHotel* request, then booking price will in request will be interpreted as LTL price with commission rate and you will be invoice with these details in mind.

Rule of thumb: use same *currency* and *netRates* values for the whole session – from search to booking. Whatever values have been used for search, the very same values should be used later for *getBookingTerms* and *bookHotel* methods. Otherwise the booking may result in money loss for you agency.

Method Response Type

Every method has a response type which is derived from generic type MethodResponse.

Example of generic method response:

```
<MethodResponse>
  <success>false</success>
  <error code="201" message="Locale is missing!"/>
</MethodResponse>
```

Parameter values:

Name	Type	Description
success	Boolean	Indicates whether request has succeeded or not. In case value is false; look for error element bellow.
error	Complex	
code	Integer	Error code
message	String	Error message

IdNameBean Type

This type is very important as it is a base type for most of the compound elements in Hotelston API. Elements like room type, board type, distance type, hotel feature, city, country and others are extended from type *IdNameBean*, hence has all of its properties. These common properties may be omitted later in parameter descriptions.

Example of IdNameBean element:

```
<country id="1870196" name="Spain" nameEn="Spain">
```

Parameter values:

Name	Type	Description
id	Integer	Hotelston ID of that particular element, unique in its domain. It can be used anytime later to refer to this particular element given domain.

name	String	Deprecated, use nameEn, nameLt and other name<locale> instead.
nameEn, nameLt, ...	String	Name of an element in specific locale. See section „Locales“ for more details

Locales

Every request must have at least 1 element *locale*. Multiple *locale* elements are supported too. There are certain elements in response that has names, like room type, board type, distance type and other based on IdNameBean type. Every of these elements will have name attributes like *nameEn*, *nameLt*. Given requested locale(s) Hotelston API will decide which of these name attributes will be available in response.

Currently there 2 locales are supported: *en* and *lt* as for English and Lithuanian

Client Nationality

Some methods, namely **searchHotels** and **bookHotel**, has element *clientNationality* in method request. Some hotels require passenger nationality, because they have different availability and pricing policy for passenger from different countries and/or regions. It is highly recommended to provide *clientNationality* value for every search and booking as this can prevent possible (although rare) problems during check-in at a hotel.

In case it is not possible to provide *clientNationality* – the location of the travel agency is used as default value for approximation. Even if you don't have *clientNationality* field in your search form, we recommend to provide approximate/likely client locale identified by IP address or own market/domain knowledge.

All possible *clientNationality* values can be found in WSDL file of Hotelston WebService. Please refer to enumeration type *ClientNationality*.

IMPORTANT: failure to provide the correct or any value as *clientNationality*, eliminates Hotelston responsibilities in an unlikely event of passenger being not accepted during check-in at a hotel. In this case Hotelston would do our best to help solve the problem, but any additional costs resulting from failure to provide *clientNationality* during search and reservation would not be subject for financial compensation.

Integration Scenario

Hotelston.com API is designed based on Hotelston developer experience working with hotel search and reservation APIs of many providers. Here is a scenario proven to be most efficient, while integrating any similar API:

1. Build client stub/proxy for Hotelston.com WS.
2. Import destinations – use method **getDestinationList** to get a full list of Hotelston destinations. It is possible to get a list of destinations from method **getHotelList**, but **getDestinationList** has additional so called meta-destinations, that are fully searchable although they have no hotels mapped directly.
3. Import hotel data – use methods **getHotelList** and **getHotelDetails**. As the number of hotel is huge, please note that initial static data import is going to take a while. After initial import is done, scheduled update imports (please provide last update date in **getHotelList** request) should be run at least once a week, although once a day is recommended. *If possible, please schedule full data imports between 20:00 and 05:00 GM T.*
4. Implement hotel search using method **searchHotels**. Search can be executed for a city or some particular hotel. City search is used as the start search on a site. Particular hotel search is used to check availability of a specific hotel.
5. *Optional:* implement availability check using method **checkAvailability**, if you experience many price change issues while making bookings or have this feature requirement in your system. **checkAvailability** skips over all the caches and makes sure the availability result is as accurate as possible. We recommend using it before getting to the reservation phase as well as right before making the actual booking.
6. Test hotel search. Use different dates, destinations, room configurations (single, double, triple, add children) to test that search results on your site fully matches search results at Hotelston.com online search.
7. Implement booking terms. Booking terms must be presented to the end customer before making any booking. Hotelston.com API provides a method **getBookingTerms** to get important booking remarks (like city tax information) and structured cancellation rules to generate cancellation policy text. *It is very important to present booking terms to an end user on your site before confirming any booking.*
8. Contact it@hotelston.com before you start implementing and testing booking methods. Use method **bookHotel** with *test = true* (see details regarding bookHotel method) and see if all data comes to our system correctly.
9. Hotel booking. **Important:** *When you start live testing, please choose check-in date at least 3 months in the future, check cancellation policy and make sure you will be able to cancel reservation free of charge and always cancel test bookings ASAP.*
10. Use **cancelBooking** method to implement booking cancellation. Booking reference received from **bookHotel** method must be used here.

ATTENTION 1: Please contact it@hotelston.com before testing hotel booking and cancellation methods!

ATTENTION 2: When doing live testing and making real bookings in real hotels, please note that if you book a non-refundable hotel or forget to cancel a booking, agency will be invoiced and will have to pay for these bookings. Please be careful!

Static Data Import

Static data is information you have to store on your side for efficiency. That includes destinations which are later used to make searches and hotel descriptions that you have to present given Hotelston API search response. Hotelston API only returns hotel IDs and names in all further methods.

3 methods are needed to implement static data import: **getDestinationList**, **getHotelList** and **getHotelDetails**. Please find information about each of these methods below.

Method getDestinationList

getDestinationList method is used to get the full list of Hotelston searchable destinations, including meta-destinations (usually wide-search destinations like Tenerife (id 1870796), Crete (id 1870364) or Paris (1870638)), that may or may not have directly mapped hotels.

Example request:

```
<DestinationListRequest>
  <locale>en</locale>
  <loginDetails email="test@hotelston.com" password="test19"/>
</DestinationListRequest>
```

Parameter values:

Name	Type	Description
		Same elements as in <i>MethodRequest</i> type; see section "Method Request"

Example response:

```
<DestinationListResponse>
  <success>true</success>
  <country id="1870142" name="France" nameEn="France">
    <city id="1933818" name="Dijon" nameEn="Dijon"/>
    <city id="1870638" name="Paris" nameEn="Paris"/>
    <city id="1870614" name="Nice" nameEn="Nice"/>
    <city id="1870564" name="Marseille" nameEn="Marseille"/>
    <city id="1870545" name="Lyon" nameEn="Lyon"/>
  </country>
  <country id="1870208" name="USA" nameEn="USA">
    <state id="50000281" name="California" nameEn="California">
      <city id="50013259" name="Los Angeles (Ca)" nameEn="Los Angeles
(Ca)"/>
      <city id="1870724" name="San Francisco Area CA" nameEn="San
Francisco Area CA"/>
    </state>
    <state id="50000309" name="New York" nameEn="New York">
```

```

        <city id="37289966" name="New York, Ny" nameEn="New York, Ny"/>
    </state>
</country>
<country id="1870146" name="Greece" nameEn="Greece">
    <city id="1870860" name="Zakynthos" nameEn="Zakynthos"/>
    <city id="50009289" name="Corfu Town" nameEn="Corfu Town"/>
    <city id="1870696" name="Rhodes" nameEn="Rhodes"/>
    <city id="1870247" name="Athens" nameEn="Athens"/>
    <city id="50000383" name="Faliraki (Rhodes)" nameEn="Faliraki
(Rhodes)"/>
    <city id="50000822" name="Heraklion Region" nameEn="Heraklion Region"/>
    <city id="50000725" name="Chania region" nameEn="Chania region"/>
    <city id="1870364" name="Crete" nameEn="Crete"/>
</country>
<country id="1870196" name="Spain" nameEn="Spain">
    <city id="50000558" name="Palma de Mallorca" nameEn="Palma de
Mallorca"/>
    <city id="1870265" name="Barcelona" nameEn="Barcelona"/>
    <city id="1870506" name="Lanzarote" nameEn="Lanzarote"/>
    <city id="50000576" name="Los Cristianos" nameEn="Los Cristianos"/>
    <city id="50000598" name="Los Gigantes" nameEn="Los Gigantes"/>
    <city id="1870355" name="Costa Brava/Lloret de Mar" nameEn="Costa
Brava/Lloret de Mar"/>
    <city id="50000599" name="Santa Cruz de Tenerife" nameEn="Santa Cruz de
Tenerife"/>
    <city id="1870796" name="Tenerife" nameEn="Tenerife"/>
</country>
</DestinationListResponse>

```

Parameter values:

Name	Type	Description
Same elements as in <i>MethodResponse</i> type; see section “Method Response”		
country	IdNameBean	Country
state	IdNameBean	State (optional)
city	IdNameBean	City/Destination

Method getHotelList

getHotelList method is used to get a list of hotels: full list or a list of hotels updated after some particular date.

Example request:

```

<HotelListRequest>
  <locale>en</locale>
  <loginDetails email="test@hotelston.com" password="test19"/>

```

```
<lastUpdate>2012-05-04T00:00:00</lastUpdate>
</HotelListRequest>
```

Parameter values:

Name	Type	Description
Same elements as in <i>MethodRequest</i> type; see section “Method Request”		
lastUpdate	Timestamp	Date and time of the last time Hotelston inventory has been updated. Do not include this element for initial import.

Hotels are grouped by destinations in response. So use it to import hotels as well as destinations.

Example response:

```
<HotelListResponse>
  <success>true</success>
  <country id="1870196" name="Spain" nameEn="Spain">
    <city id="1870265" name="Barcelona" nameEn="Barcelona" isoCode="BCN">
      <hotel id="33775442" name="AC Som" lastUpdated="2012-07-
28T00:04:49.427+03:00"/>
      <hotel id="33779070" name="AC Hotel Gava" lastUpdated="2012-07-
28T00:05:08.997+03:00" deleted="true"/>
      <resort id="1871142" name="Carmona" nameEn="Carmona">
        <hotel id="33794888" name="Alcazar de la Reina" lastUpdated="2012-07-
28T00:43:26.345+03:00"/>
      </resort>
    </city>
  </country>
</HotelListResponse>
```

Parameter values:

Name	Type	Description
Same elements as in <i>MethodResponse</i> type; see section “Method Response”		
country	IdNameBean	Country
city	IdNameBean	City/Destination
city:isoCode	String	IATA code of a nearest airport
resort	IdNameBean	Some district in a city or resort in a region
hotel	IdNameBean	Hotel

hotel:lastUpdated	Timestamp	Date of last time information of a particular hotel has been updated
hotel:deleted	Boolean	Indicates if this hotel is actually deleted. This attribute is meant for those, who want update all hotels instead of those updated after certain date.

Method getHotelDetails

This method is used to get all details for a particular hotel.

Example request:

```
<HotelDetailsRequest>
  <locale>en</locale>
  <loginDetails email="test@hotelston.com" password="test19"/>
  <hotelId>42450941</hotelId>
</HotelDetailsRequest>
```

Parameter values:

Name	Type	Description
Same elements as in <i>MethodRequest</i> type; see section “Method Request”		
hotelId	Integer	ID of the hotel whose details are requested You can get this id from the method getHotelList

Example response:

```
<HotelDetailsResponse>
  <success>true</success>
  <hotel id="50141221" name="Hostel 4 of Academy of Commerce">
    <description><![CDATA[<h2>Hotel Description</h2>...</p>]]></description>
    <descriptionEn><![CDATA[<h2>Hotel Description</h2>...</p>]]></descriptionEn>
    <lastUpdated>2012-04-26T07:09:43.444+03:00</lastUpdated>
    <city id="33396104" name="Lviv" nameEn="Lviv"/>
    <starRating>1.0</starRating>
    <address street1="7 CHERVONOYI KALYNY STR" street2="" city="LVIV" state=""
country="Ukraine" zip="79034">7 CHERVONOYI KALYNY STR, 79034, LVIV,
Ukraine</address>
    <email>l.goncharova@tui.ua</email>
    <distance>
      <type id="2326543" name="Center" nameEn="Center"/>
      <value>3 km</value>
```

```

</distance>
<distance>
  <type id="33775413" name="Tourist Center" nameEn="Tourist Center"/>
  <value>3 km</value>
</distance>
<distance>
  <type id="32913330" name="Bus Station" nameEn="Bus Station"/>
  <value>5 km</value>
</distance>
<distance>
  <type id="32913335" name="Train Station" nameEn="Train Station"/>
  <value>5 km</value>
</distance>
<distance>
  <type id="38860782" name="Shopping" nameEn="Shopping"/>
  <value>200 m</value>
</distance>
<distance>
  <type id="38860779" name="Restaurants" nameEn="Restaurants"/>
  <value>400 m</value>
</distance>
<distance>
  <type id="38860773" name="Bar / Pub" nameEn="Bar / Pub"/>
  <value>400 m</value>
</distance>
<distance>
  <type id="38860778" name="Public Transport" nameEn="Public Transport"/>
  <value>200 m</value>
</distance>
<coordinates latitude="49.813085" longitude="24.046165"/>
<image
url="http://www.hotelston.com/resources/hotelImages/5/0/0/0/3/8/4/1/50199809.jpg
"/>
  <image
url="http://www.hotelston.com/resources/hotelImages/5/0/0/0/3/8/4/1/50199810.jpg
"/>
  <image
url="http://www.hotelston.com/resources/hotelImages/5/0/0/0/3/8/4/1/50199811.jpg
"/>
</hotel>
</HotelDetailsResponse>

```

Parameter values:

Name	Type	Description
Same elements as in <i>MethodResponse</i> type; see section “Method Response”		
hotel	IdNameBean	
description	String	Deprecated; use descriptionEn, descriptionLt instead
descriptionEn, descriptionLt, ...	String	Hotel description given locale. May contain HTML tags (H2 for headings, P for paragraphs and UL,LI for

		bullets)
lastUpdated	Timestamp	Date of last time information of the hotel has been updated
city	IdNameBean	City/destination of the hotel
resort	IdNameBean	Resort/district of the hotel (optional)
starRating	Decimal	Official star rating of the hotel
address	Complex/String	Hotel address In all cases full address is provided as element text value. In most (not all) cases structured address details (street, city, postal code) is also provided for better formatting options.
address:street1	String	Street1 part of address (optional)
address:street2	String	Street2 part of address (optional)
address:city	String	City part of address (optional)
address:state	String	State part of address (optional)
address:country	String	Country part of address (optional)
address:zip	String	ZIP part of address (optional)
phone	String	Hotel phone number (optional)
fax	String	Hotel fax number (optional)
email	String	Hotel email address (optional)
website	String	Hotel website URL (optional)
feature	IdNameBean	Hotel feature (optional)
distance	Complex	Distance from the hotel to a certain point like city center, beach or bus stop. (optional)
distance:type	IdNameBean	Distance type – defines to what type of point the distance is provided.
distance:value	String	Distance value, like “5 km” or “10 min by foot”

coordinates	Complex	Hotel coordinates (optional)
coordinates:latitude	Float	Latitude of hotel coordinates
coordinates:longitude	Float	Longitude of hotel coordinates
image	Complex	Hotel image
image:url	String	Image URL
checkIn	Time	Standard check-in time of the hotel (optional)
checkOut	Time	Standard check-out time of the hotel (optional)
customerRating (cr)	Complex	Customer rating of the hotel (optional)
cr:overall cr:room cr:facilities cr:cleanness cr:food cr:staff cr:checkIn cr:valueForMoney	Decimal	Specific hotel rating values (optional)
cr:customerCount	Integer	Number of customers whose ratings have been considered
customerComments (cc)	Complex	Customer comments (optional)
cc:comment	Complex	Comment
cc:comment:positives cc:comment:negatives cc:comment:general cc:comment:location	String	Specific aspects of customer comment (optional)
cc:comment:date	Date	Date of comment (optional)
cc:comment:rating	Decimal	Rating provided with comment (optional)

TIP: most of the hotel details are optional, please make sure your import can cope with that.

Hotel Search

Having static data imported, another important step is to implement hotel search. **searchHotels** method must be used for that and it can be executed on city/destination or specific hotel. If *hotelId* is provided, search is executed on hotel (*cityId* does not matter then).

Example request:

```
<SearchHotelsRequest>
  <locale>en</locale>
  <loginDetails email="test@hotelston.com" password="test19"/>
  <currency>EUR</currency>
  <netRates>true</netRates>
  <criteria>
    <checkIn>2012-10-27</checkIn>
    <checkOut>2012-11-03</checkOut>
    <cityId>1870359</cityId>
    <clientNationality>DE</clientNationality>
    <room adults="2" children="0"/>
    <room adults="2" children="2">
      <childAge>1</childAge>
      <childAge>5</childAge>
    </room>
  </criteria>
</SearchHotelsRequest>
```

Parameter values:

Name	Type	Description
Same elements as in <i>PriceMethodRequest</i> type; see section “Price Method Request”		
criteria	Complex	Search criteria
checkIn	Date	Check-in date
checkOut	Date	Check-out date
cityId	Integer	City ID
hotelId	Integer	Hotel ID
clientNationality	Enumeration	Nationality of the client. Available values: refer to type <i>ClientNationality</i> in WSDL document; see section “Client Nationality” for more information
room	Complex	Room criteria Multiple instances possible! Min 1, max 5

room:adults	Integer	Number of adults in room (min 1, max 6)
room:children	Integer	Number of children in room (min 0, max 3)
room:childAge	Integer	Age of a single child. There must be as many childAge elements in room, as children attribute declares. Value: min 0, max 14

Example response:

```
<SearchHotelsResponse>
  <success>true</success>
  <searchId>51026043</searchId>
  <hotel id="41057732" name="N-Ch" lastUpdated="2012-04-26T05:01:34.351+03:00">
    <room seqNo="0" id="37307017" specialOffer="false" price="700.47">
      <boardType groupId="30309844" groupName="Room Only" id="2359299"
name="Room Only" nameEn="Room Only"/>
      <roomType id="36292256" name="Double Standard" nameEn="Double Standard"/>
    </room>
    <room seqNo="0" id="37307017" specialOffer="false" price="700.47">
      <boardType groupId="30309844" groupName="Room Only" id="2359299"
name="Room Only" nameEn="Room Only"/>
      <roomType id="36292256" name="Double Standard" nameEn="Double Standard"/>
    </room>
  </hotel>
</SearchHotelsResponse>
```

Parameter values:

Name	Type	Description
Same elements as in <i>MethodResponse</i> type; see section “Method Response”		
searchId	Integer	Search ID <i>Save it for booking terms request!</i>
hotel	Complex	Represents a single hotel in a search result list
id	Integer	Hotel ID
name	String	Hotel name
lastUpdated	Timestamp	Date of last time information of the hotel has been updated
room	Complex	Offer for a room

room:seqNo	Integer	Room sequence number starting with 0 for the first requested room.
room:id	Integer	ID of a particular price of a room in the hotel. <i>Save it for booking and other requests!</i>
room:specialOffer	Boolean	Indicates whether this is a special offer or not
room:originalPrice	Decimal	Offer price before discounts (if any applied)
room:price	Decimal	Total price of a room offer
room:visaSupport	Boolean	Indicates whether Hotelston provides Visa Support for this offer
room:boardType (board)	Complex IdNameBean	Board (meal) type of the offer Since there are many board types for breakfast, half board etc, Hotelston have them grouped so that English Breakfast and Continental Breakfast both get into group Breakfast which means, that breakfast are offered for that room choice. <i>Save id for booking and other requests!</i>
board:groupId	Integer	Board type group ID
board:groupName	String	Board type group name
room:roomType	IdNameBean	Room type of the offer <i>Save id for booking and other requests!</i>
room:specificSpecialOffer (sso)	Complex	Specific special offer (if any) applied to the offer. May be multiple entries.
sso:type	Enumeration	Special offer type Possible values: SPECIAL_OFFER, LAST_MINUTE, FREE_NIGHTS, KIDS_STAY_FREE, FREE_BOARD, EARLY_BOOKING, GOOD_DEAL, PACKAGE_RATE
sso:details	String	Additional details for Specific Special

		Offer. Example: type FREE_NIGHTS, details “Stay 4 nights, pay for 3”
--	--	--

TIP: save room ID, board type ID, room type ID and room price for every offer as you will need them for method **bookHotel**. Save search ID as you will need it for method **getBookingTerms**.

Availability Check

In order to provide better search response times, caching is used in multiple layers. However in some situations it may results in lower accuracy in hotel prices and availabilities. Therefore availability check is recommended before starting the reservation process (before asking for the pax details) and right before reservation confirmation, especially when the process takes more than 5 minutes. Method **checkAvailability** is meant exactly for that and allows handling price change situations.

Example request:

```
<CheckAvailabilityRequest>
  <locale>en</locale>
  <loginDetails email="test@hotelston.com" password="test19"/>
  <currency>EUR</currency>
  <netRates>true</netRates>
  <criteria>
    <checkIn>2015-06-23</checkIn>
    <checkOut>2015-06-27</checkOut>
    <hotelId>50509298</hotelId>
    <clientNationality>IT</clientNationality>
    <room adults="2" children="2">
      <roomId>169817094</roomId>
      <roomTypeId>53107680</roomTypeId>
      <boardTypeId>2359299</boardTypeId>
      <childAge>2</childAge>
      <childAge>5</childAge>
    </room>
  </criteria>
</CheckAvailabilityRequest>
```

Parameter values:

Name	Type	Description
Same elements as in <i>PriceMethodRequest</i> type; see section “Price Method Request”		
criteria	Complex	Search criteria
checkIn	Date	Check-in date
checkOut	Date	Check-out date
hotelId	Integer	Hotel ID
clientNationality	Enumeration	Nationality of the client. Available values: refer to type <i>ClientNationality</i> in WSDL document; see section “Client Nationality” for more information
room	Complex	Room criteria

		Multiple instances possible! Min 1, max 5
room:adults	Integer	Number of adults in room (min 1, max 6)
room:children	Integer	Number of children in room (min 0, max 3)
room:childAge	Integer	Age of a single child. There must be as many childAge elements in room, as children attribute declares. Value: min 0, max 14
room:roomId	Integer	Room ID of the offer
room:roomTypeId	Integer	Room type ID of the offer
room:boardTypeId	Integer	Board type ID of the offer

Example response:

```
<CheckAvailabilityResponse>
  <success>true</success>
  <hotel id="50509298" name="Aletto Jugendhotel Kreuzberg" lastUpdated="2015-01-14T20:41:02.298+02:00">
    <room seqNo="0" id="169817094" specialOffer="false" price="385.56" visaSupport="false">
      <boardType groupId="30309844" groupName="Room Only" id="2359299" name="Room Only" nameEn="Room Only"/>
      <roomType id="53107680" hotelstonName="Room Quad - 4 Beds" name="Quadruple Room (4 Beds)" nameEn="Quadruple Room (4 Beds)"/>
      <bookingTerms seqNo="0" nonRefundable="true">
        <cancellationPolicy cxlDate="2015-03-30">
          <cancellationRule>
            <cancellationDeadline timeUnit="DAY" effectMoment="BEFORE_ARRIVAL" amount="84"/>
            <cancellationPenalty penaltyUnit="PERCENTAGE" amount="100"/>
          </cancellationRule>
        </cancellationPolicy>
      </bookingTerms>
    </room>
  </hotel>
</CheckAvailabilityResponse>
```

Parameter values:

Name	Type	Description
Same elements as in <i>MethodResponse</i> type; see section “Method Response”. Generally the same structure as in <i>SearchHotelsResponse</i> type with the addition of the <i>bookingTerms</i>		

element of the same type as in <i>BookingTermsResponse</i> .		
hotel	Complex	Represents a single hotel in a search result list. Same as in <i>SearchHotelsResponse</i>
id	Integer	Hotel ID
name	String	Hotel name
lastUpdated	Timestamp	Date of last time information of the hotel has been updated
room	Complex	Offer for a room
room:seqNo	Integer	Room sequence number starting with 0 for the first requested room.
room:id	Integer	ID of a particular price of a room in the hotel. <i>Save it for booking and other requests!</i>
room:specialOffer	Boolean	Indicates whether this is a special offer or not
room:originalPrice	Decimal	Offer price before discounts (if any applied)
room:price	Decimal	Total price of a room offer
room:visaSupport	Boolean	Indicates whether Hotelston provides Visa Support for this offer
room:boardType (board)	Complex IdNameBean	Board (meal) type of the offer Since there are many board types for breakfast, half board etc, Hotelston have them grouped so that English Breakfast and Continental Breakfast both get into group Breakfast which means, that breakfast are offered for that room choice. <i>Save id for booking and other requests!</i>
board:groupId	Integer	Board type group ID
board:groupName	String	Board type group name
room:roomType	IdNameBean	Room type of the offer

		<i>Save id for booking and other requests!</i>
room:specificSpecialOffer (sso)	Complex	Specific special offer (if any) applied to the offer. May be multiple entries.
sso:type	Enumeration	Special offer type Possible values: SPECIAL_OFFER, LAST_MINUTE, FREE_NIGHTS, KIDS_STAY_FREE, FREE_BOARD, EARLY_BOOKING, GOOD_DEAL, PACKAGE_RATE
sso:details	String	Additional details for Specific Special Offer. Example: type FREE_NIGHTS, details “Stay 4 nights, pay for 3”
room:bookingTerms (bt)	Complex	Booking terms for a specific room Same as in <i>BookingTermsResponse</i>
bt:seqNo	Integer	Room sequence number starting with 0 for the first requested room.
bt:nonRefundable	Boolean	A helper attribute: if its value is <i>true</i> , this booking is nonrefundable and there is no need to parse the cancellation policy. IMPORTANT: please note, if the value is false, it does not mean that cancellation is refundable – it is necessary to parse cancellation policy then.
bookingRemarks	String	Important booking remarks that, like city tax information or notifications about special conditions
cancellationPolicy (cxl)	Complex	Cancellation policy for a specific room choice. It contains rules based on which cancellation penalties are defined.
cxl:cancellationRule (cr)	Complex	A single rule of cancellation policy Multiple instances possible!
cxl:cancellationDeadline (cd)	Complex	Defines a time moment when the rule is applied

cd:timeUnit	Enumeration	Time unit Possible values: FULL_DURATION, YEAR, MONTH, WEEK, DAY, HOUR, SECOND
cd:effectMoment	Enumeration	Defines how is the time unit applied Possible values: BEFORE_ARRIVAL, AFTER_BOOKING For example, it may be “2 days after booking” or “72 hours before arrival”
cd:amount	Integer	Amount of time defined by time unit
cr:cancellationPenalty (cp)	Complex	Defines a penalty (cancellation fee) when the rule is applied
cp:penaltyUnit	Enumeration	Penalty unit Possible values: NUMBER_OF_NIGHTS, PERCENTAGE, MONEY_AMOUNT, PERCENTAGE_OF_FIRST_NIGHT
cp:amount	Decimal	Amount of the penalty If penaltyUnit is set to NUMBER_OF_NIGHTS, then this will be the number of nights. Penalty will be the price of a room for these nights. If penaltyUnit is set to PERCENTAGE, then this will be a percent of a total value for this room booking which will be charged if this cancellation rule is applied.

Booking Terms

Booking terms are very important part of the booking. There are 2 main elements: booking remark (like “Please note Rome city tax 3 EUR per person per night”) and cancellation policy which describes what are the possibilities to cancel the booking when it’s made: deadlines for free cancellation (if any) and penalties for cancellations after deadline or in case of no-show.

IMPORTANT: All of these terms must be presented to an end customer upon the booking and (according to new EU regulations) presented in a voucher end customer receives.

Method **getBookingTerms** provides all this necessary information.

Example request:

```
<BookingTermsRequest>
  <locale>en</locale>
  <loginDetails email="test@hotelston.com" password="test19"/>
  <currency>EUR</currency>
  <netRates>true</netRates>
  <hotelId>41057732</hotelId>
  <searchId>51026043</searchId>
  <room>
    <roomId>37307017</roomId>
    <roomTypeId>36292256</roomTypeId>
    <boardTypeId>2359299</boardTypeId>
  </room>
</BookingTermsRequest>
```

Parameter values:

Name	Type	Description
Same elements as in <i>PriceMethodRequest</i> type; see section “Price Method Request”		
hotelId	Integer	Hotel ID
searchID	Integer	Search ID from searchHotels method
room	Complex	Room offer details All IDs come from searchHotels method response
room:roomId	Integer	Room ID of the offer
room:roomTypeId	Integer	Room type ID of the offer
room:boardTypeId	Integer	Board type ID of the offer

Example response:

```
<BookingTermsResponse>
  <success>true</success>
  <bookingTerms seqNo="0" nonRefundable="false">
    <bookingRemarks>Please note Rome city tax 3 Euro per person per
night!</bookingRemarks>
    <cancellationPolicy>
      <cancellationRule>
        <cancellationDeadline timeUnit="DAY" effectMoment="BEFORE_ARRIVAL"
amount="5"/>
        <cancellationPenalty penaltyUnit="PERCENTAGE" amount="30"/>
      </cancellationRule>
      <cancellationRule>
        <cancellationDeadline timeUnit="DAY" effectMoment="BEFORE_ARRIVAL"
amount="3"/>
        <cancellationPenalty penaltyUnit="PERCENTAGE" amount="50"/>
      </cancellationRule>
      <cancellationRule>
        <cancellationDeadline timeUnit="DAY" effectMoment="BEFORE_ARRIVAL"
amount="1"/>
        <cancellationPenalty penaltyUnit="PERCENTAGE" amount="100"/>
      </cancellationRule>
    </cancellationPolicy>
  </bookingTerms>
</BookingTermsResponse>
```

Parameter values:

Name	Type	Description
Same elements as in <i>MethodResponse</i> type; see section “Method Response”		
bookingTerms (bt)	Complex	Booking terms for specific room
bt:seqNo	Integer	Room sequence number starting with 0 for the first requested room.
bt:nonRefundable	Boolean	A helper attribute: if its value is <i>true</i> , this booking is nonrefundable and there is no need to parse the cancellation policy. IMPORTANT: please note, if the value is false, it does not mean that cancellation is refundable – it is necessary to parse cancellation policy then.
bookingRemarks	String	Important booking remarks that, like city tax information or notifications about special conditions
cancellationPolicy (cxl)	Complex	Cancellation policy for a specific room choice. It contains rules based on which cancellation penalties are defined.

cxl:cancellationRule (cr)	Complex	A single rule of cancellation policy Multiple instances possible!
cxl:cancellationDeadline (cd)	Complex	Defines a time moment when the rule is applied
cd:timeUnit	Enumeration	Time unit Possible values: FULL_DURATION, YEAR, MONTH, WEEK, DAY, HOUR, SECOND
cd:effectMoment	Enumeration	Defines how is the time unit applied Possible values: BEFORE_ARRIVAL, AFTER_BOOKING For example, it may be “2 days after booking” or “72 hours before arrival”
cd:amount	Integer	Amount of time defined by time unit
cr:cancellationPenalty (cp)	Complex	Defines a penalty (cancellation fee) when the rule is applied
cp:penaltyUnit	Enumeration	Penalty unit Possible values: NUMBER_OF_NIGHTS, PERCENTAGE, MONEY_AMOUNT, PERCENTAGE_OF_FIRST_NIGHT
cp:amount	Decimal	Amount of the penalty If penaltyUnit is set to NUMBER_OF_NIGHTS, then this will be the number of nights. Penalty will be the price of a room for these nights. If penaltyUnit is set to PERCENTAGE, then this will be a percent of a total value for this room booking which will be charged if this cancellation rule is applied.

TIP: As this cancellation policy structure may be a bit difficult to understand from the first sight, here is the text you should be able to format given cancellation policy in the example above:

Cancellation made 5 days before arrival or later will be charged 30% of the booking value;

Cancellation made 3 days before arrival or later will be charged 50% of the booking value;

Cancellation made 1 day before arrival or in case of no-show will be charged 100% of the booking value.

IMPORTANT: As of 2015-01-01 all cancellation policies should come with *cd:timeUnit = DAY*, *cd:effectMoment = BEFORE_ARRIVAL* and *cc:penaltyUnit = PERCENTAGE*. Old values are left for backward compatibility and will be removed in future versions.

Booking

ATTENTION 1: before making any kind of booking testing, please contact it@hotelston.com first!

When testing booking please consider these requirements:

1. Before starting real tests, use parameter `test = true` in **bookHotel** response and see if all data comes to Hotelston.com system correctly. Check it at Hotelston.com website. These bookings must have status *New* and does not need to be cancelled.
2. When you start live testing (`test = false`), please choose check-in date at least 3 months in the future, check cancellation policy and make sure you will be able to cancel reservation free of charge and always cancel test bookings ASAP.

ATTENTION 2: When doing live testing and making real bookings in real hotels, please note that if you book a non-refundable hotel or forget to cancel the booking, agency will be invoiced and will have to pay for these bookings. Please be careful!

Example request:

```
<BookHotelRequest>
  <locale>en</locale>
  <loginDetails email="test@hotelston.com" password="test19"/>
  <currency>EUR</currency>
  <netRates>true</netRates>
  <hotelId>50064945</hotelId>
  <checkIn>2012-06-26</checkIn>
  <checkOut>2012-07-02</checkOut>
  <agentReferenceNumber>HT000000</agentReferenceNumber>
  <confirmedBooking>true</confirmedBooking>
  <clientNationality>DE</clientNationality>
  <contactPerson title="MR" firstname="Contact" lastname="Person"
email="contact@person.com" phone="+37060000000"/>
  <room>
    <roomId>50264107</roomId>
    <roomTypeId>38862559</roomTypeId>
    <boardTypeId>2359298</boardTypeId>
    <price>385.38</price>
    <adult title="MR" firstname="Pax1" lastname="Passenger1"/>
    <adult title="MRS" firstname="Pax2" lastname="Passenger2"/>
  </room>
</BookHotelRequest>
```

Parameter values:

Name	Type	Description
Same elements as in <i>PriceMethodRequest</i> type; see section “Price Method Request”		
hotelId	Integer	Hotel ID
checkIn	Date	Check-in date

checkOut	Date	Check-out date
agentReferenceNumber	String	Agent (your) booking reference number Must be unique for the agent!
confimerdBooking	Boolean	Always use <i>true</i>
testBooking	Boolean	If set to <i>true</i> , Hotelston.com will only create a new booking without actually confirming it. Set to <i>false</i> when doing real bookings.
clientNationality	Enumeration	Nationality of the client. Available values: refer to type <i>ClientNationality</i> in WSDL document. See section “Client Nationality” for more information
contactPerson (cp)	Complex	Contact person for the booking Can bean lead passenger or responsible agent who has direct contact with the lead passenger as these contacts may be necessary in emergency situations
cp:title	Enumeration	Possible values: MR, MRS, MS, MISS
cp:firstname	String	Contact person firstname
cp:lastname	String	Contact person firstame
cp:email	String	Contact person email <i>Please note it will be validated. Make sure you have validation on your site.</i>
cp:phone	String	Contact person phone <i>Please note it will be validated. Must be in international format +XXXXXXXX like +37061551519.</i>
room	Complex	Room booking details
room:roomId	Integer	Room offer id from searchHotels response
room:roomTypeId	Integer	Room type id from searchHotels

		response
room:boardTypeId	Integer	Board type id from searchHotels response
room:price	Decimal	Room offer price as from searchHotels response This price is used to check availability. The booking will fail if the price has increased.
room:adult (a)	Complex	Adult passenger
a:title	Enumeration	Passenger title Possible values: MR, MRS, MS, MISS
a:firstname	String	Passenger first name
a:lastname	String	Passenger surname
room:child (c)	Complex	Child passenger
c:firstname	String	Child first name
c:lastname	String	Child surname
c:age	Integer	Child age Must be the same as in search, otherwise Hotelston.com is not responsible for the consequences.

TIP: Hotelston.com API uses UTF-8 encoding, so you can use local characters in passenger names.

Example response:

```
<BookHotelResponse>
  <success>true</success>
  <bookingReference>T00000000</bookingReference>
  <room seqNo="0">
    <providerDetails name="Booking Office" bookingReference="DZ35S4"/>
  </room>
</BookHotelResponse>
```

Parameter values:

Name	Type	Description
Same elements as in <i>MethodResponse</i> type; see section “Method Response”		

bookingReference	String	Hotelston.com booking reference
room	Complex	Booking information for specific rooms
room:seqNo	Integer	Room sequence number starting with 0 for the first requested room.
room:providerDetails (pd)	Complex	Reservation provider details. Must be present in the voucher as it may be necessary for a hotel to locate the booking.
name	String	Provider name
bookingReference	String	Provider booking reference
emergencyPhone	String	Provider emergency phone number

Cancel Booking

Booking cancellation is quite simple and obvious.

Example request:

```
<CancelHotelBookingRequest>
  <locale>en</locale>
  <loginDetails email="test@hotelston.com" password="test19"/>
  <currency>EUR</currency>
  <netRates>true</netRates>
  <bookingReference>T00000000</bookingReference>
</CancelHotelBookingRequest>
```

Parameter values:

Name	Type	Description
Same elements as in <i>PriceMethodRequest</i> type; see section “Price Method Request”		
bookingReference	String	Hotelston.com booking reference

Example response:

```
<CancelHotelBookingResponse>
  <success>true</success>
  <cancellationFee>30.46</cancellationFee>
</CancelHotelBookingResponse>
```

Parameter values:

Name	Type	Description
Same elements as in <i>MethodResponse</i> type; see section “Method Response”		
cancellationFee	Decimal	Cancellation fee if any applied.

Questions?

In case you need comments or have questions, please do not hesitate to contact it@hotelston.com. Please include “XML <agent name>” (e.g. “XML Hotelston - Data import”) in subject.