



Travel Studio XML API Specification

Version 3.5.8

1	Introduction.....	3
1.1	Testing Environment IP Addresses	9
1.2	Development Timetable.....	9
1.3	XML Standards, Upgrades and Compatibility	10
2	XML API INTERFACE.....	11
2.1	Communicating with the Travel Studio API	11
2.2	API Supported Messages.....	11
3	API Static Data Requests	13
3.1	The GeoTree API request	13
3.2	The Geo_Tree_Response	13
3.3	The Facilities API request.....	14
3.4	The Facilities API response.....	14
3.5	HotelList API request.....	14
3.6	HotelList API response	15
3.7	Hotel Details API Request	16
3.8	Hotel Details API Response	16
3.9	Service Images.....	18
4	API Service Search and Availability Requests.....	19
4.1	Services Search API on given criteria	19
4.2	Service Search Response	24
4.3	Availability and Pricing Search	29
4.4	Availability and Pricing Search Response	32
5	API Booking Request	36
5.1	Booking Request Structure for Hotel API	36
5.2	Booking Response	43
5.3	Booking Details Request	43
5.4	Booking Details Response	44
5.5	Booking Statues	50
6	API Cancellation Process	52
6.1	Cancellation Policy Request.....	52
6.2	Cancellation Policy Response	54
6.3	Cancellation Request	59
6.4	Cancellation Response.....	59
7	Error Handling.....	61
8	Appendices	62
8.1	Booking conversation	62
8.2	Service Search Request and Response.....	62
8.3	Live Static Data Process	76
8.4	Room Occupancies	80

1 Introduction

This specification outlines the API that will be available to allow information to be passed to and from the Travel Studio system used by a Tour Operator. The API will be accessible via the Http post/get protocol on a non-secure IP address. A data set will be provided to API-Client by Tour Operators; this will include all the hotels that the API-Client will be able to book through this interface. The data that will be passed to the API-Client is included in this document. We are assuming that this data will then be loaded into the API-Client system and that by using the unique ID's associated with each Hotel the API-Client will be able to perform various queries on the Tour Operators data.

The API will allow the following functions:

- Find the room price and availability of a specific hotel
- Search for hotels meeting specified criteria, i.e. dates, location and star rating.
- Create a new booking for a specific hotel
- Fetch details of a booking by giving a unique booking reference number.
- Fetch booking cancellation policy by giving a unique booking reference number
- Cancel an existing Booking
- All APIs support the searching, booking and modification of 'on-request' hotels by providing certain parameters in XML. The APIs are backward compatible so if no parameters are provided for the searching/ booking etc of 'on-request' hotels, the API would still work on the basis of 'available' only hotels. This is an additional, optional parameter set.

API Flow:

The API Client can use the APIs in a variety of ways; the most common approaches are listed below.

Minimum Required HTTP Timeout settings:

- Service search timeout connect = 30 sec
- Cancellation policy timeout connect = 30 sec
- Booking timeout connect = 60 sec
- Cancel booking timeout connect = 30 sec

New Booking

Approach A:

- Client searches hotels on the API-Client website using criteria entered by the user.
- API-Client will query their own database and determine which hotels they want to get price and availability information for.
- API-Client will send a query to the Tour Operator's Travel Studio system with the ID's of the hotels that they require further information on.
- Travel Studio will return the price and availability information for each hotel requested.
- API-Client will display, to the user, a list of available hotels.

- The Client will select a hotel from the list of available hotels and view the availability and price of each room type.
- Client can request cancellation policy at room type.
- The Client can book a specific room type
- API-Client will validate the clients credit card and if approved will issue a booking Confirmation to Travel Studio
- The Travel Studio system will return a Booking Reference number to confirm the booking transaction.
- Cancellation policy can be retrieved for booking

Approach B:

- Client searches hotels on Tour Operator's system using search service API. The API returns hotels along with their prices and availability.
- API-Client displays all hotels to its customers. The Client will select a hotel from the list of available hotels and view the availability and price of each room type.
- Client can request cancellation policy at room type.
- The Client can book a specific room type.
- API-Client will validate the clients credit card and if approved will issue a booking Confirmation to Travel Studio
- The Travel Studio system will return a Booking Reference number to confirm the booking transaction.
- Cancellation policy can be retrieved for booking

Recommend booking flow:

1. Service Search

Location Search Methods:

SERVICE_SEARCH_REQUEST can be utilized in two methods, both being efficient in response.

1. Geo Location based search

```
<SERVICE_SEARCH_REQUEST>
  <VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="XMLFileName"
  LICENCE_KEY="Key" TS_API_VERSION="TSAPIVersion">
    <XML_VERSION_NO>3.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <GEO_LOCATION_NAME>London</GEO_LOCATION_NAME>
  <START_DATE>20 Dec 12</START_DATE>
  <NUMBER_OF_NIGHTS>0</NUMBER_OF_NIGHTS>
  <AVAILABLE_ONLY>true</AVAILABLE_ONLY>
  <GET_START_PRICE>true</GET_START_PRICE>
```

Geo Locations passed must match values given in GEO_TREE_RESPONSE to be valid.

The API will return services (up to 300 at a time) from the geo location requested, and any geo locations underneath that location.

For example below shows that a search on “London” will return services from “London – Outer Zones”, London Zones 1 & 2”, London Zones 3,4,5 and “London Zones 6 & 7” and all the related child nodes of those nodes.



The further down the GeoTree hierarchy you go, the more specific the search.

2. Service ID based search

A comma separated list of service Ids, (up to 340 at a time) can be passed for a more refined search by sorting services locally and requesting preselected services:

```
<SERVICE_SEARCH_REQUEST>
<VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="XMLFileName" LICENCE_KEY="Key"
TS_API_VERSION="TSAPIVersion">
<XML_VERSION_NO>3.0</XML_VERSION_NO>
</VERSION_HISTORY>
  <SERVICEIDs>12035,12036,10452,12040</SERVICEIDs>
<START_DATE>20 Dec 12</START_DATE>
```

Room Search Methods:

The API provides room based search by default but there are some methods that can be used to enforce a type of passenger based search. This is useful when requiring multiple room options to provide the most choice to clients.

ALL ROOM/ ANY ROOM

ALL ROOM and ANY ROOM tags provide a limited ability to enforce a logical OR clause to allow for multiple room types to return grouped by occupancy and quantity.

For example Hotel A and B have the following total rooms available for three days.

Hotel	Dec 20 th Rooms available	Dec 21 st Rooms available	Dec 22 nd Rooms available
1234 – Hotel A	1	1	1
5678 – Hotel B	2	2	2

This means any rooms contracted to the hotel can be sold up to the specified maximum during each day. Both hotels have Single, Twin and Double room types loaded therefore Hotel A can sell one of each room type each day and Hotel B can sell any room combination up to a maximum of two each day.

ALL ROOM

The default API behavior is to check all rooms provided are available to return a valid result set

```

<SERVICE_SEARCH_REQUEST>
<VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="XMLFileName" LICENCE_KEY="Key"
TS_API_VERSION="TSAPIVersion">
<XML_VERSION_NO>3.0</XML_VERSION_NO>
</VERSION_HISTORY>
<SERVICEIDs>1234, 5678</SERVICEIDs>
<START_DATE>20 Dec 12</START_DATE>
<NUMBER_OF_NIGHTS>2</NUMBER_OF_NIGHTS>
<AVAILABLE_ONLY>true</AVAILABLE_ONLY>
<GET_START_PRICE>true</GET_START_PRICE>
<ROOM_REPLY>
<ALL_ROOM/>
</ROOM_REPLY>
<ROOMS_REQUIRED>
<ROOM>
<OCCUPANCY>2</OCCUPANCY>
<QUANTITY>1</QUANTITY>
</ROOM>
<ROOM>
<OCCUPANCY>3</OCCUPANCY>
<QUANTITY>1</QUANTITY>
</ROOM>
</ROOMS_REQUIRED>
</SERVICE_SEARCH_REQUEST>

```

Example above returns Hotel B(5678) because 2 rooms are available on Dec 20th etc, filling the criteria for both Twin and Double search. Hotel A is not included with only one room available.

ANY ROOM

```

<ROOM_REPLY>
<ANY_ROOM/>
</ROOM_REPLY>
<ROOMS_REQUIRED>
<ROOM>
<OCCUPANCY>2</OCCUPANCY>
<QUANTITY>1</QUANTITY>
</ROOM>
<ROOM>
<OCCUPANCY>3</OCCUPANCY>
<QUANTITY>1</QUANTITY>
</ROOM>
</ROOMS_REQUIRED>
</SERVICE_SEARCH_REQUEST>

```

Example above now returns both Hotel A and Hotel B because ANY_ROOM locates hotels that have any of the specified room types available **in the quantity requested**. In this case one twin OR one double is required – Hotel A can provide this one room occupancy.

Examples:

Twin OR Double (returns Hotel A and B):

```
<ROOM_REPLY>
  <ANY_ROOM/>
</ROOM_REPLY>
<ROOMS_REQUIRED>
  <ROOM>
    <OCCUPANCY>2</OCCUPANCY>
    <QUANTITY>1</QUANTITY>
  </ROOM>
  <ROOM>
    <OCCUPANCY>3</OCCUPANCY>
    <QUANTITY>1</QUANTITY>
  </ROOM>
</ROOMS_REQUIRED>
```

2 Twins OR 2 Doubles (returns Hotel B):

```
<ROOM_REPLY>
  <ANY_ROOM/>
</ROOM_REPLY>
<ROOMS_REQUIRED>
  <ROOM>
    <OCCUPANCY>2</OCCUPANCY>
    <QUANTITY>2</QUANTITY> - Two rooms required
  </ROOM>
  <ROOM>
    <OCCUPANCY>3</OCCUPANCY>
    <QUANTITY>2</QUANTITY> Two rooms required
  </ROOM>
</ROOMS_REQUIRED>
```

2 Twins OR 2 Doubles OR 2 Singles (returns Hotel B):

```
<ROOM_REPLY>
  <ANY_ROOM/>
</ROOM_REPLY>
<ROOMS_REQUIRED>
  <ROOM>
    <OCCUPANCY>2</OCCUPANCY>
    <QUANTITY>2</QUANTITY> - Two rooms required
  </ROOM>
  <ROOM>
    <OCCUPANCY>3</OCCUPANCY>
    <QUANTITY>2</QUANTITY> Two rooms required
  </ROOM>
  <ROOM>
    <OCCUPANCY>1</OCCUPANCY>
    <QUANTITY>2</QUANTITY> Two rooms required
  </ROOM>
</ROOMS_REQUIRED>
```

2. Cancellation Policy

Once result set returns, cancellation policy should be obtained prior to booking. The finer points for understanding cancellation policies are provided in JacTravel development documents but policies are assigned across room types for the same stay/pay periods. For example if booking Single and Twin room for 3 nights, only one cancellation policy request would be required to retrieve the policy to be assigned:

E.G. Option 1 – Single room
Option 2 – Twin room

Nov 20th to 23rd:

Rather than requesting 2x policy requests for options 1 and 2, instead send one request for either option using the stay-pay period. The policy returned will be applied to the booking being made for Single and Twin room.

```
<CANCELLATION_POLICY_DETAILS_REQUEST>
  <VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="XMLFileName" LICENCE_KEY="key"
  TS_API_VERSION="TSAPIVersion">
    <XML_VERSION_NO>3.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <BODY>
    <SERVICE_CHARGE>
      <OPTION_ID>1</OPTION_ID>
      <START_DATE>20 Nov 12</START_DATE>
      <END_DATE>23 Nov 12</END_DATE>
    </SERVICE_CHARGE>
  </BODY>
</CANCELLATION_POLICY_DETAILS_REQUEST>
```

3. Pre-Booking availability check

Prior to any bookings it's strongly recommended to perform a final availability check. This allows for the inevitable delay between original search and final booking which can introduce errors from other clients reserving the same hotel and travel dates. If Service Search result sets are cached for any amount of time this final check will prevent unexpected loss in availability.

4. Booking

With step 3 in place ensuring availability and price, cancellation policy already known, booking should function without error. We recommend an HTTP timeout period for booking at 60 secs to allow for times of peak traffic.

1.1 Testing Environment IP Addresses

Three Live Environment IP Addresses for production

One Testing Environment IP Address for testing

The live environment incorporates a load-balanced search farm in order to cope with search demand, so while all the API interfaces on the test environment exist on a single IP address, the live environment splits booking operations (booking, cancellation etc) and search operations between two different IP addresses. These two addresses will be assigned to you when your account is migrated from test to production. A third IP is assigned for the obtaining of static data.

1.2 Development Timetable

A 4 week development plan provides the required support for the clients to get their system on the production environment. During this period all questions should be posted to apitest@jactravel.co.uk, we endeavor to answer all questions within 24hrs although normal response time is much faster.

The XML/API can be programmed within a 2 week period by an experienced developer which leaves a solid two weeks for testing and debugging.

JacTravel will check in once a week to see how development is progressing and to answer any further queries that have arisen.

1.3 XML Standards, Upgrades and Compatibility

The following information outlines the XML standards that will be adopted in all the supported messages. The following table shows the standard header information, which will be used:

Name	Description
VERSION_HISTORY	Tag Structure that holds the versioning information. This tag is not fully functional, except for the licence key tag but is added to support future versioning information.
APPLICATION_NAME	Name of the application using this XML file. API client would add "API-Client" to this tag. Not functional currently.
XML_VERSION_NO	Version number of XML file. Open Destinations will provide this. Not currently functional.
XML_FILE_NAME	Name of XML file. For example <i>AvailabilitySearchRequest.XML</i> . Not currently functional.
LICENCE_KEY	The Licence key will be provided by the tour operator to its API clients. This is functional and the licence key is checked for the client's validity on the tour operator's system.
TS_API_VERSION	This will be provided by Open Destinations and will indicate the version number of Travel Studio being used. Not currently functional.

The XML file will be created using the Unicode-UTF-8 (for APIs) and Unicode-UTF-16 (for Data Export) standard to allow extended character support. Examples of these extended characters are: à, â, ê, etc.

XML Format: Each XML file would have this as the header.

```
<VERSION_HISTORY APPLICATION_NAME="AppName " XML_FILE_NAME="XMLFileName" LICENCE_KEY="LicKey"
TS_API_VERSION="3.5.8">
  <XML_VERSION_NO>3.0</XML_VERSION_NO>
</VERSION_HISTORY>
```

2 XML API INTERFACE

2.1 *Communicating with the Travel Studio API*

The Travel Studio interface will accept data via an ASP page. This will ensure consistency of the existing Travel Studio architecture and will allow Tour Operators to maintain their existing web infrastructure. The existing Travel Studio architecture uses the Microsoft XML-HTTP-POST protocol. The API-Client will need to use the XML HTTP POST protocol to post XML to asp pages. The API-Client will post the XML file on to the ASP page hosted by the Tour Operator. The Tour Operator will provide the exact URL/IP address. The Tour Operator will maintain both a Test and Live environment. The Test environment will contain valid data but will not affect any live data held by the Tour Operator. The result of any XML call made by the API-Client will be sent back to the API-Client in the response object of that ASP page.

An example of the code that the API-Client would need to implement is given below in JavaScript. If the API-Client needs advice on using these calls from within their existing programs this can be provided.

```
var g_XMLReq = new ActiveXObject("Microsoft.XMLHTTP");
g_XMLReq.open("POST", "http://aggregator/tsweb2.0/sd/testXML.asp",
false);
g_XMLReq.send( unescape(XMLdata) );
/* the result of the script that is resulting XML will be returned in */
g_XMLReq.responseText
/* which can be parsed to extract required data. */
```

2.2 *API Supported Messages*

The API will support the following API calls

XML file names	Utility	DTD file name	Contents
GeoTreeRequest	ExportGeoTree.asp(post request using XML-HTTP-POST)	GeoTreeRequest.dtd	Details request
GeoTreeResponse	GeoTreeResponse.asp: Response object		Geotree details
FacilitiesRequest	ExportFacility.asp(post request using XML-HTTP-POST)	FacilityRequest.dtd	Facilities Request
FacilityResponse	FacilityResponse.asp:Response object	Facilities.dtd	Facility Details

HotelListRequest	ExportHotelList.asp (post request using XML-HTTP-POST)	HotelListRequest.dtd	Search criteria for hotel list search
HotelListResponse	HotelListResponse.asp: Response object	HotelList.dtd	Search results giving list of hotels
HotelDetailsRequest	ExportHotelDetail.asp (post request using XML-HTTP-POST)	HotelDetailsRequest.dtd	Hotel Details request
HotelDetailsResponse	HotelDetailsResponse.asp: Response object	HotelDetailsResponse.dtd	Search result giving hotel details
SearchServiceRequest	ServiceSearch.asp (post request using XML-HTTP-POST)	Service_Search_Request.dtd	Search criteria for Hotels search along with availability and price search.
SearchServiceResponse	ServiceSearch.asp Response object	Service_Search_Response.dtd	Search results giving details of services and prices and availability
AvailabilityAndPricingSearchRequest	AvailabilityAndPrices.asp (post request using XML-HTTP-POST)	AvailabilityAndPricesSearch.dtd	Search criteria for availability and price search
AvailabilityAndPricingSearchResponse	AvailabilityAndPricesResponse object	AvailabilityAndPricesResponse.dtd	Result of availability and price search
Booking	Booking.asp (Post request using XML-HTTP-POST)	Booking.dtd	Data required to create a Booking in Travel Studio
BookingConfirmation	Booking.asp Response object	BookingConfirmation.dtd	Result of Booking request – will return a Booking reference number and other confirmation details
BookingDetailsRequest	BookingInfoRequest.asp (post request using XML-HTTPPOST)	BookingDetailsRequest.dtd	Request to fetch details about booking
BookingDetailsResponse	BookingInfoRequest.asp Response object	BookingDetailsResponse.dtd	Booking Details
CancellationPolicyDetailsRequest	CancellationPolicyDetails.asp (Post request using XML-HTTP-POST)	CancellationPolicyDetailsRequest.dtd	Data required to return cancellation policies
CancellationPolicyDetailsResponse	Booking.asp Response object	CancellationPolicyDetailsConfirmation.dtd	Cancellation Policy results
BookingCancellation	CancelBooking.asp (Post request using XML-HTTP-POST)	BookingCancellation.dtd	Request to cancel an existing Booking

CancellationResponse	CancelBooking.asp Response object	BookingCancellationCo nfi rmation.dtd	Acknowledgement of a Cancellation
Error	Response object for all APIs if an error occurs.	Error.dtd	Results of Error generated by an incorrect request.

3 API Static Data Requests

3.1 The GeoTree API request

Geo Tree Request utility also allows Tour Operators to upload their geo tree details from the Travel Studio system. The file will be created as Geo_Tree_Response.

```
<GEO_TREE_REQUEST>
  <VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="XMLFileName" LICENCE_KEY="LicKey"
    TS_API_VERSION="3.5.6">
    <XML_VERSION_NO>3.0</XML_VERSION_NO>
  </VERSION_HISTORY>
</GEO_TREE_REQUEST>
```

3.2 The Geo_Tree_Response

Any Geo locational search would return all hotels in the specified Geo Location node and all child nodes under that node.

```
<GEO_TREE_RESPONSE>
  <VERSION_HISTORY APPLICATION_NAME="TS HOTELS API 1.0" XML_FILE_NAME="Response.XML"
    LICENCE_KEY="LicKey" TS_API_VERSION="v309.0.0">
    <XML_VERSION_NO>1.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <CONTINENT NAME="">
    <COUNTRY NAME="">
      <REGION NAME="">
        <AREA NAME="">
          <CITY NAME="">
            <CITYCENTRESUBURBS NAME="">
              <DISTRICT NAME="">
                <STATION NAME="" />
              </DISTRICT>
            </CITYCENTRESUBURBS>
          </CITY>
        </AREA>
      </REGION>
    </COUNTRY>
  </CONTINENT>
</GEO_TREE_RESPONSE>
```

3.3 *The Facilities API request*

A separate API is given to list all the facilities in the Travel Studio system

```
<FACILITIES_REQUEST>
  <VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="XMLFileName" LICENCE_KEY="LicKey"
    TS_API_VERSION="3.5.6">
    <XML_VERSION_NO>3.0</XML_VERSION_NO>
  </VERSION_HISTORY>
</FACILITIES_REQUEST>
```

3.4 *The Facilities API response*

Facilities response will return in all available languages.

```
<FACILITY_RESPONSE>
  <VERSION_HISTORY APPLICATION_NAME="TS HOTELS API 1.0" XML_FILE_NAME="Response.XML"
    LICENCE_KEY="LicKey" TS_API_VERSION="v309.0.0">
    <XML_VERSION_NO>1.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <FACILITY ID="">
    <NAME LANGNAME="">
      <DESCRIPTION />
    </NAME>
    <NAME LANGNAME="">
      <DESCRIPTION />
    </NAME>
  </FACILITY>
</FACILITY_RESPONSE>
```

3.5 *HotelList API request*

Functionality of API provides clients with a convenient method to receive a list of recently modified services.

```
<HOTEL_LIST_REQUEST>
  <VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="XMLFileName" LICENCE_KEY="LicKey"
    TS_API_VERSION="3.5.6">
    <XML_VERSION_NO>3.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <BODY>
    <STARTDATE/>
    <ENDDATE />
  </BODY>
</HOTEL_LIST_REQUEST>
```

The License key would be validated for every HotelList request. If an invalid or blank license key is passed, the following error would be returned: "The license key is invalid. Please supply a valid license key."

3.5.1 <STARTDATE>

Type – tag
Optional – No

Description - if supplied, the HotelList response would contain only those hotel services which have been modified between the start and end date supplied. If terminated nodes are supplied for the <StartDate> and <EndDate> tags, all services would be returned.

3.5.2 <ENDDATE>

Type – tag
Optional – No

Description - if supplied, the HotelList response would contain only those hotel services which have been modified between the start and end date supplied. If terminated nodes are supplied for the <StartDate> and <EndDate> tags, all services would be returned.

3.6 *HotelList API response*

Response returns a list of serviceid's and last updated date.

```
<HOTEL_LIST_RESPONSE>
  <VERSION_HISTORY APPLICATION_NAME="" XML_FILE_NAME="" LICENCE_KEY="" TS_API_VERSION="">
  <XML_VERSION_NO />
</VERSION_HISTORY>
  <HOTEL ID="" NAME="" LASTUPDATEDATE="yyyy-mm-dd HH:MM:SS" ISDELETED="" />
  <HOTEL ID="" NAME="" LASTUPDATEDATE="yyyy-mm-dd HH:MM:SS" ISDELETED="" />
  <HOTEL ID="" NAME="" LASTUPDATEDATE="yyyy-mm-dd HH:MM:SS" ISDELETED="" />
</HOTEL_LIST_RESPONSE>
```

3.6.1 <HOTEL>

Type – tag
Optional – no

Description – This element would contain the following attributes:

- **ID attribute:** represents the ServiceID
- **Name:** Service name
- **LastUpdatedDate:** When the service was last updated.
- **IsDeleted:** If a service has deleted status, the IsDeleted flag will be 1. In case of Internet status, the flag will be 0.

3.7 Hotel Details API Request

Response returns a list of serviceid's and last updated date

```
<HOTEL_DETAILS_REQUEST>
  <VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="XMLFileName" LICENCE_KEY="LicKey"
    TS_API_VERSION="3.5.6">
    <XML_VERSION_NO>3.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <BODY>
    <SERVICEID />
  </BODY>
</HOTEL_DETAILS_REQUEST>
```

3.7.1 <SERVICEID>

Type – tag

Optional – no

Description –Serviceid to be returned in HotelDetails response.

3.8 Hotel Details API Response

```
<HOTEL_DETAILS_RESPONSE>
  <VERSION_HISTORY APPLICATION_NAME="TS HOTELS API 1.0" XML_FILE_NAME="Response.XML"
    LICENCE_KEY="LicKey" TS_API_VERSION="v309.0.0">
    <XML_VERSION_NO>1.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <HOTELDETAILS>
    <SERVICEID />
    <LONGNAME />
    <DESCRIPTIONS>
      <DESCRIPTION LANGUAGE="English" />
      <DESCRIPTION LANGUAGE="French" />
      <DESCRIPTION LANGUAGE="German" />
      <DESCRIPTION LANGUAGE="Spanish" />
    </DESCRIPTIONS>
    <TERMS_INCLUSION />
    <TERMS_EXCLUSION />
    <PRICE_START_RANGE />
    <PRICE_END_RANGE />
    <CURRENCY_CODE />
    <FACILITIES />
    <STAR_RATING />
    <LOCATION />
    <IMAGES>
      <CATEGORY>
        <CATEGORYNAME />
        <IMAGE />
        <THUMBNAIL />
      </CATEGORY>
    </IMAGES>
  </HOTELDETAILS>
</HOTEL_DETAILS_RESPONSE>
```



```

</IMAGES>
<ADDRESS>
  <LINE1 />
  <LINE2 />
  <LINE3 />
  <LINE4 />
  <POSTCODE />
  <CITY />
  <COUNTRY />
  <TELEPHONENUMBER />
  <FAXNUMBER />
  <EMERGENCYPHONENUMBER />
  <EMAILADDRESS />
  <WEBSITEURL />
  <STATE />
</ADDRESS>
<SERVICE_ADDINFO>
  <PRIMARYAIRPORT />
  <YEAROFOOPENING />
  <NEARESTAIRPORT />
  <SERVICELONGITUDE />
  <SERVICELATITUDE />
  <SERVICECHECKINTIME />
  <SERVICECHECKOUTTIME />
</SERVICE_ADDINFO>
<SERVICE_CREDITCARDS_ACCEPTED>
  <CREDITCARD_ACCEPTED />
</SERVICE_CREDITCARDS_ACCEPTED>
<RATINGS>
  <RATINGTYPE RATINGTYPENAME="">
    <RATING />
  </RATINGTYPE>
</RATINGS>
<OPTIONS>
  <OPTION>
    <OPTION_ID />
    <OCCUPANCY_TYPE_ID />
    <OCCUPANCY_TYPE_NAME />
    <OPTION_NAME LANGNAME="French" />
    <OPTION_NAME LANGNAME="Spanish" />
    <OPTION_NAME LANGNAME="German" />
    <CHILD_POLICY_NAME LANGNAME="English" />
    <CHILD_POLICY_NAME LANGNAME="French" />
    <CHILD_POLICY_NAME LANGNAME="German" />
    <CHILD_POLICY_NAME LANGNAME="Spanish" />
    <CHILD_RATES>
      <RATE AGESTART="" AGEEND="" RATEVALUE="" />
    </CHILD_RATES>
  </OPTION>
</OPTIONS>
</HOTELDETAILS>
</HOTEL_DETAILS_RESPONSE>

```

3.9 ***Service Images***

Image URLs for main image and thumbnail versions are returned from the ExportHotelDetails API response.

Open Travel Alliance (OTA) Code refers to the various picture categories:

OTA Code	Picture category
1	Exterior view
2	Lobby view
3	Pool view
4	Restaurant
5	Health club
6	Guest room
7	Suite
8	Meeting room
9	Ballroom
10	Golf course
11	Beach
12	Spa
13	Bar/Lounge
14	Recreational facility
15	Logo
16	Basics
17	Map
18	Promotional
19	Hot news
20	Miscellaneous
21	Guest room amenity
22	Property amenity
23	Business center

3.9.1 **Image name format**

- <Hotel code>: Service id
- <Category Number>: Incremental number generated by the system, consecutive and unique within a repeating OTA category image
- <OTA code>: OTA picture category code

4 API Service Search and Availability Requests

4.1 Services Search API on given criteria

This XML request message will allow the API-Client to view the availability and price information for a specific Geo Location or Service ID(s).

```
<SERVICE_SEARCH_REQUEST>
  <VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="XMLFileName" LICENCE_KEY="LicKey"
    TS_API_VERSION="3.5.6">
    <XML_VERSION_NO>3.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <GEO_LOCATION_NAME />
  <SERVICEIDs />
  <START_DATE />
  <NUMBER_OF_NIGHTS />
  <AVAILABLE_ONLY />
  <GET_START_PRICE />
  <ROOM_REPLY>
    <ALL_ROOM />
    <ANY_ROOM />
  </ROOM_REPLY>
  <PRICING>
    <MULTIPLY_QTY />
  </PRICING>
  <ROOMS_REQUIRED>
    <ROOM>
      <OCCUPANCY />
      <QUANTITY />
      <CHILDREN>
        <CHILD_RATE CHILD_QUANTITY="" CHILD_AGE="" />
      </CHILDREN>
    </ROOM>
  </ROOMS_REQUIRED>
</SERVICE_SEARCH_REQUEST>
```

4.1.1 <GEO_LOCATION_NAME>

Type – tag

Optional – yes

Description - Geo tree location name that will be used to filter services for the location. This name will be used for an exact match. If this tag is not provided then the SERVICEIDs tag must be used

4.1.2 <SERVICEIDs>

Type – tag

Optional – yes

Description – Comma separated service ID's which will be used in search criteria. Format: x1, x2, x3.... This tag must only be used when multiple services are being searched. For a single service ID please use the "Availability and Pricing Search". This tag is combined with the GEO_LOCATION_NAME and HOTEL_CATEGORY values. This tag can be used as stand-alone

but when combined with location and category tags, the location and category must match the service id criteria.

4.1.3 <HOTEL_CATEGORY/>

Type – tag
Optional – yes

Description - Star rating of the hotel. If this is provided then only hotels with this star rating would be returned.

Star Ratings:

1 *
2 *
3 *
4 *
5 *
2 * +
3 * +
4 * +

- “+” means 2 star deluxe which is one category higher than the standard 2 * hotel
- Each hotel is rated by the countries own international standards rating authority. What ever the ratings system of that country is; JacTravel ensures that the hotel is accredited by that organization.

4.1.4 <START_DATE>

Type – tag
Optional – no

Description - Start date for Service search. This should be supplied in dd mmm yyyy format

4.1.5 <NUMBER_OF_NIGHTS>

Type – tag
Optional – no

Description - End number of nights for service search. This will determine the end date of search. Please note that the values given in search are not intuitive and require the arrival date to be treated as 0:

eg: 0 = 1 night stay
1 = 2 nights stay
2 = 3 night stay
Etc

4.1.6 <AVAILABLE_ONLY>

Type – tag
Optional – no

Description - Boolean flag specifying whether only 'available' services will be returned or if 'on-request' services will also returned. If not specified it will default to false and all services will be returned.

4.1.7 <GET_START_PRICE>

Type – tag
Optional – no

Description - Boolean flag specifying whether the API result must contain prices for each service.

4.1.8 <ROOM_REPLY>

Type – tag
Optional – yes

Description – Tag structure that should contain one of the following Empty Tags.

1. ALL_ROOM: All options of occupancy type defined in Room_Required (Default implementation). Service should support all required options.
2. ANY_ROOM: Any options defined in Room_Required (Additional Implementation). Service should support at least one required option.

4.1.9 <PRICING>

Type – tag
Optional – yes

Description – Tag structure that will contain any/both of the following attributes.

1. MULTIPLY_QTY: This will return the pricing by multiplying the number with qty with unit price.

4.1.10 <ROOMS_REQUIRED>

Type – tag
Optional – no

Description – Will contain rooming requirements in the form of room type and the quantity required for each room.

4.1.11 <ROOM>

Type – tag
Optional – no

Description – Room details structure, which can be multiple to allow search for multiple room types.

4.1.12 <OCCUPANCY>

Type – tag
Optional – no

Description – Room Occupancy ID. (see appendices). Family room types require the number of children and all their ages to retrieve prices.

4.1.13 <QUANTITY>

Type – tag
Optional – no

Description – Number of rooms required

4.1.14 <CHILDREN>

Type – tag
Optional – yes

Description – Tag that holds children information

4.1.15 <CHILD_RATE>

Type – tag
Optional – yes
Description – No of children - Required when requesting family rooms

4.1.16 <AGES_OF_CHILDREN>

Type – tag
Optional – yes

Description – comma separated list of all child ages

4.1.17 <ROOMS_REQUIRED>

Type – tag
Optional – yes

Description – Will contain rooming requirements in the form of room type and the quantity required for each room.

4.1.18 <ROOM>

Type – tag
Optional – no

Description – Room details structure, which can be multiple to allow search for multiple room types.

4.1.19 <OCCUPANCY>

Type – tag
Optional – yes

Description – Room Occupancy ID. (see appendices). Family room types require the number of children and all their ages to retrieve prices.

4.1.20 <QUANTITY>

Type – tag
Optional – no

Description – Number of rooms required

4.1.21 <CHILDREN>

Type – tag
Optional – yes

Description –No of children - Required when requesting family rooms

4.1.22 <CHILD_RATE>

Type – tag

Optional – yes

Description – Tag that hold information of child at age level. If child prices are required for multiple children with different ages then this tag will be repeated for each age.

Attributes

CHILD_QUANTITY – Number of children

CHILD_AGE – Age of children

4.2 Service Search Response

Output Result

```
<SERVICE_SEARCH_RESPONSE>
  <VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="FileName" LICENCE_KEY="LicKey"
TS_API_VERSION="v3.0">
    <XML_VERSION_NO>1.0</XML_VERSION_NO>
    </VERSION_HISTORY>
    <SERVICES>
      <SERVICE SERVICE_ID="" SERVICE_NAME="" LOCATION="" RATING="" AVAILABLE=""
ISRECOMMENDEDPRODUCT="" STARTING_PRICE="" CURRENCY="">
        <OPTIONS>
          <OPTION>
            <OPTIONID />
            <OPTION_NAME />
            <MinAdult />
            <MaxAdult />
            <MinChild />
            <MaxChild />
            <ChildMaxAge />
            <OCCUPANCY />
            <PRICES>
              <PRICE>
                <PRICE_DATE />
                <SELL_PRICE_ID />
                <SELL_PRICE_AMOUNT />
                <SELL_CURRENCY_CODE />
                <MEAL_PLAN>
                  <MEAL_PLAN_TEXT />
                  <MEAL_PLAN_TYPE>
                    <INCLUDESBREAKFAST />
                    <INCLUDESLUNCH />
                    <INCLUDESDINNER />
                  </MEAL_PLAN_TYPE>
                </MEAL_PLAN>
                <CHILD_PRICES>
                  <CHILD_PRICE>
                    <AGE />
                    <SELL_PRICE_AMOUNT />
                  </CHILD_PRICE>
                </CHILD_PRICES>
                <ORIGINAL_SELL_PRICE_AMOUNT />
              </PRICE>
            </PRICES>
            <OPTION_STATUS />
            <RULE_TEXT />
          </OPTION>
        </OPTIONS>
      </SERVICE>
    </SERVICES>
  </SERVICE_SEARCH_RESPONSE>
```


4.2.1 <SERVICES>

Type: Tag

Description – Tag structure to hold services

4.2.2 <SERVICE>

Type: tag

Description – Rating of service

Attributes:

SERVICEID – Unique ID of the service that must be used to invoke the booking API.

SERVICE_NAME – Long name of service

LOCATION – Geo Tree location assigned to the service

RATING – Rating of service

AVAILABLE – legacy tag, clients should ignore

ISRECOMMENDEDPRODUCT - – legacy tag, clients should ignore

STARTING_PRICE -Starting price of the service. This will be the minimum price of the service as per the rooming requirements. This will be per room price per day

CURRENCY: ISO code of currency

4.2.3 <OPTIONS>

Type: tag

Description – Tag that holds multiple room details.

4.2.4 <OPTION>

Type – tag

Description – Tag that holds room details.

4.2.5 <OPTIONID>

Type: Integer

Description – The ID in the Travel Studio database that uniquely identifies a hotel room.

4.2.6 <OPTION_NAME>

Type: Integer

Description –Option name. For example, “Twin Room”, “Single Room”, “Double for Single Occupancy”

4.2.7 <MinAdult>

Type: Integer

Description – Min adult supported by option

- 4.2.8 <MaxAdult>**
Type: Integer
Description – Max adult supported by option
- 4.2.9 <MinChild>**
Type: Integer
Description – Min child supported by option
- 4.2.10 <MaxChild>**
Type: Integer
Description – Max child supported by option
- 4.2.11 <ChildMaxAge>**
Type: Integer
Description – Max child age
- 4.2.12 <OCCUPANCY>**
Type: Integer
Description – Room Occupancy ID
- 4.2.13 <PRICES>**
Type: Integer
Description – Repeating information for price on each day
- 4.2.14 <PRICE>**
Type: tag
Description – Tag contains price information
- 4.2.15 <PRICE_DATE>**
Type: date
Description – Date of price. This will be supplied in *dd mmm yyyy* format
- 4.2.16 <SELL_PRICE_ID>**
Type: Integer
Description – The ID that identifies the price record on the TS database. This will be used by the API-Client when booking a room type on Travel Studio.
- 4.2.17 <SELL_PRICE_AMOUNT>**
Type: Integer
Description – Price of the option on a specific date. Prices are Net rates.

- 4.2.18 <SELL_CURRENCY_CODE>**
Type: String
Description – ISO Currency code
- 4.2.19 <MEAL_PLAN>**
Type: Tag
Description – Tag structure containing meal plan information for a single day
- 4.2.20 <MEAL_PLAN_TEXT>**
Type: String
Description – Meal plan description. For example “Room Only”, “Full English Breakfast”.
- 4.2.21 <INCLUDESBREAKFAST>**
Type: Integer
Description – Value of 1 indicates breakfast provided, 0 indicates the opposite
- 4.2.22 <INCLUDESLUNCH>**
Type: Integer
Description – Value of 1 indicates lunch provided, 0 indicates the opposite
- 4.2.23 <INCLUDESDINNER>**
Type: Integer
Description – Value of 1 indicates dinner provided, 0 indicates the opposite
- 4.2.24 <CHILD_PRICES>**
Type: Tag structure
Description – Structure to return child prices
- 4.2.25 <CHILD_PRICE>**
Type: Tag Structure
Description – Structure that contains child price per age.
- 4.2.26 <AGE>**
Type: Integer
Description – Age of child for which price is returned.
- 4.2.27 <SELL_PRICE_AMOUNT>**
Type: Integer
Description – Price of the child calculated according to child policies within Travel Studio. Prices are Net rates.

4.2.28 <ORIGINAL_SELL_PRICE_AMOUNT>

Type: Integer

Description – Legacy tag, client should ignore this value

4.2.29 <OPTION_STATUS>

Type: String

Description – Shows availability of option. Either “Available” or “On Request”

4.2.30 <RULE_TEXT>

Type: String

Description – Displays any special rules applied in the Travel Studio system. For example “Stay 3 nights, pay for 2”.

4.3 Availability and Pricing Search

AvailabilityAndPricingSearchRequest.XML

This XML request message will allow the API-Client to view the availability and price information for a specific hotel.

```
<HOTEL_AVAILABILITY_AND_PRICE_SEARCH_CRITERIA>
  <VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="XMLFileName" LICENCE_KEY="LicKey"
    TS_API_VERSION="3.5.6">
    <XML_VERSION_NO>3.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <SERVICE_ID AVAILABLE_ONLY="" />
  <CLIENT_NAME />
  <BOOKING_START_DATE />
  <BOOKING_END_DATE />
  <ROOM_REPLY>
    <ALL_ROOM />
    <ANY_ROOM />
  </ROOM_REPLY>
  <PRICING>
    <MULTIPLY_QTY />
    <VALIDATE_QTY />
  </PRICING>
  <ROOMS_REQUIRED>
    <ROOM>
      <OCCUPANCY />
      <QUANTITY />
      <NO_OF_PASSENGERS />
      <NO_OF_CHILDREN />
      <AGES_OF_CHILDREN />
    </ROOM>
  </ROOMS_REQUIRED>
</HOTEL_AVAILABILITY_AND_PRICE_SEARCH_CRITERIA>
```

4.3.1 <SERVICE_ID>

Type – tag

Optional – no

Attribute – Available_Only - This is an optional attribute at service level, which allows API client to search for 'available' and 'onrequest' services. True by default.

Description – The ID on Travel Studio database that uniquely identifies a hotel

4.3.2 <CLIENT_NAME>

Type – tag

Optional – yes

Description - Name of API Client

4.3.3 <BOOKING_START_DATE>

Type – tag

Optional – no

Description - Start date for both availability and prices. This should be supplied in dd mmm yyyy format

4.3.4 <BOOKING_END_DATE>

Type – tag
Optional – no

Description - End date for both availability and prices. This should be supplied in dd mmm yyyy format and is representative of the last night of stay. Date of departure is not specified in this request.

Eg 1 night stay =

```
<BOOKING_START_DATE>01 Nov 11</BOOKING_START_DATE>  
<BOOKING_END_DATE>01 Nov 11</BOOKING_END_DATE>
```

2 night stay =

```
<BOOKING_START_DATE>01 Nov 11</BOOKING_START_DATE>  
<BOOKING_END_DATE>02 Nov 11</BOOKING_END_DATE>
```

4.3.5 <ROOM_REPLY>

Type – tag
Optional – yes

Description – Tag structure that should contain one of the following Empty Tags.

1. ALL_ROOM: All options of occupancy type defined in Room_Required (Default implementation). Service should support all required options.
2. ANY_ROOM: Any options defined in Room_Required (Additional Implementation). Service should support at least one required option.

4.3.6 <PRICING>

Type – tag
Optional – yes

Description – Tag structure that will contain any/both of the following attributes.

1. MULTIPLY_QTY: This will return the pricing by multiplying the number with qty with unit price.
2. VALIDATE_QTY: If present, the Adult qty will be checked against the capacity and error will be returned in case of mismatch.

4.3.7 <ROOMS_REQUIRED>

Type – tag
Optional – yes

Description – Will contain rooming requirements in the form of room type and the quantity required for each room.

4.3.8 <ROOM>

Type – tag
Optional – no

Description – Room details structure, which can be multiple to allow search for multiple room types.

4.3.9 <OCCUPANCY>

Type – tag
Optional – no

Description – Room Occupancy ID. (See appendices). Family room types require the number of children and all their ages to retrieve prices.

4.3.10 <QUANTITY>

Type – tag
Optional – no

Description – Number of rooms required

4.3.11 <NO_OF_PASSENGERS>

Type: Integer
Optional: Yes

Description: Required when specifying the VALIDATE_QTY tag and represents the total amount of adult passengers.

4.3.12 <NO_OF_CHILDREN>

Type – tag
Optional – yes

Description –No of children - Required when requesting family rooms and represents the total amount of children passengers.

4.3.13 <AGES_OF_CHILDREN>

Type – tag
Optional – yes

Description – comma separated list of all child ages

4.4 *Availability and Pricing Search Response*

Output Result:

```
</HOTEL_AVAILABILITY_AND_PRICE_SEARCH_RESULT>
  <VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="FileName" LICENCE_KEY="LicKey"
    TS_API_VERSION="v3.0">
    <XML_VERSION_NO>1.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <SERVICE_ID />
  <RATING />
  <AVAILABILITY />
  <OPTIONS>
    <OPTION>
      <OPTIONID />
      <OPTION_NAME />
      <MinAdult />
      <MaxAdult />
      <MinChild />
      <MaxChild />
      <ChildMaxAge />
      <OCCUPANCY />
      <PRICES>
        <PRICE>
          <PRICE_DATE />
          <SELL_PRICE_ID />
          <SELL_PRICE_AMOUNT />
          <SELL_CURRENCY_CODE />
          <MEAL_PLAN>
            <MEAL_PLAN_TEXT />
            <MEAL_PLAN_TYPE>
              <INCLUDESBREAKFAST />
              <INCLUDESLUNCH />
              <INCLUDESDINNER />
            </MEAL_PLAN_TYPE>
          </MEAL_PLAN>
          <CHILD_PRICES>
            <CHILD_PRICE>
              <AGE />
              <SELL_PRICE_AMOUNT />
            </CHILD_PRICE>
          </CHILD_PRICES>
          <ORIGINAL_SELL_PRICE_AMOUNT />
        </PRICE>
      </PRICES>
      <OPTION_STATUS />
      <RULE_TEXT />
    </OPTION>
  </OPTIONS>
</HOTEL_AVAILABILITY_AND_PRICE_SEARCH_RESULT>
```


- 4.4.1 <SERVICE_ID>**
Type: Integer
Description – ID on Travel Studio database that uniquely identifies a hotel.
- 4.4.2 <RATING>**
Type: String
Description – Rating of service
- 4.4.3 <AVAILABILITY>**
Type: String
Description – Availability status. Whether service is available or not. We will return either “YES” or “NO”.
- 4.4.4 <OPTIONS>**
Type: tag
Description – Tag that holds multiple room details.
- 4.4.5 <OPTIONID>**
Type: Integer
Description – The ID in the Travel Studio database that uniquely identifies a hotel room.
- 4.4.6 <OPTION_NAME>**
Type: Integer
Description – Option name. For example, “Twin Room”, “Single Room”, “Double for Single Occupancy”
- 4.4.7 <MinAdult>**
Type: Integer
Description – Min adult supported by option
- 4.4.8 <MaxAdult>**
Type: Integer
Description – Max adult supported by option
- 4.4.9 <MinChild>**
Type: Integer
Description – Min child supported by option
- 4.4.10 <MaxChild>**
Type: Integer

Description – Max child supported by option

4.4.11 <ChildMaxAge>

Type: Integer

Description – Max child age

4.4.12 <OCCUPANCY>

Type: Integer

Description – Room Occupancy ID

4.4.13 <PRICES>

Type: Integer

Description – Repeating information for price on each day

4.4.14 <PRICE>

Type: tag

Description – Tag contains price information

4.4.15 <PRICE_DATE>

Type: date

Description – Date of price. This will be supplied in *dd mmm yyyy* format

4.4.16 <SELL_PRICE_ID>

Type: Integer

Description – The ID that identifies the price record on the TS database. This will be used by the API-Client when booking a room type on Travel Studio.

4.4.17 <SELL_PRICE_AMOUNT>

Type: Integer

Description – Price of the option on a specific date. Prices are Net rates.

4.4.18 <SELL_CURRENCY_CODE>

Type: String

Description – ISO Currency code

4.4.19 <MEAL_PLAN>

Type: Tag

Description – Tag structure containing meal plan information for a single day

4.4.20 <MEAL_PLAN_TEXT>

Type: String

Description – Meal plan description. For example “Room Only” , “Full English Breakfast”.

4.4.21 <INCLUDESBREAKFAST>

Type: Integer

Description – Value of 1 indicates breakfast provided, 0 indicates the opposite

4.4.22 <INCLUDESLUNCH>

Type: Integer

Description – Value of 1 indicates lunch provided, 0 indicates the opposite

4.4.23 <INCLUDESDINNER>

Type: Integer

Description – Value of 1 indicates dinner provided, 0 indicates the opposite

4.4.24 <CHILD_PRICES>

Type: Tag structure

Description – Structure to return child prices

4.4.25 <CHILD_PRICE>

Type: Tag Structure

Description – Structure that contains child price per age.

4.4.26 <AGE>

Type: Integer

Description – Age of child for which price is returned.

4.4.27 <SELL_PRICE_AMOUNT>

Type: Integer

Description – Price of the child calculated according to child policies within Travel Studio. Prices are Net rates.

4.4.28 <ORIGINAL_SELL_PRICE_AMOUNT>

Type: Integer

Description – Legacy tag, client should ignore this value

4.4.29 <OPTION_STATUS>

Type: String

Description – Shows availability of option. Either “Available” or “On Request”

4.4.30 <RULE_TEXT>

Type: String

Description – Displays any special rules applied in the Travel Studio system. For example “Stay 3 nights, pay for 2”.

5 API Booking Request

5.1 Booking Request Structure for Hotel API

```
<BOOKING_DETAILS>
  <VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="XMLFileName" LICENCE_KEY="LicKey"
    TS_API_VERSION="3.5.6">
    <XML_VERSION_NO>3.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <BOOKING>
    <CLIENT_NAME />
    <BOOKING_NAME />
    <BOOKING_DATE />
    <BOOKING_START_DATE />
    <BOOKING_END_DATE />
    <NUMBER_OF_NIGHTS />
    <CLIENT_REFERENCE />
    <VALIDATE_QTY />
    <SERVICE_ID AVAILABLE_ONLY="" />
    <TOTAL_ADULTS />
    <TOTAL_CHILDREN />
    <RETURN_BOOKING_DETAILS />
    <PAX_OCCUPANCYS>
      <PAX_OCCUPANCY TYPE="">
        <OCCUPANCYID />
        <NO_OF_PAX />
      </PAX_OCCUPANCY>
    </PAX_OCCUPANCYS>
    <PASSENGERS>
      <PASSENGER TYPE="">
        <FIRST_NAME />
        <LAST_NAME />
        <AGE />
        <TITLE />
      </PASSENGER>
    </PASSENGERS>
    <OPTIONS>
      <OPTION>
        <OPTION_DATE />
        <OPTION_ID />
        <QUANTITY />
        <NO_OF_ADULTS />
        <NO_OF_CHILDREN />
        <SELL_PRICE_ID />
        <CHILDREN>
          <AGES>
            <AGE />
            <COUNT />
          </AGES>
        </CHILDREN>
        <AVAILABLE_ONLY />
      </OPTION>
    </OPTIONS>
    <NOTES>
      <NOTE />
    </NOTES>
  </BOOKING>
</BOOKING_DETAILS>
```

- 5.1.1 <BOOKING>**
Type: tag
Optional - no

Description – Structure tag that contains booking information
- 5.1.2 <CLIENT_NAME>**
Type: String
Optional - yes

Description – API Client name
- 5.1.3 <BOOKING_NAME>**
Type: String (150 characters allowed)
Optional - no

Description – Name of Booking – Unique booking name is strongly recommended.
- 5.1.4 <BOOKING_DATE>**
Type: date
Optional - no

Description – Date on which booking is made. Supplied in dd mmm yyyy
- 5.1.5 <BOOKING_START_DATE>**
Type: date
Optional - no

Description – Booking Start Date. Supplied in dd mmm yyyy
- 5.1.6 <BOOKING_END_DATE>**
Type: date
Optional - no

Description – Booking End Date, date of departure inclusive. Supplied in dd mmm yyyy
- 5.1.7 <NUMBER_OF_NIGHTS>**
Type: integer
Optional - no

Description – No of nights which is the intuitive value, eg 1 = 1 night, 2 = 2 nights, 3 = etc
- 5.1.8 <CLIENT_REFERENCE>**
Type: String (50 characters allowed)
Optional – yes

Description – Client reference –Using the same unique value as <BOOKING_NAME> is strongly recommended

5.1.9 <VALIDATE_QTY>

Type: tag
Optional - yes

Description – If present, the Adult qty will be checked against the capacity and error will be returned in case of mismatch.

5.1.10 <SERVICE_ID>

Type: Integer
Optional – no

Attribute – Available_Only - This is an optional attribute at service level, which allows the API client to book 'available' and 'on-request' services. The default if this flag is not provided is TRUE that is to book only 'available' services. If the API client wants to book 'on-request' services also, then this attribute should be set to FALSE.

Description – ID within the Travel Studio database, which uniquely identifies a hotel.

5.1.11 <TOTAL_ADULTS>

Type: Integer
Optional - no

Description – Total Adults in booking

5.1.12 <TOTAL_CHILDREN>

Type: Integer
Optional - no

Description – Total children in booking

5.1.13 <RETURN_BOOKING_DETAILS>

Type: tag
Optional - yes

Description – Returns booking details in booking response (recommended)

5.1.14 <PAX_OCCUPANCYS>

Type: tag
Optional - no

Description – Structure to hold multiple PAX_OCCUPANCY.

5.1.15 <PAX_OCCUPANCY>

Type: tag

Optional – no

Attribute – TYPE. The Type attribute represents either an Adult or Child.

1=Adult

2=CHILD

Description – Structure to hold passenger types, grouped by occupancyID. Each PAX_OCCUPANCY displays the no. of passengers per occupancy ID. For example if there are 5 adult passengers + 2 children staying in 2 x Double/Twin + 1 child rooms and 1 Single room:

```
<PAX_OCCUPANCYS>
  <PAX_OCCUPANCY TYPE="1"> Adults
    <OCCUPANCYID>7</OCCUPANCYID> Double/Twin + 1 child room
    <NO_OF_PAX>4</NO_OF_PAX> 4 adults staying
  </PAX_OCCUPANCY>
  <PAX_OCCUPANCY TYPE="2"> Children
    <OCCUPANCYID>7</OCCUPANCYID> Double/Twin + 1 child room
    <NO_OF_PAX>2</NO_OF_PAX> 2 children staying
  </PAX_OCCUPANCY>
  <PAX_OCCUPANCY TYPE="1"> Adults
    <OCCUPANCYID>1</OCCUPANCYID> Single room
    <NO_OF_PAX>1</NO_OF_PAX> 1 adult staying
  </PAX_OCCUPANCY>
</PAX_OCCUPANCYS>
```

5.1.16 <OCCUPANCYID>

Type: Integer

Optional - no

Description – Type of room -Family room types require the number of children and all their ages.

5.1.17 <NO_OF_PAX>

Type: Integer

Optional - no

Description – No of passengers staying in the same type of room

5.1.18 <PASSENGERS>

Type: tag

Optional - no

Description – Tag that will hold passenger information, such as name, age etc. Age is mandatory if the passenger is a 'child sharing'. In all other cases, age is optional.

5.1.19 <PASSENGER>

Type: tag

Optional – no

Attribute – TYPE - Adult or child sharing. The ID associated with passenger type will be 1=Adult and 2=CHILD

Description – 'Adult' or 'child sharing'. The ID associated with a passenger type will be passed in the type attribute.

5.1.20 <FIRST_NAME>

Type: String (50 characters allowed)
Optional - no

Description – First name of passenger

5.1.21 <LAST_NAME>

Type: String (30 characters allowed)
Optional - no

Description – Last name of passenger

5.1.22 <AGE>

Type: Integer
Optional – yes

Description – Age of passenger. This is mandatory in the case of a passenger of type 'child sharing'.

5.1.23 <TITLE>

Type: String
Optional – yes

Description – Passenger title. Valid values include – Mr, Miss, Mrs, Ms and Chd

5.1.24 <OPTIONS>

Type: tag
Optional - no

Description – Main tag that will contain multiple tags of option.

5.1.25 <OPTION>

Type: tag
Optional - no

Description – Individual room/option tag

5.1.26 <OPTION_DATE>

Type: Date
Optional - no

Description – Date on which option is booked

5.1.27 <OPTION_ID>

Type: Integer
Optional - no

Description – ID on Travel Studio database which uniquely identifies a hotel option (Room Type).

- 5.1.28 <QUANTITY>**
Type: Integer
Optional - no

Description – No of rooms required for the specified <OPTION_ID>
- 5.1.29 <NO_OF_ADULTS>**
Type: Integer
Optional - no

Description – Adults per option
- 5.1.30 <NO_OF_CHILDREN>**
Type: Integer
Optional - no

Description – Children per option
- 5.1.31 <SELL_PRICE_ID>**
Type: Integer
Optional - no

Description – Sell Price ID that identifies the pricing in the Travel Studio system. There would be a Sell Price ID for each day being booked and this would confirm the price that is being used.
- 5.1.32 <CHILDREN>**
Type: Tag
Optional - yes

Description – Structure to hold child details for each option. Required when children are present.
- 5.1.33 <AGES >**
Type: Tag
Optional - no

Description – Tag structure to hold ages
- 5.1.34 <AGE>**
Type: Integer
Optional - no

Description – Age of child
- 5.1.35 <COUNT>**
Type: Integer
Optional - no

Description – Number of children of the same age specified in <COUNT>
- 5.1.36 <AVAILABLE_ONLY>**
Type: String
Optional - yes

Description – This is an optional element, which is effective only if the similar flag at service level is set to false (which implies book 'on-request' options also). This allows the API client to book 'available' and 'on-request' options. This flag is only used when service level AVAILABLE_ONLY flag is set to FALSE. The default if this flag is not provided is TRUE which means book only 'available' options.

5.1.37 <NOTES>

Type: Tag
Optional - yes

Description – Contains one note tag

5.1.38 <NOTE>

Type: String
Optional - no

Description – Note to booked service. Only one booking <NOTE> tag is supported.

The Code example below shows how the Booking.asp file would be called. This example is given using visual basic code (in an ASP page). API clients using a different platform need to look for an equivalent approach.

```
var g_XMLReq = new ActiveXObject("Microsoft.XMLHTTP");  
g_XMLReq.open("POST", "http://[Server_Name]/Booking.asp ", false);  
g_XMLReq.send( unescape(XMLdata) );  
/* the result of the script that is resulting XML will be returned in */  
g_XMLReq.responseText  
/* which can be parsed to extract required data. */
```

5.2 **Booking Response**

Output Result

```
<BOOKING_CONFIRMATION>
  <VERSION_HISTORY APPLICATION_NAME="TS HOTELS API 1.0" XML_FILE_NAME="Response.XML"
    LICENCE_KEY="LicKey" TS_API_VERSION="v309.0.0">
    <XML_VERSION_NO>1.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <BOOKING_NAME />
  <BOOKING_REFERENCE />
  <STATUS />
</BOOKING_CONFIRMATION>
```

5.2.1 **<BOOKING_NAME>**

Type: tag

Description – Booking name

5.2.2 **<BOOKING_REFERENCE>**

Type: tag

Description – Booking reference generated by Travel Studio

5.2.3 **<STATUS>**

Type: tag

Description – Status indicates whether booking request is confirmed or not. This will return either “**CONFIRMED**” or “**On Request**”

5.3 **Booking Details Request**

Here we can verify details of the booking. This is often used to get updated details regarding an “On Request” booking

```
<BOOKING_DETAILS>
  <VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="XMLFileName" LICENCE_KEY="LicKey"
    TS_API_VERSION="3.5.6">
    <XML_VERSION_NO>3.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <BOOKING>
    <BOOKING_REFERENCE_NO />
  </BOOKING>
</BOOKING_DETAILS>
```

5.3.1 **<BOOKING_REFERENCE_NO>**

Type: String

Optional: No

Description – JacTravel Booking Reference

5.4 Booking Details Response

Output Result

```
<BOOKING_DETAILS>
  <VERSION_HISTORY APPLICATION_NAME="TS HOTELS API 1.0" XML_FILE_NAME="Response.XML"
  LICENCE_KEY="LicKey" TS_API_VERSION="v309.0.0">
    <XML_VERSION_NO>1.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <BOOKING>
    <BOOKING_REFERENCE_NO />
    <CLIENT_NAME />
    <BOOKING_NAME />
    <BOOKING_DATE />
    <BOOKING_CURRENCY />
    <BOOKING_START_DATE />
    <BOOKING_END_DATE />
    <NUMBER_OF_NIGHTS />
    <TOTAL_ADULTS />
    <TOTAL_CHILDREN />
    <BOOKING_STATUS />
    <TOTAL_BOOKING_PRICE />
    <PAX_OCCUPANCYS>
      <PAX_OCCUPANCY TYPE="">
        <OCCUPANCYID />
        <NO_OF_PAX />
      </PAX_OCCUPANCY>
    </PAX_OCCUPANCYS>
    <PASSENGERS>
      <PASSENGER TYPE="">
        <PASSENGERID />
        <FIRST_NAME />
        <LAST_NAME />
      </PASSENGER>
    </PASSENGERS>
    <SERVICES>
      <BOOKED_SERVICE BOOKEDSERVICEID="" SERVICEID="" CANCELLATION_POLICY="">
        <OPTIONS>
          <BOOKED_OPTION SPLITID="" ISEXTRA="">
            <BOOKED_OPTION_ID />
            <BOOKED_OPTION_CURRENCY />
            <BOOKED_OPTION_STATUS />
            <BOOKED_OPTION_TOTAL_AMOUNT />
            <MEAL_PLAN />
            <OPTION_ID />
            <OPTION_NAME />
            <BOOKED_OPTION_IN_DATE />
            <BOOKED_OPTION_OUT_DATE />
            <NO_OF_NIGHTS />
            <NO_OF_ADULTS />
            <QUANTITY />
            <NO_OF_CHILDREN />
            <CHILDREN>
              <AGES>
                <AGE />
                <COUNT />
              </AGES>
            </CHILDREN>
          </BOOKED_OPTION>
        </OPTIONS>
      </BOOKED_SERVICE>
    </SERVICES>
  </BOOKING>
</BOOKING_DETAILS>
```

- 5.4.1 <BOOKING_REFERENCE_NO>**
Type: tag
Description – JacTravel Booking Reference Number
- 5.4.2 <CLIENT_NAME>**
Type: String
Description – API Client name
- 5.4.3 <BOOKING_NAME>**
Type: String
Description –Booking name
- 5.4.4 <BOOKING_DATE>**
Type: date
Description – Date on which booking is made. Supplied in dd mmm yyyy
- 5.4.5 <BOOKING_CURRENCY>**
Type: String
Description – Booking currency
- 5.4.6 <BOOKING_START_DATE>**
Type: date
Description – Booking Start Date. Supplied in dd mmm yyyy
- 5.4.7 <BOOKING_END_DATE>**
Type: date
Description – Booking End Date, date of departure inclusive. Supplied in dd mmm yyyy
- 5.4.8 <NUMBER_OF_NIGHTS>**
Type: integer
Description – No of nights
- 5.4.9 <TOTAL_ADULTS>**
Type: String
Optional – yes
Description – Total number of adults in the booking
- 5.4.10 <TOTAL_CHILDREN>**
Type: Integer
Description – Total children in booking

5.4.11 <BOOKING_STATUS>

Type: String

Description: Current Status of booking

5.4.12 <TOTAL_BOOKING_PRICE>

Type: Double

Description – Total booking price

5.4.13 <PAX_OCCUPANCYS>

Type: tag

Description – Structure to hold multiple PAX_OCCUPANCY.

5.4.14 <PAX_OCCUPANCY>

Type: tag

Optional – no

Attribute – TYPE - Adult or child sharing. The ID associated with passenger type will be 1=Adult and 2=CHILD

Description – Structure to hold passenger numbers and their associated occupancy. Each tag displays the no. of passengers staying in each kind of room. For example if there are 5 adult passengers staying in 2 singles and 1 triple, then there will be 2 tags. One specifying TYPE = 1 (adult) of passengers in single (Occupancyid =1) and NO_OF_PAX = 2 and another tag specifying TYPE = 1 (adult) in triple and NO_OF_PAX = 3.

5.4.15 <OCCUPANCYID>

Type: Integer

Description – Type of room -Family room types require the number of children and all their ages.

5.4.16 <NO_OF_PAX>

Type: Integer

Description – No of passengers staying in the same type of room

5.4.17 <PASSENGERS>

Type: tag

Description – Tag that will hold passenger information, such as name, age etc. Age is mandatory if the passenger is a 'child sharing'. In all other cases, age is optional.

5.4.18 <PASSENGER>

Type: tag

Optional – no

Attribute – TYPE - Adult or child sharing. The ID associated with passenger type will be 1=Adult and 2=CHILD

Description – 'Adult' or 'child sharing'. The ID associated with a passenger type will be passed in the type attribute.

5.4.19 <PASSENGERID>

Type: String

Description – The Unique identifier for a passenger, by which a passenger id is identified in Travel Studio.

5.4.20 <FIRST_NAME>

Type: String

Description – First name of passenger

5.4.21 <LAST_NAME>

Type: String

Description – Last name of passenger

5.4.22 <SERVICES>

Type: tag

Description – Main tag that will contain multiple tags of booked service.

5.4.23 <BOOKED_SERVICE>

Type: tag

Description – Individual booked option details

Attributes:

- **BOOKEDSERVICEID:** Unique identifier for a booked service in Travel Studio.
- **SERVICEID:** Service ID
- **CANCELLATION_POLICY:** legacy tag that clients should ignore.

5.4.24 <OPTIONS>

Type: tag

Description – Tag containing options

5.4.25 <BOOKED_OPTION>

Type: tag

Description: Booked option structure containing all booked room details.

Attributes:

- **SPLITID:** SplitID of booked option. Only an internal system generated number.
- **ISEXTRA:** legacy tag that clients should ignore

5.4.26 <BOOKED_OPTION_ID>

Type: Integer

Description – ID by which this booked room is identified in the Travel Studio system. Internal system generated number only.

- 5.4.27 <BOOKED_OPTION_CURRENCY>**
Type: Double
Description: Currency of booked option.
- 5.4.28 <BOOKED_OPTION_STATUS>**
Type: Double
Description: The Status of an individual room as booked within the Travel Studio system
- 5.4.29 <BOOKED_OPTION_TOTAL_AMOUNT>**
Type: Double
Description: Total cost of the option/room. This value would use the currency code returned in BOOKING_CURRENCY at booking level.
- 5.4.30 <MEAL_PLAN>**
Type: String
Description: Meal plan associated with the room booked.
- 5.4.31 <OPTION_ID>**
Type: Integer
Description – ID by which this booked room is identified in the Travel Studio system.
- 5.4.32 <OPTION_NAME>**
Type: String
Description: Name of Room in Travel Studio system.
- 5.4.33 <BOOKED_OPTION_IN_DATE>**
Type: String
Description: In Date of the booked room in format dd mmm yyyy
- 5.4.34 <BOOKED_OPTION_OUT_DATE>**
Type: String
Description: Out Date of the booked room in format dd mmm yyyy
- 5.4.35 <NO_OF_NIGHTS>**
Type: Integer
Description: No of nights in booked room
- 5.4.36 <NO_OF_ADULTS>**
Type: Integer
Description – No of Adults in booked room

5.4.37 <QUANTITY>

Type: Integer

Description – No of rooms required for the specified <OPTION_ID>

5.4.38 <NO_OF_CHILDREN>

Type: Integer

Description – No of Children in booked room

5.4.39 <SELL_PRICE_ID>

Type: Integer

Description – Sell Price ID that identifies the pricing in the Travel Studio system. There would be a Sell Price ID for each day being booked and this would confirm the price that is being used.

5.4.40 <CHILDREN>

Type: Tag

Description – Structure to hold child details for each option.

5.4.41 <AGES >

Type: Tag

Description – Tag structure to hold ages

5.4.42 <AGE>

Type: Integer

Description – Age of child

5.4.43 <COUNT>

Type: Integer

Description – Number of children of the same age specified in <COUNT>

Split rates implementation:

In travel studio, rooms are returned according to price. If a room is booked that spans over two different rates then the Booking Details API will return two tags for that same room.

Consider the following example:

A single room is booked for 5 nights. The In-Date is 29th March and Out-date is 3rd April. The prices as defined in travel studio are different for March and April. As such, when the booking details are fetched, the system will return two booked option records: one with an in-date of 29 Mar and an out-date of 1 Apr, the other with an in-date of 1 Apr and an out-date of 3 Apr. If the same rate applied to the entire date range requested then a single tag would be returned with an in-date of 29 Mar and an out-date of 3rd Apr.

5.5 **Booking Statues**

There are a number of possible bookings status that can be utilized by Travel Studio, most of the statues below are for accounting reasons or post stay investigations:

Booking Status	Description	Mapping
Alloc + Extra	"On Request" booking that is confirmed	Confirmed
Alternative Conf	"On Request" booking that uses different hotel to original	Confirmed
Amendment	Booking amended	Confirmed
Book Out - Denied	Alternative hotel for confirmed booking denied	Confirmed
Book Out Alternative	Alternative hotel for confirmed booking confirmed	Confirmed
Cancelled	Cancelled booking	Cancelled
Client Staff	Agent booking	Confirmed
Confirmed	Confirmed booking	Confirmed
Customer Care	Post dates of stay enquires	Confirmed
Customer Care Admin	Post dates of stay enquires	Confirmed
Do Not Process	Booking being held by accounts	Confirmed
Extra To Allocation	"On Request" booking is confirmed	Confirmed
Failure Booking	Booking failed during processing	Cancelled
Failure Cancellation	Cancellation attempt failed	Confirmed
Late Cancellation	Late cancellation charges applied to booking	Late Cancellation
Local Late Cancellation	Late cancellation charges applied to booking	Late Cancellation
No Show	Only occurs after a confirmed booking	Confirmed
NOT Available	"On Request" booking denied	Not available
On Request	No availability found at time of booking. Reservations to inquire if additional rooms are available	On Request

There is also a range of status that are used by Accounts but this only occurs after a confirmed booking, the statuses are listed below for completeness:

- Accs Action
- Accs Action Bk Out
- Accs Action Cust Care
- Accs Action Customer Care Admin
- Accs Action No Show
- Accs Action P/L
- Accs ONLY Client NOT Charged
- Accs ONLY Closed
- Accs ONLY Closed Bk Out
- Accs ONLY Closed Cust Care
- Accs ONLY Closed Cust Care Admin
- Accs ONLY Closed FX
- Accs ONLY Closed No Show
- Accs ONLY Incomplete
- Accs ONLY Late Cxl Charge
- Accts Action - Returned to Ops

6 API Cancellation Process

Cancellation Process can be broken down into 3 steps

1. Cancellation Policy Request for service option
2. Cancellation Policy Request for Booking
3. Cancellation Request

The combination of the above 3 requests allows for complete transparency of all cancellation charges both prior and post booking.

6.1 Cancellation Policy Request

```
<CANCELLATION_POLICY_DETAILS_REQUEST>
<VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="XMLFileName" LICENCE_KEY="LicKey"
TS_API_VERSION="3.5.6">
  <XML_VERSION_NO>3.0</XML_VERSION_NO>
</VERSION_HISTORY>
  <BODY>
    <BOOKING_CHARGE>
      <BOOKING_REFERENCE />
    </BOOKING_CHARGE>
    <SERVICE_CHARGE>
      <OPTION_ID />
      <START_DATE />
      <END_DATE />
    </SERVICE_CHARGE>
  </BODY>
</CANCELLATION_POLICY_DETAILS_REQUEST>
```

6.1.1 <BODY>

Type – tag

Optional - No

Description – The <Body> element contains the entire request

6.1.2 <BOOKING_CHARGE>

Type – tag

Optional – Yes

Description – Two types of searches are possible namely “Booking Reference based search” or “Service based Option ID search”. Both types of searches would be mutually exclusive and in a given cancellation policy request, user would pass either the <Booking_Charge> element or the <Service_Charge> element. Only one of either element is supported per request.

6.1.3 <BOOKING_REFERENCE>

Type: tag

Optional - no

Description – Contains JacTravel Booking Reference Number that would be used to generate the appropriate cancellation policy response.

6.1.4 <SERVICE_CHARGE>

Type – tag

Optional – Yes

Description – Two types of searches are possible namely “Booking Reference based search” or “Service based Option ID search”. Both types of searches would be mutually exclusive and in a given cancellation policy request, pass either the <BookingCharge> element or the <ServiceCharge> element. Only one of either element is supported per request.

6.1.5 <OPTION_ID>

Type: Integer

Optional - no

Description – OptionID of the requested cancellation policy

6.1.6 <START_DATE>

Type: Date

Optional - no

Description – Specify the start date for the cancellation policy requested. The end date would be inclusive of departure date. For example searching for a policy over 2 nights starting 15th of May, the <Start_Date> and <End_Date> tags would be as follows:

```
<Start_Date>15-May-2009</Start_Date>
```

```
<End_Date>17-May-2009</End_Date>
```

6.1.7 <END_DATE>

Type: Date

Optional - no

Description – The <End_Date> tag would be used to specify the end date for the cancellation policy requested. The end date would be inclusive of departure date. For example if one is searching for a policy over 2 nights starting 15th of May the <Start_Date> and <End_Date> would be as follows:

```
<Start_Date>15-May-2009</Start_Date>
```

```
<End_Date>17-May-2009</End_Date>
```

6.2 Cancellation Policy Response

Responses are interpreted according to the request type, either Option Id or Booking Reference request.

Option Id Response:

<CHARGE_TYPE> provides the method of calculating the charge and can be one of 4 types:

Fixed fee – Where the chargeable amount would be fixed and applicable for each option booked.

Percentage of full cost – Where the percentage would be charged.

Number of nights – Where a specified number of nights would be charged.

Percentage of first night – Where the percentage would be entered charged.

<CHARGE_VALUE> needs to be compared to <CHARGE_TYPE> to understand whether the value represents a dollar value, percentage or number or nights.

Value returned in <CHARGE> tag when passing OPTION_ID is a repeat of the <CHARGE_VALUE> information.

Charge Type	Charge Value
Fixed Fee	Dollar value per option booked
Percentage of full cost	% value of full cost of booking
Number of Nights	Specified number would be charged (cannot exceed total nights stayed)
Percentage of first night	% value of first night booked

Booking Reference Response:

Returns cancellation charges of the requested booking on a per day basis

Output Response:

```
<CANCELLATION_POLICY_DETAILS_RESPONSE>
  <VERSION_HISTORY APPLICATION_NAME="TS HOTELS API 1.0" XML_FILE_NAME="Response.XML"
  LICENCE_KEY="LicKey" TS_API_VERSION="v309.0.0">
    <XML_VERSION_NO>1.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <BODY>
    <OPTION ID="" / <SERVICE ID="">
      <CHARGE_DESCRIPTION />
      <CHARGE_TYPE />
      <METHOD>
        <DAYS_BEFORE_CHECK_IN />
        <CHARGE_VALUE />
      </METHOD>
      <CANCELLATION_POLICY_BREAK_DOWN>
        <CURRENCY />
        <DAYS>
          <CHARGE_DATE />
          <CHARGE />
        </DAYS>
      </CANCELLATION_POLICY_BREAK_DOWN>
    </OPTION ID="" / </SERVICE ID="">
  </BODY>
</CANCELLATION_POLICY_DETAILS_RESPONSE>
```

6.2.1 <BODY>

Type – tag

Optional - No

Description – The <Body> element contains the entire response

6.2.2 <OPTION> or <SERVICE>

Type – tag

Description – The <Option> element would contain the ID attribute that would contain the Option_Id requested.

The <Service> element would contain the ID attribute that would contain the Service_Id for the requested Booking Reference number passed.

6.2.3 <CHARGE_DESCRIPTION>

Type: String

Description – This would be the description of the cancellation policy. The description displayed would be accurate in how cancellation charges would be calculated, for example:

```
<OPTION ID="65874">
  <CHARGE_DESCRIPTION>2 day(s) prior to arrival</CHARGE_DESCRIPTION>
  <CHARGE_TYPE>Percentage of first night</CHARGE_TYPE>
```

6.2.4 <CHARGE_TYPE>

Type – String

Description – This would contain the charge type cancellation policy details. The allowed values and their meaning for charge type would be as follows:

Fixed fee – Where the chargeable amount would be fixed and applicable for each option booked.

Percentage of full cost – Where the percentage would be charged.

Number of nights – Where a specified number of nights would be charged.

Percentage of first night – Where the percentage would be entered charged.

6.2.5 <METHOD>

Type: tag

Description – The element would repeat for any applicable cancellation policy days to allow for complete transparency on how cancellation charges are applied.

6.2.6 <DAYS_BEFORE_CHECK_IN>

Type: Integer

Description – This would be the number of days prior to the check-in date within which if a booking is cancelled cancellation charges would apply.

6.2.7 <CHARGE_VALUE>

Type: integer

Description – Charge Value would be used in conjunction with <Days_Before_Check_In> to see how much charge is applied, for example:

```
<CHARGE_TYPE>Fixed Fee</CHARGE_TYPE>
<METHOD>
  <DAYS_BEFORE_CHECK_IN>5</DAYS_BEFORE_CHECK_IN>
  <CHARGE_VALUE>50.00</CHARGE_VALUE>
</METHOD>
```

6.2.8 <CANCELLATION_POLICY_BREAK_DOWN>

Type: tag

Description – This would contain cancellation policy related information on a per day basis.

6.2.9 <CURRENCY>

Type: String

Description – This would simply return the currency ISO code that is relevant to the booking.

6.2.10 <DAYS>

Type: Integer

Description – This would be the number of days prior to check-in date. This element would contain the <Charge_Date> and <Charge> tags for each day from the cancellation policy prior to and inclusive of the booking start date.

Each <DAYS> container represents a 24hr block of time.

6.2.11 <CHARGE_DATE>

Type: String

Description – Contains a date for each day starting from the check in date depending on how the Cancellation Policy is defined and proceeds in descending order

For example if the CP states that

If cancelled 1 or 2 days in advance, cancellation charge is 100% of full cost

If cancelled 3 days in advance, cancellation charge is 50% of full cost

Assuming the a hotel is booked for 3 nights from 3rd Feb to 6th Feb for a total cost of 400.00 Euro and also assuming that the service commencement time is 12:00 mid day, if the booking is made on 15th Jan, the cancellation policy XML in the booking response would appear as below:

```
<CANCELLATION_POLICY_BREAK_DOWN>
  <CURRENCY>EUR</CURRENCY>
  <DAYS>
    <CHARGE_DATE>Feb 3 2009 11:59</CHARGE_DATE>
    <CHARGE>400.00</CHARGE>
  </DAYS>
  <DAYS>
    <CHARGE_DATE>Feb 2 2009 12:00</CHARGE_DATE>
    <CHARGE>400.00</CHARGE>
  </DAYS>
  <DAYS>
    <CHARGE_DATE>Feb 1 2009 12:00</CHARGE_DATE>
    <CHARGE>200.00</CHARGE>
  </DAYS>
</CANCELLATION_POLICY_BREAK_DOWN>
```

- Any attempt to cancel the booking AFTER 3rd Feb 11.59AM would receive the following error message: "This booking cannot be cancelled as arrival time has passed. Please contact JACTRAVEL Reservations".
- If the cancellation request is made BEFORE 3rd Feb 11:59:00 (11:59 AM inclusive) and AFTER 2nd Feb 12:01:00 (12:01:00 inclusive) the charge would be 100% of the booking cost which is 400.00.
- If the cancellation request is made BEFORE 2nd Feb 12:00 mid day (12:00 AM inclusive) and AFTER 1st Feb 12:01:00 (12:01:00 inclusive) the charge would be 100% of the booking cost which is 400.00.
- If the cancellation request is made BEFORE 1st Feb 12:00 mid day (12:00 AM inclusive) and AFTER 31st Jan 12:01:00 (12:01:00 inclusive) the charge would be 50% of the booking cost which is 200.00.

Times returned will be in 24 hour blocks with the final 24th hour being implied. In the above example the final time of 31st Jan 12:01:00 which completes the last 24 hour block up to 1st Feb 12:00 is implied thus completing the full 3 day cancellation policy assigned.

Times returned also take into consideration the exact time of the request and only return relevant dates and times. Again assuming that a hotel is booked for 3 nights from 3rd Feb to 6th Feb and also assuming that the service commencement time is 12:00 mid day, if the booking is made on 15th Jan and the *policy request is made on 1 Feb 15:00:00*, the cancellation policy XML in the booking response would appear as below:

```
<CANCELLATION_POLICY_BREAK_DOWN>
  <CURRENCY>EUR</CURRENCY>
  <DAYS>
    <CHARGE_DATE>Feb 3 2009 11:59</CHARGE_DATE>
    <CHARGE>400.00</CHARGE>
  </DAYS>
  <DAYS>
    <CHARGE_DATE>Feb 2 2009 12:00</CHARGE_DATE>
    <CHARGE>400.00</CHARGE>
  </DAYS>
</CANCELLATION_POLICY_BREAK_DOWN>
```

Note that as the final 24 hour period of BEFORE 1st Feb 12:00 mid day (12:00 AM inclusive) and AFTER 31Jan 12:01:00 (12:01:00 inclusive) is no longer valid when taking into consideration the time of request.

6.2.12 <CHARGE>

Type: Integer

Description – This would contain the cancellation charge amount calculated based on the value of Charge Type for each date represented in the <Charge_Date> tag

6.3 *Cancellation Request*

Cancellation Request

```
<BOOKING_CANCELLATION>
<VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="XMLFileName" LICENCE_KEY="LicKey"
TS_API_VERSION="3.5.6">
  <XML_VERSION_NO>3.0</XML_VERSION_NO>
</VERSION_HISTORY>
<BOOKING_REFERENCE />
</BOOKING_CANCELLATION>
```

6.3.1 <BOOKING_REFERENCE>

Type: String
Optional: No

Description – JacTravel Booking Reference

6.4 *Cancellation Response*

Cancellation response returns the current status of the booking displaying any applied cancellation charges if applicable.

```
<BOOKING_CANCELLATION_RESPONSE>
<VERSION_HISTORY APPLICATION_NAME="TS HOTELS API 1.0" XML_FILE_NAME="Response.XML"
LICENCE_KEY="LicKey" TS_API_VERSION="v309.0.0">
  <XML_VERSION_NO>1.0</XML_VERSION_NO>
</VERSION_HISTORY>
  <BODY>
    <BOOKING_REFERENCE />
    <STATUS />
    <CANCELLATIONCHARGES>
      <CURRENCY />
      <CHARGE />
    </CANCELLATIONCHARGES>
  </BODY>
</BOOKING_CANCELLATION_RESPONSE>
```

6.4.1 <BOOKING_REFERENCE>

Type: String

Description – JacTravel Booking Reference number

6.4.2 <STATUS>

Type: String

Description –Booking status after the booking is cancelled

Status	Description
Cancelled	Booking cancelled
Late Cancellation	Booking cancelled with late charges

6.4.3 <CANCELLATIONCHARGES>

Type: String

Description – This element is mandatory and would contain the cancellation charges attracted (Including 0.00).

6.4.4 <CURRENCY>

Type – tag

Description – This would contain the booking currency.

6.4.5 <CHARGE>

Type – String

Description – This is the cancellation charge calculated based on the cancellation policy applied.

7 Error Handling

If an error is detected in a request sent to the API Gateway the following Error XML message will be returned.

```
<ERROR_CATEGORY="">
  <VERSION_HISTORY APPLICATION_NAME="Application" XML_FILE_NAME="XMLFileName" LICENCE_KEY="LicenceKey"
  TS_API_VERSION="TSAPIVersion">
    <XML_VERSION_NO>XMLVersionNumber</XML_VERSION_NO>
  </VERSION_HISTORY>
  <ERROR_NUMBER />
  <ERROR_DESC />
  <EXTENDED_ERROR_DESC />
</ERROR>
```

We would always suggest that API customers put in place an appropriate level of error trapping into their applications for instances where the server, connection or API fail to respond. An error trapping strategy will enable you to both catch any errors when they occur and ensure that when your system goes live there is a suitable level of error trapping within your application to handle exceptions.

Please do not assume within your finished application that the API service will always be there, sometimes servers and services fail, and these failures need to be catered for within your application workflow. Some of the server failures will come in Microsoft format or API errors, but may also extend to communication errors on network or Internet connections problems, where possible these should all be trapped and handled.

7.1.1 <ERROR_NUMBER>

Type: Integer

Description – The error number

7.1.2 <ERROR_DESC>

Type: String

Description – A message describing the error

7.1.3 <EXTENDED_ERROR_DESC>

Type: String

Description – Extended error description.

8 Appendices

8.1 Booking conversation

The following request/response examples show the process of searching for, booking, reviewing the booking details, and canceling a hotel.

The license key and service, option and sell price IDs are all for demonstration only.

In this example we'll simulate a request for the following rooms in Paris:

2x "Family room 1" (Occupancy ID 7, 2 Adults + 1 Child)

1x "Single room" (Occupancy ID 1, 1 Adults)

(5 adults and two children in total, child ages 8 and 10)

We are looking for a two night stay in Paris from 1st December 2009, and the booking will be on an "Available Only" basis, i.e. services available "from allocation" only.

8.2 Service Search Request and Response

For **Search** requests (both service search, and availability and price), the values are **NOT intuitive**, i.e.

<BOOKING_START_DATE> is the date that the customer is expected to arrive at the hotel.

<BOOKING_END_DATE> is **NOT** inclusive of the date of departure, i.e. the value is the date of the **last night stayed at the hotel**.

<NUMBER_OF_NIGHTS> is the intuitive value **minus one**

For example to search for a one night stay on 21st December:

<BOOKING_START_DATE>21 Dec 2009</BOOKING_START_DATE>

<NUMBER_OF_NIGHTS>0</NUMBER_OF_NIGHTS>

Note how <NUMBER_OF_NIGHTS> is not as you would intuitively expect.

Another example, for a 7 night stay from 10th April 2009 (last night stayed at the hotel is 16th April, Date of departure from the hotel is 17th April)

<BOOKING_START_DATE>10 Apr 2009</BOOKING_START_DATE>

<NUMBER_OF_NIGHTS>6</NUMBER_OF_NIGHTS>

It is important to use the correct values for search requests- if they are incorrect, prices returned may be subject to length-of-stay based special offers that could potentially not apply when the correct values are used for the booking request.

Geolocational Search

```
<SERVICE_SEARCH_REQUEST>
  <VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="XMLFileName" LICENCE_KEY="FA39F277-
    944B-971C-BC64-79E80DEE81C4" TS_API_VERSION="TSAPIVersion">
    <XML_VERSION_NO>3.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <GEO_LOCATION_NAME>Paris</GEO_LOCATION_NAME>
  <START_DATE>01 Dec 09</START_DATE>
```

```

<NUMBER_OF_NIGHTS>1</NUMBER_OF_NIGHTS> <-- Note value reduced by 1 compared to booking requests-->
<AVAILABLE_ONLY>true</AVAILABLE_ONLY>
<GET_START_PRICE>true</GET_START_PRICE>
<ROOM_REPLY>
  <ALL_ROOM />
</ROOM_REPLY>
<ROOMS_REQUIRED>
  <ROOM>
    <OCCUPANCY>1</OCCUPANCY>
    <QUANTITY>1</QUANTITY>
  </ROOM>
  <ROOM>
    <OCCUPANCY>7</OCCUPANCY>
    <QUANTITY>2</QUANTITY>
    <CHILDREN>
      <CHILD_RATE CHILD_QUANTITY="1" CHILD_AGE="8" />
      <CHILD_RATE CHILD_QUANTITY="1" CHILD_AGE="10" />
    </CHILDREN>
  </ROOM>
</ROOMS_REQUIRED>
</SERVICE_SEARCH_REQUEST>

```

Service Search Response – available rooms

```

<SERVICE_SEARCH_RESPONSE>
  <VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="FileName" LICENCE_KEY="LicKey"
    TS_API_VERSION="3.0">
    <XML_VERSION_NO>1.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <SERVICES>
    <SERVICE SERVICE_ID="12036" SERVICE_NAME="Test Millennium Opera Paris" LOCATION="Opera" RATING="4 *
      +" AVAILABLE="Yes" ISRECOMMENDEDPRODUCT="0" STARTING_PRICE="39.9000" CURRENCY="EUR">
      <OPTIONS>
        <OPTION>
          <OPTIONID>34176</OPTIONID>
          <OPTION_NAME>Single Room</OPTION_NAME>
          <MinAdult>1</MinAdult>
          <MaxAdult>1</MaxAdult>
          <MinChild>0</MinChild>
          <MaxChild>0</MaxChild>
          <ChildMaxAge>0</ChildMaxAge>
          <OCCUPANCY>1</OCCUPANCY>
          <PRICES>
            <PRICE>
              <PRICE_DATE>01 Dec 2009</PRICE_DATE>
              <SELL_PRICE_ID>721252</SELL_PRICE_ID>
              <SELL_PRICE_AMOUNT>39.9000</SELL_PRICE_AMOUNT>
              <SELL_CURRENCY_CODE>EUR</SELL_CURRENCY_CODE>
              <MEAL_PLAN>
                <MEAL_PLAN_TEXT>Bed & Continental Buffet</MEAL_PLAN_TEXT>
                <MEAL_PLAN_TYPE>
                  <INCLUDESBREAKFAST>1</INCLUDESBREAKFAST>
                  <INCLUDESLUNCH>0</INCLUDESLUNCH>
                  <INCLUDESDINNER>0</INCLUDESDINNER>
                </MEAL_PLAN_TYPE>
              </MEAL_PLAN>
              <ORIGINAL_SELL_PRICE_AMOUNT>39.9000</ORIGINAL_SELL_PRICE_AMOUNT>
            </PRICE>
            <PRICE>
              <PRICE_DATE>02 Dec 2009</PRICE_DATE>
              <SELL_PRICE_ID>721252</SELL_PRICE_ID>
              <SELL_PRICE_AMOUNT>39.9000</SELL_PRICE_AMOUNT>
            </PRICE>
          </PRICES>
        </OPTION>
      </OPTIONS>
    </SERVICE>
  </SERVICES>
</SERVICE_SEARCH_RESPONSE>

```

```

        <SELL_CURRENCY_CODE>EUR</SELL_CURRENCY_CODE>
        <MEAL_PLAN>
            <MEAL_PLAN_TEXT>Bed & Continental Buffet</MEAL_PLAN_TEXT>
            <MEAL_PLAN_TYPE>
                <INCLUDESBREAKFAST>1</INCLUDESBREAKFAST>
                <INCLUDESLUNCH>0</INCLUDESLUNCH>
                <INCLUDESDINNER>0</INCLUDESDINNER>
            </MEAL_PLAN_TYPE>
        </MEAL_PLAN>
        <ORIGINAL_SELL_PRICE_AMOUNT>39.9000</ORIGINAL_SELL_PRICE_AMOUNT>
    </PRICE>
</PRICES>
<OPTION_STATUS>AVAILABLE</OPTION_STATUS>
<RULE_TEXT />
</OPTION>
<OPTION>
    <OPTIONID>34177</OPTIONID>
    <OPTION_NAME>Double/Twin + 1 Child</OPTION_NAME>
    <MinAdult>2</MinAdult>
    <MaxAdult>2</MaxAdult>
    <MinChild>1</MinChild>
    <MaxChild>1</MaxChild>
    <ChildMaxAge>0</ChildMaxAge>
    <OCCUPANCY>7</OCCUPANCY>
    <PRICES>
        <PRICE>
            <PRICE_DATE>01 Dec 2009</PRICE_DATE>
            <SELL_PRICE_ID>721253</SELL_PRICE_ID>
            <SELL_PRICE_AMOUNT>52.4400</SELL_PRICE_AMOUNT>
            <SELL_CURRENCY_CODE>EUR</SELL_CURRENCY_CODE>
            <MEAL_PLAN>
                <MEAL_PLAN_TEXT>Bed & Continental Buffet</MEAL_PLAN_TEXT>
                <MEAL_PLAN_TYPE>
                    <INCLUDESBREAKFAST>1</INCLUDESBREAKFAST>
                    <INCLUDESLUNCH>0</INCLUDESLUNCH>
                    <INCLUDESDINNER>0</INCLUDESDINNER>
                </MEAL_PLAN_TYPE>
            </MEAL_PLAN>
            <CHILD_PRICES>
                <CHILD_PRICE>
                    <AGE>8</AGE>
                    <SELL_PRICE_AMOUNT>26.2200</SELL_PRICE_AMOUNT>
                </CHILD_PRICE>
                <CHILD_PRICE>
                    <AGE>10</AGE>
                    <SELL_PRICE_AMOUNT>26.2200</SELL_PRICE_AMOUNT>
                </CHILD_PRICE>
            </CHILD_PRICES>
            <ORIGINAL_SELL_PRICE_AMOUNT>26.2200</ORIGINAL_SELL_PRICE_AMOUNT>
        </PRICE>
    </PRICE>
    <PRICE>
        <PRICE_DATE>02 Dec 2009</PRICE_DATE>
        <SELL_PRICE_ID>721253</SELL_PRICE_ID>
        <SELL_PRICE_AMOUNT>52.4400</SELL_PRICE_AMOUNT>
        <SELL_CURRENCY_CODE>EUR</SELL_CURRENCY_CODE>
        <MEAL_PLAN>
            <MEAL_PLAN_TEXT>Bed & Continental Buffet</MEAL_PLAN_TEXT>
            <MEAL_PLAN_TYPE>
                <INCLUDESBREAKFAST>1</INCLUDESBREAKFAST>
                <INCLUDESLUNCH>0</INCLUDESLUNCH>
                <INCLUDESDINNER>0</INCLUDESDINNER>
            </MEAL_PLAN_TYPE>
        </MEAL_PLAN>
        <CHILD_PRICES>
            <CHILD_PRICE>
                <AGE>8</AGE>

```



```

        <SELL_PRICE_AMOUNT>26.2200</SELL_PRICE_AMOUNT>
      </CHILD_PRICE>
    </CHILD_PRICE>
    <AGE>10</AGE>
    <SELL_PRICE_AMOUNT>26.2200</SELL_PRICE_AMOUNT>
  </CHILD_PRICE>
</CHILD_PRICES>
<ORIGINAL_SELL_PRICE_AMOUNT>26.2200</ORIGINAL_SELL_PRICE_AMOUNT>
</PRICE>
</PRICES>
<OPTION_STATUS>AVAILABLE</OPTION_STATUS>
<RULE_TEXT />
</OPTION>
</OPTIONS>
</SERVICE>
</SERVICES>
</SERVICE_SEARCH_RESPONSE>

```

8.2.1 Availability Search and Response

Note below how <BOOKING_END_DATE> > is **NOT** inclusive of the date of departure, i.e. the value is the date of the **last night stayed at the hotel**.

Typically on a customer website using the production environment, the search result will include many services for the customer to choose from. Because there could be a delay of several minutes between the initial service search results, and the booking, we recommend that a final availability and price search, using the same rooming and passenger criteria is done just before the final booking request is made, in order to verify availability and make sure the allocation has not been taken by another booking and that the prices are correct.

We are using the <VALIDATE_QTY> tag here to verify that the passenger quantities are valid for the rooms specified.

Note that the Child age definition is slightly different from the service search request.

Availability Search Request

```

<HOTEL_AVAILABILITY_AND_PRICE_SEARCH_CRITERIA>
  <VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="XMLFileName" LICENCE_KEY=" FA39F277-
    944B-971C-BC64-79E80DEE81C4 " TS_API_VERSION="TSAPIVersion">
    <XML_VERSION_NO>3.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <SERVICE_ID AVAILABLE_ONLY="TRUE">12036</SERVICE_ID>
  <CLIENT_NAME>API Client</CLIENT_NAME>
  <BOOKING_START_DATE>01 Dec 09</BOOKING_START_DATE>
  <BOOKING_END_DATE>02 Dec 09</BOOKING_END_DATE>
  <ROOM_REPLY>
    <ALL_ROOM/>
  </ROOM_REPLY>
  <PRICING>
    <VALIDATE_QTY/>
  </PRICING>
  <ROOMS_REQUIRED>
    <ROOM>
      <OCCUPANCY>1</OCCUPANCY>
      <QUANTITY>1</QUANTITY>
    </ROOM>
  </ROOMS_REQUIRED>
</HOTEL_AVAILABILITY_AND_PRICE_SEARCH_CRITERIA>

```

```

        <NO_OF_PASSENGERS>1</NO_OF_PASSENGERS>
    </ROOM>
    <ROOM>
        <OCCUPANCY>7</OCCUPANCY>
        <QUANTITY>2</QUANTITY>
        <NO_OF_PASSENGERS>4</NO_OF_PASSENGERS>
        <NO_OF_CHILDREN>2</NO_OF_CHILDREN>
        <AGES_OF_CHILDREN>8,10</AGES_OF_CHILDREN>
    </ROOM>
</ROOMS_REQUIRED>
</HOTEL_AVAILABILITY_AND_PRICE_SEARCH_CRITERIA>

```

Availability Search Response

```

<HOTEL_AVAILABILITY_AND_PRICE_SEARCH_RESULT>
    <VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="FileName" LICENCE_KEY="LicKey"
    TS_API_VERSION="3.0">
        <XML_VERSION_NO>1.0</XML_VERSION_NO>
    </VERSION_HISTORY>
    <SERVICE_ID>12036</SERVICE_ID>
    <RATING>4 * +</RATING>
    <AVAILABILITY>Yes</AVAILABILITY>
    <OPTIONS>
        <OPTION>
            <OPTIONID>34176</OPTIONID>
            <OPTION_NAME>Single Room</OPTION_NAME>
            <MinAdult>1</MinAdult>
            <MaxAdult>1</MaxAdult>
            <MinChild>0</MinChild>
            <MaxChild>0</MaxChild>
            <ChildMaxAge>0</ChildMaxAge>
            <OCCUPANCY>1</OCCUPANCY>
            <PRICES>
                <PRICE>
                    <PRICE_DATE>01 Dec 2009</PRICE_DATE>
                    <SELL_PRICE_ID>721252</SELL_PRICE_ID>
                    <SELL_PRICE_AMOUNT>39.9000</SELL_PRICE_AMOUNT>
                    <SELL_CURRENCY_CODE>EUR</SELL_CURRENCY_CODE>
                    <MEAL_PLAN>
                        <MEAL_PLAN_TEXT>Bed & Continental Buffet</MEAL_PLAN_TEXT>
                        <MEAL_PLAN_TYPE>
                            <INCLUDESBREAKFAST>1</INCLUDESBREAKFAST>
                            <INCLUDESLUNCH>0</INCLUDESLUNCH>
                            <INCLUDESDINNER>0</INCLUDESDINNER>
                        </MEAL_PLAN_TYPE>
                    </MEAL_PLAN>
                    <ORIGINAL_SELL_PRICE_AMOUNT>39.9000</ORIGINAL_SELL_PRICE_AMOUNT>
                </PRICE>
                <PRICE>
                    <PRICE_DATE>02 Dec 2009</PRICE_DATE>
                    <SELL_PRICE_ID>721252</SELL_PRICE_ID>
                    <SELL_PRICE_AMOUNT>39.9000</SELL_PRICE_AMOUNT>
                    <SELL_CURRENCY_CODE>EUR</SELL_CURRENCY_CODE>
                    <MEAL_PLAN>
                        <MEAL_PLAN_TEXT>Bed & Continental Buffet</MEAL_PLAN_TEXT>
                        <MEAL_PLAN_TYPE>
                            <INCLUDESBREAKFAST>1</INCLUDESBREAKFAST>
                            <INCLUDESLUNCH>0</INCLUDESLUNCH>
                            <INCLUDESDINNER>0</INCLUDESDINNER>
                        </MEAL_PLAN_TYPE>
                    </MEAL_PLAN>
                    <ORIGINAL_SELL_PRICE_AMOUNT>39.9000</ORIGINAL_SELL_PRICE_AMOUNT>
                </PRICE>
            </PRICES>
        </OPTION>
    </OPTIONS>

```

```

        </PRICE>
    </PRICES>
    <OPTION_STATUS>AVAILABLE</OPTION_STATUS>
    <RULE_TEXT />
</OPTION>
<OPTION>
    <OPTIONID>34177</OPTIONID>
    <OPTION_NAME>Double/Twin + 1 Child</OPTION_NAME>
    <MinAdult>2</MinAdult>
    <MaxAdult>2</MaxAdult>
    <MinChild>1</MinChild>
    <MaxChild>1</MaxChild>
    <ChildMaxAge>0</ChildMaxAge>
    <OCCUPANCY>7</OCCUPANCY>
    <PRICES>
        <PRICE>
            <PRICE_DATE>01 Dec 2009</PRICE_DATE>
            <SELL_PRICE_ID>721253</SELL_PRICE_ID>
            <SELL_PRICE_AMOUNT>52.4400</SELL_PRICE_AMOUNT>
            <SELL_CURRENCY_CODE>EUR</SELL_CURRENCY_CODE>
            <MEAL_PLAN>
                <MEAL_PLAN_TEXT>Bed & Continental Buffet</MEAL_PLAN_TEXT>
                <MEAL_PLAN_TYPE>
                    <INCLUDESBREAKFAST>1</INCLUDESBREAKFAST>
                    <INCLUDESLUNCH>0</INCLUDESLUNCH>
                    <INCLUDESDINNER>0</INCLUDESDINNER>
                </MEAL_PLAN_TYPE>
            </MEAL_PLAN>
            <CHILD_PRICES>
                <CHILD_PRICE>
                    <AGE>8</AGE>
                    <SELL_PRICE_AMOUNT>26.2200</SELL_PRICE_AMOUNT>
                </CHILD_PRICE>
                <CHILD_PRICE>
                    <AGE>10</AGE>
                    <SELL_PRICE_AMOUNT>26.2200</SELL_PRICE_AMOUNT>
                </CHILD_PRICE>
            </CHILD_PRICES>
            <ORIGNAL_SELL_PRICE_AMOUNT>26.2200</ORIGNAL_SELL_PRICE_AMOUNT>
        </PRICE>
        <PRICE>
            <PRICE_DATE>02 Dec 2009</PRICE_DATE>
            <SELL_PRICE_ID>721253</SELL_PRICE_ID>
            <SELL_PRICE_AMOUNT>52.4400</SELL_PRICE_AMOUNT>
            <SELL_CURRENCY_CODE>EUR</SELL_CURRENCY_CODE>
            <MEAL_PLAN>
                <MEAL_PLAN_TEXT>Bed & Continental Buffet</MEAL_PLAN_TEXT>
                <MEAL_PLAN_TYPE>
                    <INCLUDESBREAKFAST>1</INCLUDESBREAKFAST>
                    <INCLUDESLUNCH>0</INCLUDESLUNCH>
                    <INCLUDESDINNER>0</INCLUDESDINNER>
                </MEAL_PLAN_TYPE>
            </MEAL_PLAN>
            <CHILD_PRICES>
                <CHILD_PRICE>
                    <AGE>8</AGE>
                    <SELL_PRICE_AMOUNT>26.2200</SELL_PRICE_AMOUNT>
                </CHILD_PRICE>
                <CHILD_PRICE>
                    <AGE>10</AGE>
                    <SELL_PRICE_AMOUNT>26.2200</SELL_PRICE_AMOUNT>
                </CHILD_PRICE>
            </CHILD_PRICES>
            <ORIGNAL_SELL_PRICE_AMOUNT>26.2200</ORIGNAL_SELL_PRICE_AMOUNT>
        </PRICE>
    </PRICES>

```

```

                <OPTION_STATUS>AVAILABLE</OPTION_STATUS>
                <RULE_TEXT />
            </OPTION>
        </OPTIONS>
    </SERVICE>
</HOTEL_AVAILABILITY_AND_PRICE_SEARCH_RESULT>

```

So in this example, the total price will be the Adults prices (quoted per room per night) plus the individual child prices (per person per night)
So in this case

Single Room, 1st Dec room price	€39.9000
Family Room 1, 1st Dec room price x2	€104.8800
Family Room 1, 1st Dec Child price (8yo)	€26.2200
Family Room 1, 1st Dec Child price (10yo)	€26.2200
Single Room, 2nd Dec room price	€39.9000
Family Room 1, 2nd Dec room price x2	€104.8800
Family Room 1, 2nd Dec Child price (8yo)	€26.2200
Family Room 1, 2nd Dec Child price (10yo)	€26.2200
Total	€394.4400

8.2.2 Cancellation Policy Request and Response at selected service

Here we will check the cancellation policy related to the specific services Option ID

```

<CANCELLATION_POLICY_DETAILS_REQUEST>
  <VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="XMLFileName" LICENCE_KEY="FA39F277-
    944B-971C-BC64-79E80DEE81C4" TS_API_VERSION="TSAPIVersion">
    <XML_VERSION_NO>3.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <BODY>
    <SERVICE_CHARGE>
      <OPTION_ID>34176</OPTION_ID>
      <START_DATE>01 Dec 09</START_DATE>
      <END_DATE>03 Dec 09</END_DATE>
    </SERVICE_CHARGE>
  </BODY>
</CANCELLATION_POLICY_DETAILS_REQUEST>

```

Cancellation Policy Response:

Here the response clearly shows the cancellation policy on each date of stay.

```
<CANCELLATION_POLICY_DETAILS_RESPONSE>
  <VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="XMLFileName" LICENCE_KEY="FA39F277-
    944B-971C-BC64-79E80DEE81C4" TS_API_VERSION="TSAPIVersion">
    <XML_VERSION_NO>3.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <BODY>
    <OPTION ID="34176">
      <CHARGE_DESCRIPTION>2 day(s) prior to arrival</CHARGE_DESCRIPTION>
      <CHARGE_TYPE>Percentage of first night</CHARGE_TYPE>
      <METHOD>
        <DAYS_BEFORE_CHECK_IN>2</DAYS_BEFORE_CHECK_IN>
        <CHARGE_VALUE>100.00</CHARGE_VALUE>
      </METHOD>
      <CANCELLATION_POLICY_BREAK_DOWN>
        <CURRENCY>EUR</CURRENCY>
        <DAYS>
          <CHARGE_DATE>Dec 01 2009 11:59AM</CHARGE_DATE>
          <CHARGE>100.00</CHARGE>
        </DAYS>
        <DAYS>
          <CHARGE_DATE>Nov 30 2009 12:00PM</CHARGE_DATE>
          <CHARGE>100.00</CHARGE>
        </DAYS>
      </CANCELLATION_POLICY_BREAK_DOWN>
    </OPTION>
  </BODY>
</CANCELLATION_POLICY_DETAILS_RESPONSE>
```

8.2.3 Booking Request and Conformation

Here we will book the service using the SERVICE_ID, OPTION_IDs and SELL_PRICE_IDs discovered in the earlier requests:

```
<BOOKING_DETAILS>
  <VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="XMLFileName" LICENCE_KEY="FA39F277-
    944B-971C-BC64-79E80DEE81C4" TS_API_VERSION="TSAPIVersion">
    <XML_VERSION_NO>3.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <BOOKING>
    <CLIENT_NAME>Client Name</CLIENT_NAME>
    <BOOKING_NAME>Test booking for 1xSingle, 2xFamily Room1</BOOKING_NAME>
    <BOOKING_DATE>20 Jun 2009</BOOKING_DATE>
    <BOOKING_START_DATE>01 Dec 2009</BOOKING_START_DATE>
    <BOOKING_END_DATE>03 Dec 2009</BOOKING_END_DATE>
    <NUMBER_OF_NIGHTS>2</NUMBER_OF_NIGHTS>
    <CLIENT_REFERENCE> Test booking for 1xSingle, 2xFamily Room1</CLIENT_REFERENCE>
    <VALIDATE_QTY />
    <SERVICE_ID AVAILABLE_ONLY="true">12046</SERVICE_ID>
    <TOTAL_ADULTS>5</TOTAL_ADULTS>
    <TOTAL_CHILDREN>2</TOTAL_CHILDREN>
  </BOOKING>
</BOOKING_DETAILS>
```

```

<RETURN_BOOKING_DETAILS/>
<PAX_OCCUPANCYS>
  <PAX_OCCUPANCY TYPE="1">
    <OCCUPANCYID>1</OCCUPANCYID>
    <NO_OF_PAX>1</NO_OF_PAX>
  </PAX_OCCUPANCY>
  <PAX_OCCUPANCY TYPE="1">
    <OCCUPANCYID>7</OCCUPANCYID>
    <NO_OF_PAX>4</NO_OF_PAX>
  </PAX_OCCUPANCY>
  <PAX_OCCUPANCY TYPE="2">
    <OCCUPANCYID>7</OCCUPANCYID>
    <NO_OF_PAX>2</NO_OF_PAX>
  </PAX_OCCUPANCY>
</PAX_OCCUPANCYS>
<PASSENGERS>
  <PASSENGER TYPE="1">
    <FIRST_NAME>Adult 1 - Single Room</FIRST_NAME>
    <LAST_NAME>Adult 1</LAST_NAME>
    <AGE />
  </PASSENGER>
  <PASSENGER TYPE="1">
    <FIRST_NAME>Adult 2 - Family Room</FIRST_NAME>
    <LAST_NAME>Adult 2</LAST_NAME>
    <AGE />
  </PASSENGER>
  <PASSENGER TYPE="1">
    <FIRST_NAME>Adult 3 - Family Room</FIRST_NAME>
    <LAST_NAME>Adult 3</LAST_NAME>
    <AGE />
  </PASSENGER>
  <PASSENGER TYPE="2">
    <FIRST_NAME>Child one</FIRST_NAME>
    <LAST_NAME>Child one</LAST_NAME>
    <AGE>8</AGE>
  </PASSENGER>
  <PASSENGER TYPE="1">
    <FIRST_NAME>Adult 4 - Family Room</FIRST_NAME>
    <LAST_NAME>Adult 2</LAST_NAME>
    <AGE />
  </PASSENGER>
  <PASSENGER TYPE="1">
    <FIRST_NAME>Adult 5 - Family Room</FIRST_NAME>
    <LAST_NAME>Adult 3</LAST_NAME>
    <AGE />
  </PASSENGER>
  <PASSENGER TYPE="2">
    <FIRST_NAME>Child Two</FIRST_NAME>
    <LAST_NAME>Child Two</LAST_NAME>
    <AGE>10</AGE>
  </PASSENGER>
</PASSENGERS>
<OPTIONS>
  <OPTION>
    <OPTION_DATE>01 Dec 2009</OPTION_DATE>
    <OPTION_ID>34176</OPTION_ID>
    <QUANTITY>1</QUANTITY>
    <NO_OF_ADULTS>1</NO_OF_ADULTS>
    <NO_OF_CHILDREN>0</NO_OF_CHILDREN>
    <SELL_PRICE_ID>721252</SELL_PRICE_ID>
    <AVAILABLE_ONLY>true</AVAILABLE_ONLY>
  </OPTION>
  <OPTION>
    <OPTION_DATE>02 Dec 2009</OPTION_DATE>
    <OPTION_ID>34176</OPTION_ID>
    <QUANTITY>1</QUANTITY>

```

```

        <NO_OF_ADULTS>1</NO_OF_ADULTS>
        <NO_OF_CHILDREN>0</NO_OF_CHILDREN>
        <SELL_PRICE_ID>721252</SELL_PRICE_ID>
        <AVAILABLE_ONLY>true</AVAILABLE_ONLY>
    </OPTION>
    <OPTION>
        <OPTION_DATE>01 Dec 2009</OPTION_DATE>
        <OPTION_ID>34177</OPTION_ID>
        <QUANTITY>2</QUANTITY>
        <NO_OF_ADULTS>4</NO_OF_ADULTS>
        <NO_OF_CHILDREN>2</NO_OF_CHILDREN>
        <SELL_PRICE_ID>721253</SELL_PRICE_ID>
        <CHILDREN>
            <AGES>
                <AGE>8</AGE>
                <COUNT>1</COUNT>
            </AGES>
            <AGES>
                <AGE>10</AGE>
                <COUNT>1</COUNT>
            </AGES>
        </CHILDREN>
        <AVAILABLE_ONLY>true</AVAILABLE_ONLY>
    </OPTION>
    <OPTION>
        <OPTION_DATE>02 Dec 2009</OPTION_DATE>
        <OPTION_ID>34177</OPTION_ID>
        <QUANTITY>2</QUANTITY>
        <NO_OF_ADULTS>4</NO_OF_ADULTS>
        <NO_OF_CHILDREN>2</NO_OF_CHILDREN>
        <SELL_PRICE_ID>721253</SELL_PRICE_ID>
        <CHILDREN>
            <AGES>
                <AGE>8</AGE>
                <COUNT>1</COUNT>
            </AGES>
            <AGES>
                <AGE>10</AGE>
                <COUNT>1</COUNT>
            </AGES>
        </CHILDREN>
        <AVAILABLE_ONLY>true</AVAILABLE_ONLY>
    </OPTION>
</OPTIONS>
<NOTES>
    <NOTE> Non-Smoking rooms please</NOTE>
</NOTES>
</BOOKING>
</BOOKING_DETAILS>

```

Booking Response:

This confirms our booking has been received, and provides the JacTravel booking reference number which we will use in any further requests regarding this booking.

If the service was “On Request”, and we had specified AVAILABLE_ONLY = “False” then the status would be “On Request”.

```

<BOOKING_CONFIRMATION>
    <VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="XMLFileName" LICENCE_KEY="LicenceKey"
        TS_API_VERSION="TSAPIVersion">

```

```

    <XML_VERSION_NO>XMLVersionNumber</XML_VERSION_NO>
  </VERSION_HISTORY>
  <BOOKING_NAME>Test booking for 2xTwin, 2xFamily Room1</BOOKING_NAME>
  <BOOKING_REFERENCE>JCJA2049854</BOOKING_REFERENCE>
  <STATUS>Confirmed</STATUS>
</BOOKING_CONFIRMATION>

```

8.2.4 Booking Details request

Here we can verify details of the booking. This is often used to get updated details regarding an “On Request” booking.

Booking Details Request:

```

<BOOKING_DETAILS>
  <VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="XMLFileName" LICENCE_KEY="FA39F277-944B-971C-BC64-79E80DEE81C4" TS_API_VERSION="TSAPIVersion">
    <XML_VERSION_NO>3.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <BOOKING>
    <BOOKING_REFERENCE_NO>JCJA2049854</BOOKING_REFERENCE_NO>
  </BOOKING>
</BOOKING_DETAILS>

```

Booking Details Response:

Note that the split rate for the double room results in two separate BOOKED_OPTIONS with sequential SPLITIDs, one for each price rate.

The BOOKED_OPTION_STATUS>From Allocation</BOOKED_OPTION_STATUS> tag shows us that the room is available “From Allocation”. If this booking had been “On Request” this tag would initially be “On Request”, and would later be dynamically updated as the reservation team confirm or reject the booking.

```

<BOOKING_DETAILS>
  <VERSION_HISTORY APPLICATION_NAME="TS HOTELS API 1.0" XML_FILE_NAME="BookingDetailsResponse.xml" LICENCE_KEY="FA39F277-944B-971C-BC64-79E80DEE81C4" TS_API_VERSION="v309.0.0">
    <XML_VERSION_NO>1.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <BOOKING>
    <BOOKING_REFERENCE_NO>JCJA2063124</BOOKING_REFERENCE_NO>
    <CLIENT_NAME>Client Name</CLIENT_NAME>
    <BOOKING_NAME>Test booking for 1xSingle, 2xFamily Room1</BOOKING_NAME>
    <BOOKING_DATE>20 Jun 2009</BOOKING_DATE>
    <BOOKING_CURRENCY>EUR</BOOKING_CURRENCY>
    <BOOKING_START_DATE>01 Dec 2009</BOOKING_START_DATE>
    <BOOKING_END_DATE>03 Dec 2009</BOOKING_END_DATE>
    <NUMBER_OF_NIGHTS>2</NUMBER_OF_NIGHTS>
    <TOTAL_ADULTS>3</TOTAL_ADULTS>
    <TOTAL_CHILDREN>2</TOTAL_CHILDREN>
    <BOOKING_STATUS>Confirmed</BOOKING_STATUS>
    <TOTAL_BOOKING_PRICE>394.44</TOTAL_BOOKING_PRICE>
    <PAX_OCCUPANCYS>
      <PAX_OCCUPANCY TYPE="1">
        <OCCUPANCYID>1</OCCUPANCYID>
        <NO_OF_PAX>1</NO_OF_PAX>
      </PAX_OCCUPANCY>
    </PAX_OCCUPANCYS>
  </BOOKING>
</BOOKING_DETAILS>

```



```

</PAX_OCCUPANCY>
<PAX_OCCUPANCY TYPE="1">
  <OCCUPANCYID>7</OCCUPANCYID>
  <NO_OF_PAX>2</NO_OF_PAX>
</PAX_OCCUPANCY>
<PAX_OCCUPANCY TYPE="2">
  <OCCUPANCYID>7</OCCUPANCYID>
  <NO_OF_PAX>2</NO_OF_PAX>
</PAX_OCCUPANCY>
</PAX_OCCUPANCYS>
<PASSENGERS>
  <PASSENGER TYPE="1">
    <PASSENGERID>1339728</PASSENGERID>
    <FIRST_NAME>Adult 1 - Single Room</FIRST_NAME>
    <LAST_NAME>Adult 1</LAST_NAME>
  </PASSENGER>
  <PASSENGER TYPE="1">
    <PASSENGERID>1339729</PASSENGERID>
    <FIRST_NAME>Adult 2 - Family Room</FIRST_NAME>
    <LAST_NAME>Adult 2</LAST_NAME>
  </PASSENGER>
  <PASSENGER TYPE="1">
    <PASSENGERID>1339730</PASSENGERID>
    <FIRST_NAME>Adult 3 - Family Room</FIRST_NAME>
    <LAST_NAME>Adult 3</LAST_NAME>
  </PASSENGER>
  <PASSENGER TYPE="2">
    <PASSENGERID>1339731</PASSENGERID>
    <FIRST_NAME>Child one</FIRST_NAME>
    <LAST_NAME>Child one</LAST_NAME>
  </PASSENGER>
  <PASSENGER TYPE="2">
    <PASSENGERID>1339732</PASSENGERID>
    <FIRST_NAME />
    <LAST_NAME>Child 2</LAST_NAME>
  </PASSENGER>
</PASSENGERS>
<SERVICES>
  <BOOKED_SERVICE BOOKEDSERVICEID="681995" SERVICEID="12046" CANCELLATION_POLICY="">
    <OPTIONS>
      <BOOKED_OPTION SPLITID="0" ISEXTRA="0">
        <BOOKED_OPTION_ID>630232</BOOKED_OPTION_ID>
        <BOOKED_OPTION_CURRENCY>EUR</BOOKED_OPTION_CURRENCY>
        <BOOKED_OPTION_STATUS>From Allocation</BOOKED_OPTION_STATUS>
        <BOOKED_OPTION_TOTAL_AMOUNT>79.8</BOOKED_OPTION_TOTAL_AMOUNT>
        <MEAL_PLAN>Bed & Continental Breakfast</MEAL_PLAN>
        <OPTION_ID>34176</OPTION_ID>
        <OPTION_NAME>Single Room</OPTION_NAME>
        <BOOKED_OPTION_IN_DATE>01 Dec 2009</BOOKED_OPTION_IN_DATE>
        <BOOKED_OPTION_OUT_DATE>03 Dec 2009</BOOKED_OPTION_OUT_DATE>
        <NO_OF_NIGHTS>2</NO_OF_NIGHTS>
        <NO_OF_ADULTS>1</NO_OF_ADULTS>
        <QUANTITY>1</QUANTITY>
        <NO_OF_CHILDREN>0</NO_OF_CHILDREN>
      </BOOKED_OPTION>
      <BOOKED_OPTION SPLITID="0" ISEXTRA="0">
        <BOOKED_OPTION_ID>630233</BOOKED_OPTION_ID>
        <BOOKED_OPTION_CURRENCY>EUR</BOOKED_OPTION_CURRENCY>
        <BOOKED_OPTION_STATUS>From Allocation</BOOKED_OPTION_STATUS>
        <BOOKED_OPTION_TOTAL_AMOUNT>314.64</BOOKED_OPTION_TOTAL_AMOUNT>
        <MEAL_PLAN>Bed & Continental Breakfast</MEAL_PLAN>
        <OPTION_ID>34177</OPTION_ID>
        <OPTION_NAME>Double/Twin + 1 Child</OPTION_NAME>
        <BOOKED_OPTION_IN_DATE>01 Dec 2009</BOOKED_OPTION_IN_DATE>
        <BOOKED_OPTION_OUT_DATE>03 Dec 2009</BOOKED_OPTION_OUT_DATE>
        <NO_OF_NIGHTS>2</NO_OF_NIGHTS>

```

```

        <NO_OF_ADULTS>4</NO_OF_ADULTS>
        <QUANTITY>2</QUANTITY>
        <NO_OF_CHILDREN>2</NO_OF_CHILDREN>
        <CHILDREN>
            <AGES>
                <AGE>8</AGE>
                <COUNT>1</COUNT>
            </AGES>
            <AGES>
                <AGE>10</AGE>
                <COUNT>1</COUNT>
            </AGES>
        </CHILDREN>
    </BOOKED_OPTION>
</OPTIONS>
</BOOKED_SERVICE>
</SERVICES>
</BOOKING>
</BOOKING_DETAILS>

```

8.2.5 Cancellation Policy Request and Response

Now a request is made for the cancellation policy that applies to this booking. The cancellation policy request is made on 15th Nov 2009.

Cancellation policy request:

```

<CANCELLATION_POLICY_DETAILS_REQUEST>
  <VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="XMLFileName" LICENCE_KEY="FA39F277-944B-971C-BC64-79E80DEE81C4" TS_API_VERSION="TSAPIVersion">
    <XML_VERSION_NO />
  </VERSION_HISTORY>
  <BODY>
    <BOOKING_CHARGE>
      <BOOKING_REFERENCE>JCJA2049854</BOOKING_REFERENCE>
    </BOOKING_CHARGE>
  </BODY>
</CANCELLATION_POLICY_DETAILS_REQUEST>

```

Cancellation policy response:

```

<CANCELLATION_POLICY_DETAILS_RESPONSE>
  <VERSION_HISTORY APPLICATION_NAME="TS HOTELS API 1.0" XML_FILE_NAME="CancelBookingResponse.XML" LICENCE_KEY="FA39F233-944B-471C-BC64-76E80DEE71C1" TS_API_VERSION="v309.0.0">
    <XML_VERSION_NO>1.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <BODY>
    <SERVICE ID="12046">
      <CHARGE_DESCRIPTION>2 day(s) prior to arrival</CHARGE_DESCRIPTION>
      <CHARGE_TYPE>Percentage of first night</CHARGE_TYPE>
      <METHOD>
        <DAYS_BEFORE_CHECK_IN>2</DAYS_BEFORE_CHECK_IN>
        <CHARGE_VALUE>100.00</CHARGE_VALUE>
      </METHOD>
      <CANCELLATION_POLICY_BREAK_DOWN>
        <CURRENCY>EUR</CURRENCY>
        <DAYS>
          <CHARGE_DATE>Dec 1 2009 11:59AM</CHARGE_DATE>
          <CHARGE>394.44</CHARGE>
        </DAYS>
      </CANCELLATION_POLICY_BREAK_DOWN>
    </SERVICE ID="12046">
  </BODY>
</CANCELLATION_POLICY_DETAILS_RESPONSE>

```

```

        <DAYS>
        <CHARGE_DATE>Nov 30 2009 12:00PM</CHARGE_DATE>
        <CHARGE>394.44</CHARGE>
    </DAYS>
</CANCELLATION_POLICY_BREAK_DOWN>
</SERVICE>
</BODY>
</CANCELLATION_POLICY_DETAILS_RESPONSE>

```

8.2.6 Cancellation Request and Response

Cancellation Request:

```

<BOOKING_CANCELLATION>
  <VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="XMLFileName" LICENCE_KEY="FA39F277-944B-971C-BC64-79E80DEE81C4" TS_API_VERSION="TSAPIVersion">
    <XML_VERSION_NO>3.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <BOOKING_REFERENCE>JCJA2049854</BOOKING_REFERENCE>
</BOOKING_CANCELLATION>

```

Cancellation Response:

Note again that the cancellation request is made on 15th Nov 2009, well outside of the cancellation policy so no charges are applied to the cancellation.

```

<BOOKING_CANCELLATION_RESPONSE>
  <VERSION_HISTORY APPLICATION_NAME="TS HOTELS API 1.0" XML_FILE_NAME="CancelBookingResponse.XML" LICENCE_KEY="FA39F233-944B-471C-BC64-76E80DEE71C1" TS_API_VERSION="v309.0.0">
    <XML_VERSION_NO>1.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <BODY>
    <BOOKING_REFERENCE>JCJA2049854</BOOKING_REFERENCE>
    <STATUS>Cancelled</STATUS>
    <CANCELLATIONCHARGES>
      <CURRENCY>EUR</CURRENCY>
      <CHARGE>0.00</CHARGE>
    </CANCELLATIONCHARGES>
  </BODY>
</BOOKING_CANCELLATION_RESPONSE>

```

Additional example of cancellation with charges:

```

<BOOKING_CANCELLATION_RESPONSE>
  <VERSION_HISTORY APPLICATION_NAME="TS HOTELS API 1.0" XML_FILE_NAME="CancelBookingResponse.XML" LICENCE_KEY="FA39F233-944B-471C-BC64-76E80DEE71C1" TS_API_VERSION="v309.0.0">
    <XML_VERSION_NO>1.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <BODY>
    <BOOKING_REFERENCE>JCJA2049854</BOOKING_REFERENCE>
    <STATUS>Late Cancellation</STATUS>
    <CANCELLATIONCHARGES>
      <CURRENCY>EUR</CURRENCY>
      <CHARGE>394.44</CHARGE>
    </CANCELLATIONCHARGES>
  </BODY>
</BOOKING_CANCELLATION_RESPONSE>

```

8.3 Live Static Data Process

The following request/response examples show the process of searching for and obtaining services static data information. This example simulates the obtaining of updated static data files and obtaining that information.

8.3.1 Hotel List Request

Here a request is sent tasking for all changes between the 17th and 20th August

Hotel List Request:

```
<HOTEL_LIST_REQUEST>
  <VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="XMLFileName" LICENCE_KEY="
    FA39F277-944B-971C-BC64-79E80DEE81C4" TS_API_VERSION="TSAPIVersion">
    <XML_VERSION_NO>3.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <BODY>
    <STARTDATE>17 Aug 09</STARTDATE>
    <ENDDATE> 20 Aug 09</ENDDATE>
  </BODY>
</HOTEL_LIST_REQUEST>
```

Hotel List Response:

```
<HOTEL_LIST_RESPONSE>
  <VERSION_HISTORY APPLICATION_NAME="TS HOTELS API 1.0" XML_FILE_NAME="Response.XML"
    LICENCE_KEY="LicKey" TS_API_VERSION="v309.0.0">
    <XML_VERSION_NO>1.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <HOTEL ID="10452" NAME="ZZZ Test Hotel" LASTUPDATEDDATE="2009-08-17T05:41:36.953" ISDELETED="0"
    />
  <HOTEL ID="85461" NAME="Test Bristol" LASTUPDATEDDATE="2009-08-17T15:16:31.093" ISDELETED="0" />
  <HOTEL ID="12079" NAME="Test Stellite Hotel" LASTUPDATEDDATE="2009-08-17T11:39:12.407"
    ISDELETED="0" />
  <HOTEL ID="12029" NAME="Test Hamilton" LASTUPDATEDDATE="2009-08-17T16:01:57.670" ISDELETED="1" />
</HOTEL_LIST_RESPONSE>
```

Note service 12029 – Test Hamilton contains " ISDELETED="1", indicating the service is not longer available for booking.

8.3.2 Hotel Details

Service 10452 has been modified so we obtain those details

Hotel Details Request:

```
<HOTEL_DETAILS_REQUEST>
  <VERSION_HISTORY APPLICATION_NAME="AppName" XML_FILE_NAME="XMLFileName"
    LICENCE_KEY="FA39F277-944B-971C-BC64-79E80DEE81C4" TS_API_VERSION="TSAPIVersion">
    <XML_VERSION_NO>3.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <BODY>
    <SERVICEID>10452</SERVICEID>
  </BODY>
</HOTEL_DETAILS_REQUEST>
```

Hotel Details Response:

```
<HOTEL_DETAILS_RESPONSE>
  <VERSION_HISTORY APPLICATION_NAME="TS HOTELS API 1.0" XML_FILE_NAME="Response.XML"
    LICENCE_KEY="LicKey" TS_API_VERSION="v309.0.0">
    <XML_VERSION_NO>1.0</XML_VERSION_NO>
  </VERSION_HISTORY>
  <HOTELDETAILS>
    <SERVICEID>10452</SERVICEID>
    <LONGNAME>ZZZ Test Hotel</LONGNAME>
    <DESCRIPTIONS>
      <DESCRIPTION LANGUAGE="English">This is a test hotel - description in English</DESCRIPTION>
      <DESCRIPTION LANGUAGE="French">French hotel description</DESCRIPTION>
      <DESCRIPTION LANGUAGE="German">German hotel description</DESCRIPTION>
      <DESCRIPTION LANGUAGE="Spanish">Este Hotel es solo para clientes API, solo para pruebas. No se
        pueden hacer ningún booking genuino en este hotel . JacTravel no será hará responsable de
        ninguna reserva hecha en este hotel.</DESCRIPTION>
    </DESCRIPTIONS>
    <TERMS_INCLUSION />
    <TERMS_EXCLUSION />
    <PRICE_START_RANGE>1.0000</PRICE_START_RANGE>
    <PRICE_END_RANGE>95.2000</PRICE_END_RANGE>
    <CURRENCY_CODE>GBP</CURRENCY_CODE>
    <FACILITIES>2, 5, 7, 48, 51, 86, 196, 231</FACILITIES>
    <STAR_RATING>3 * +</STAR_RATING>
    <LOCATION>Gt.Portland St.,Marylebone,London Zones 1 & 2,London,Greater London,London
      Region,England,Europe</LOCATION>
    <IMAGES>
      <CATEGORY>
        <CATEGORYNAME>Exterior View</CATEGORYNAME>
        <IMAGE>/10452_1_1.jpg</IMAGE>
        <THUMBNAIL>/10452_1_1_tn.jpg</THUMBNAIL>
      </CATEGORY>
    </IMAGES>
    <ADDRESS>
      <LINE1>Nowhere</LINE1>
      <LINE2 />
      <LINE3 />
      <LINE4 />
      <POSTCODE>none</POSTCODE>
```

```

<CITY>Sunderland</CITY>
<COUNTRY>England</COUNTRY>
<TELEPHONENUMBER>12345</TELEPHONENUMBER>
<FAXNUMBER>12345</FAXNUMBER>
<EMERGENCYPHONENUMBER />
<EMAILADDRESS>itservices@jactravel.co.uk</EMAILADDRESS>
<WEBSITEURL />
<STATE />
</ADDRESS>
<SERVICE_ADDINFO>
  <PRIMARYAIRPORT>Abu Dhabi International</PRIMARYAIRPORT>
  <YEAROFOPENING>2004</YEAROFOPENING>
  <NEARESTAIRPORT>Heathrow</NEARESTAIRPORT>
  <COMPASSDIRECTION>W</COMPASSDIRECTION>
  <SERVICEDISTANCE>12</SERVICEDISTANCE>
  <SERVICELONGITUDE />
  <SERVICELATITUDE />
  <SERVICECHECKINTIME>12:00:00</SERVICECHECKINTIME>
  <SERVICECHECKOUTTIME>12:00:00</SERVICECHECKOUTTIME>
</SERVICE_ADDINFO>
<SERVICE_CREDITCARDS_ACCEPTED>
  <CREDITCARD_ACCEPTED>American Express</CREDITCARD_ACCEPTED>
  <CREDITCARD_ACCEPTED>Visa</CREDITCARD_ACCEPTED>
</SERVICE_CREDITCARDS_ACCEPTED>
<RATINGS>
  <RATINGTYPE RATINGTYPENAME="JAC Rating">
    <RATING>3 * +</RATING>
  </RATINGTYPE>
  <RATINGTYPE RATINGTYPENAME="Hotel Chain">
    <RATING>Oxford Hotels</RATING>
  </RATINGTYPE>
  <RATINGTYPE RATINGTYPENAME="Property Type">
    <RATING>Castle</RATING>
  </RATINGTYPE>
  <RATINGTYPE RATINGTYPENAME="Property Location">
    <RATING>Countryside</RATING>
  </RATINGTYPE>
</RATINGS>
<OPTIONS>
  <OPTION>
    <OPTION_ID>17650</OPTION_ID>
    <OCCUPANCY_TYPE_ID>3</OCCUPANCY_TYPE_ID>
    <OCCUPANCY_TYPE_NAME>Double Room</OCCUPANCY_TYPE_NAME>
    <OPTION_NAME LANGNAME="Spanish">Habitación Doble</OPTION_NAME>
    <OPTION_NAME LANGNAME="German">Doppelzimmer</OPTION_NAME>
    <CHILD_POLICY_NAME LANGNAME="French">0 – 12= 50%</CHILD_POLICY_NAME>
    <CHILD_POLICY_NAME LANGNAME="German">0-12=50%</CHILD_POLICY_NAME>
    <CHILD_POLICY_NAME LANGNAME="Spanish">0-12=50%</CHILD_POLICY_NAME>
    <CHILD_RATES>
      <RATE AGESTART="0" AGEEND="12" RATEVALUE="50.0000" />
      <RATE AGESTART="13" AGEEND="18" RATEVALUE="100.0000" />
    </CHILD_RATES>
  </OPTION>
  <OPTION>
    <OPTION_ID>17651</OPTION_ID>
    <OCCUPANCY_TYPE_ID>2</OCCUPANCY_TYPE_ID>
    <OCCUPANCY_TYPE_NAME>Twin Room</OCCUPANCY_TYPE_NAME>
    <OPTION_NAME LANGNAME="German">Zweibettzimmer</OPTION_NAME>
    <OPTION_NAME LANGNAME="Spanish">Habitación Twin</OPTION_NAME>
    <CHILD_POLICY_NAME LANGNAME="Spanish">0-12=50%</CHILD_POLICY_NAME>
    <CHILD_POLICY_NAME LANGNAME="German">0-12=50%</CHILD_POLICY_NAME>
    <CHILD_POLICY_NAME LANGNAME="French">0 – 12= 50%</CHILD_POLICY_NAME>
    <CHILD_RATES>
      <RATE AGESTART="0" AGEEND="12" RATEVALUE="50.0000" />
      <RATE AGESTART="13" AGEEND="18" RATEVALUE="100.0000" />
    </CHILD_RATES>
  </OPTION>
</OPTIONS>

```

```

</OPTION>
<OPTION>
  <OPTION_ID>17652</OPTION_ID>
  <OCCUPANCY_TYPE_ID>1</OCCUPANCY_TYPE_ID>
  <OCCUPANCY_TYPE_NAME>Single Room</OCCUPANCY_TYPE_NAME>
  <OPTION_NAME LANGNAME="German">Einzelzimmer</OPTION_NAME>
  <OPTION_NAME LANGNAME="Spanish">Habitación Individual</OPTION_NAME>
  <CHILD_POLICY_NAME LANGNAME="French">0 – 12= 50%</CHILD_POLICY_NAME>
  <CHILD_POLICY_NAME LANGNAME="German">0-12=50%</CHILD_POLICY_NAME>
  <CHILD_POLICY_NAME LANGNAME="Spanish">0-12=50%</CHILD_POLICY_NAME>
  <CHILD_RATES>
    <RATE AGESTART="0" AGEEND="12" RATEVALUE="50.0000" />
    <RATE AGESTART="13" AGEEND="18" RATEVALUE="100.0000" />
  </CHILD_RATES>
</OPTION>
<OPTION>
  <OPTION_ID>17653</OPTION_ID>
  <OCCUPANCY_TYPE_ID>4</OCCUPANCY_TYPE_ID>
  <OCCUPANCY_TYPE_NAME>Triple Room</OCCUPANCY_TYPE_NAME>
  <OPTION_NAME LANGNAME="German">Dreibettzimmer</OPTION_NAME>
  <OPTION_NAME LANGNAME="Spanish">Habitación Triple</OPTION_NAME>
  <CHILD_POLICY_NAME LANGNAME="Spanish">0-12=50%</CHILD_POLICY_NAME>
  <CHILD_POLICY_NAME LANGNAME="German">0-12=50%</CHILD_POLICY_NAME>
  <CHILD_POLICY_NAME LANGNAME="French">0 – 12= 50%</CHILD_POLICY_NAME>
  <CHILD_RATES>
    <RATE AGESTART="0" AGEEND="12" RATEVALUE="50.0000" />
    <RATE AGESTART="13" AGEEND="18" RATEVALUE="100.0000" />
  </CHILD_RATES>
</OPTION>
<OPTION>
  <OPTION_ID>17697</OPTION_ID>
  <OCCUPANCY_TYPE_ID>7</OCCUPANCY_TYPE_ID>
  <OCCUPANCY_TYPE_NAME>Double/Twin + 1 Child</OCCUPANCY_TYPE_NAME>
  <OPTION_NAME LANGNAME="Spanish">Doble/Twin + 1 niño</OPTION_NAME>
  <OPTION_NAME LANGNAME="German">Doppel-/Zweibettzimmer + 1 Kind</OPTION_NAME>
  <CHILD_POLICY_NAME LANGNAME="French">0 – 12= 50%</CHILD_POLICY_NAME>
  <CHILD_POLICY_NAME LANGNAME="German">0-12=50%</CHILD_POLICY_NAME>
  <CHILD_POLICY_NAME LANGNAME="Spanish">0-12=50%</CHILD_POLICY_NAME>
  <CHILD_RATES>
    <RATE AGESTART="0" AGEEND="12" RATEVALUE="50.0000" />
    <RATE AGESTART="13" AGEEND="18" RATEVALUE="100.0000" />
  </CHILD_RATES>
</OPTION>
<OPTION>
  <OPTION_ID>17698</OPTION_ID>
  <OCCUPANCY_TYPE_ID>8</OCCUPANCY_TYPE_ID>
  <OCCUPANCY_TYPE_NAME>Double/Twin + 2 Children</OCCUPANCY_TYPE_NAME>
  <OPTION_NAME LANGNAME="Spanish">Doble/Twin + 2 niños</OPTION_NAME>
  <OPTION_NAME LANGNAME="German">Doppel-/Zweibettzimmer + 2
    Kinder</OPTION_NAME>
  <CHILD_POLICY_NAME LANGNAME="Spanish">0-12=50%</CHILD_POLICY_NAME>
  <CHILD_POLICY_NAME LANGNAME="German">0-12=50%</CHILD_POLICY_NAME>
  <CHILD_POLICY_NAME LANGNAME="French">0 – 12= 50%</CHILD_POLICY_NAME>
  <CHILD_RATES>
    <RATE AGESTART="0" AGEEND="12" RATEVALUE="50.0000" />
    <RATE AGESTART="13" AGEEND="18" RATEVALUE="100.0000" />
  </CHILD_RATES>
</OPTION>
</OPTIONS>
</HOTELDETAILS>
</HOTEL_DETAILS_RESPONSE>

```

8.4 Room Occupancies

7 Occupancy types are supplied for various room type searches. Each type enforces Min/Max passenger rules.

Room Type	Occupancy ID	Max/ Min Adult	Max/Min Child
Single	1	1 / 1	0 / 0
Twin	2	2 / 2	0 / 0
Double	3	2 / 2	0 / 0
Triple	4	3 / 3	0 / 0
Quad	5	4 / 4	0 / 0
Family Room 1	7	2 / 2	1 / 1
Family Room 2	8	2 / 2	2 / 2

Each room type can contain several “styles” of room under a one occupancy id category, with each identified though their unique OPTIONID, for example below shows an abbreviated response for 1 single room search. All rooms returned are OccupancyID 1. It’s important to obtain the correct OPTIONID required for booking as each unique OPIONID will have is own associated costs.

```
<OPTIONS>
<OPTION>
  <OPTIONID>34169</OPTIONID>
  <OPTION_NAME>Single Room</OPTION_NAME>
  .....

</OPTION>
<OPTION>
  <OPTIONID>34242</OPTIONID>
  <OPTION_NAME>Double for Single Occupancy</OPTION_NAME>
  .....

</OPTION>
<OPTION>
  <OPTIONID>34248</OPTIONID>
  <OPTION_NAME>Double for Single Occupancy (Without Breakfast)</OPTION_NAME>
  .....

</OPTION>
</OPTIONS>
```