

**OMNIBEES Pull   
Interface Specification**

**Version 2.6**

**Setember 2017**

Contents

[Index of figures 6](#_Toc493758895)

[Index of tables 7](#_Toc493758896)

[1 Revision 8](#_Toc493758897)

[1.1 Version notes 12](#_Toc493758898)

[1.1.1 Version 2.6 12](#_Toc493758899)

[1.1.2 Version 2.5 12](#_Toc493758900)

[1.1.3 Version 2.4 13](#_Toc493758901)

[1.1.4 Version 2.3 13](#_Toc493758902)

[1.1.5 Version 2.2 13](#_Toc493758903)

[1.1.6 Version 2.1 14](#_Toc493758904)

[1.1.7 Version 2.0.1 14](#_Toc493758905)

[1.1.8 Version 2.0 15](#_Toc493758906)

[1.1.9 Version 1.9 16](#_Toc493758907)

[1.1.10 Version 1.8 17](#_Toc493758908)

[2 Documentation Overview 19](#_Toc493758909)

[3 Protocols 22](#_Toc493758910)

[3.1 Pull Service 22](#_Toc493758911)

[3.2 Notify System 22](#_Toc493758912)

[4 Connectivity Endpoints 23](#_Toc493758913)

[4.1 Endpoint URL for testing environment 23](#_Toc493758914)

[4.2 Endpoint URL for dev/cert Environment 23](#_Toc493758915)

[4.2.1 SOAP Xml messages 23](#_Toc493758916)

[4.2.2 REST Json messages 23](#_Toc493758917)

[4.3 Endpoint URL for Production/Live environment 24](#_Toc493758918)

[4.3.1 SOAP Xml messages 24](#_Toc493758919)

[4.3.2 REST Json messages 24](#_Toc493758920)

[4.4 Endpoint URL for Production/Live environment for cache systems 24](#_Toc493758921)

[4.4.1 SOAP Xml messages 24](#_Toc493758922)

[4.4.2 REST Json messages 24](#_Toc493758923)

[4.5 Method 24](#_Toc493758924)

[5 System information 25](#_Toc493758925)

[5.1 Login 25](#_Toc493758926)

[5.2 Message 25](#_Toc493758927)

[5.2.1 Communication Success 25](#_Toc493758928)

[5.2.2 Communication Success but fail on business rules 25](#_Toc493758929)

[5.2.3 Communication Fail 26](#_Toc493758930)

[5.3 Login Type 27](#_Toc493758931)

[5.4 Operator 27](#_Toc493758932)

[5.4.1 Login Object 27](#_Toc493758933)

[5.4.2 Payment Types 27](#_Toc493758934)

[5.5 Self-Booking Tool 28](#_Toc493758935)

[5.5.1 Login 28](#_Toc493758936)

[5.5.2 Paymet Types 28](#_Toc493758937)

[5.5.3 Agency 29](#_Toc493758938)

[5.5.4 Agency Office 29](#_Toc493758939)

[5.5.5 Corporate 30](#_Toc493758940)

[5.6 Booking Engine 31](#_Toc493758941)

[5.6.1 Login Object 31](#_Toc493758942)

[5.6.2 Payment Types 31](#_Toc493758943)

[5.7 Hotel Avail– Best Practices 32](#_Toc493758944)

[5.7.1 Filter 32](#_Toc493758945)

[5.7.2 Best Only rates 33](#_Toc493758946)

[5.7.3 Paging 33](#_Toc493758947)

[5.8 Notify Reservations 34](#_Toc493758948)

[5.8.1 Reservation Status 34](#_Toc493758949)

[5.9 Images 35](#_Toc493758950)

[5.9.1 OTA\_HotelDescriptiveInfo 35](#_Toc493758951)

[5.9.2 OTA\_HotelRatePlan 38](#_Toc493758952)

[5.9.3 OTA\_HotelAvail 42](#_Toc493758953)

[6 Business Rules 48](#_Toc493758954)

[6.1 Deposit Policies 48](#_Toc493758955)

[6.1.1 For OTA\_HotelAvailRS 48](#_Toc493758956)

[6.1.2 For OTA\_HotelResRQ 50](#_Toc493758957)

[6.2 Cancelation Policies 51](#_Toc493758958)

[6.2.1 Class Diagram 52](#_Toc493758959)

[6.2.2 Examples 52](#_Toc493758960)

[6.3 Tax Policies 56](#_Toc493758961)

[6.3.1 Class Diagram 56](#_Toc493758962)

[6.3.2 Examples 57](#_Toc493758963)

[6.4 Meal Plan Included 58](#_Toc493758964)

[6.4.1 Class Diagram 58](#_Toc493758965)

[6.4.2 Example 58](#_Toc493758966)

[6.5 Offers 59](#_Toc493758967)

[6.5.1 Class Diagram 60](#_Toc493758968)

[6.5.2 Examples 61](#_Toc493758969)

[6.6 Included Extras 67](#_Toc493758970)

[6.6.1 Class Diagram 67](#_Toc493758971)

[6.6.2 Example 67](#_Toc493758972)

[6.7 Additional Extras 68](#_Toc493758973)

[6.7.1 Class Diagram 68](#_Toc493758974)

[6.7.2 Example 69](#_Toc493758975)

[6.8 Payment Policies 71](#_Toc493758976)

[6.8.1 For OTA\_HotelAvailRS (Object used to search availability and prices) 71](#_Toc493758977)

[6.8.2 For OTA\_HotelResRQ 72](#_Toc493758978)

[6.8.3 Codes that can be returned on search 73](#_Toc493758979)

[6.9 Reservation Status 73](#_Toc493758980)

[6.10 Booking on Request 74](#_Toc493758981)

[6.10.1 Search (Hotel Avail) including “On Request” rates 75](#_Toc493758982)

[6.10.2 HotelRes Status 76](#_Toc493758983)

[6.10.3 Reservation that is on status RequestAccepted 77](#_Toc493758984)

[6.10.4 Accept reservation 77](#_Toc493758985)

[6.10.5 Ignore Reservation 78](#_Toc493758986)

[6.10.6 Reservation that is on status RequestDenied 78](#_Toc493758987)

[6.11 Rates that are not available for sell 79](#_Toc493758988)

[7 Messages 81](#_Toc493758989)

[7.1 Hotel Search 81](#_Toc493758990)

[7.1.1 Description 81](#_Toc493758991)

[7.1.2 OTA\_HotelSearchRQ 81](#_Toc493758992)

[7.1.3 OTA\_HotelSearchRS 88](#_Toc493758993)

[7.2 Availability and prices Service 94](#_Toc493758994)

[7.2.1 Description 94](#_Toc493758995)

[7.2.2 OTA\_HotelAvailRQ 95](#_Toc493758996)

[7.2.3 OTA\_HotelAvailRS 114](#_Toc493758997)

[7.3 Booking Service 136](#_Toc493758998)

[7.3.1 SendHotelRes 136](#_Toc493758999)

[7.3.2 SendHotelResModify 157](#_Toc493759000)

[7.3.3 SendHotelResCancel 173](#_Toc493759001)

[7.3.4 Get Bookings 185](#_Toc493759002)

[7.3.5 HotelResModifyNotif 186](#_Toc493759003)

[7.4 Hotel Descriptive Info 196](#_Toc493759004)

[7.4.1 OTA\_HotelDescriptiveInfoRQ 196](#_Toc493759005)

[7.4.2 OTA\_HotelDescriptiveInfoRS 199](#_Toc493759006)

[7.5 Rate Plan Request 211](#_Toc493759007)

[7.5.1 OTA\_HotelRatePlanRQ 211](#_Toc493759008)

[7.5.2 OTA\_HotelRatePlanRS 214](#_Toc493759009)

[7.6 GetAgencies 215](#_Toc493759010)

[7.6.1 Description 215](#_Toc493759011)

[7.6.2 Deprecated objects 215](#_Toc493759012)

[7.6.3 Class Diagram 215](#_Toc493759013)

[7.6.4 Get all agencies 215](#_Toc493759014)

[7.6.5 Get agency by PCC code 215](#_Toc493759015)

[7.7 GetCorporates 216](#_Toc493759016)

[7.7.1 Description 216](#_Toc493759017)

[7.7.2 Get all Corporates 216](#_Toc493759018)

[7.7.3 Get Corporate by corporate code 216](#_Toc493759019)

[7.8 GetCountryList 216](#_Toc493759020)

[7.8.1 Description 216](#_Toc493759021)

[7.9 GetStateList 217](#_Toc493759022)

[7.9.1 Description 217](#_Toc493759023)

[7.10 GetCityList 217](#_Toc493759024)

[7.10.1 Description 217](#_Toc493759025)

[7.11 GetZoneList 218](#_Toc493759026)

[7.11.1 Description 218](#_Toc493759027)

[8 Attachments 219](#_Toc493759028)

[8.1 Examples 219](#_Toc493759029)

[8.2 Suported Codes 219](#_Toc493759030)

[9 References 220](#_Toc493759031)

# Index of figures

[Figure 1 - Main messages 19](#_Toc493758870)

[Figure 2 - Pagging according to client website 33](#_Toc493758871)

[Figure 3 - Deposit Policies Class Diagram 48](#_Toc493758872)

[Figure 4 - Cancelation Policy class diagram 52](#_Toc493758873)

[Figure 5 - Tax Policies class Diagram 56](#_Toc493758874)

[Figure 6 - Meal plan class diagram 58](#_Toc493758875)

[Figure 7 - Offers class diagram 60](#_Toc493758876)

[Figure 8 - Rate Plan inclusions class diagram 67](#_Toc493758877)

[Figure 9 - Additional Extras class diagram 68](#_Toc493758878)

[Figure 10 - Payment methods before version 1.8 class diagram 71](#_Toc493758879)

[Figure 11 - Payment methods after version 1.8 class diagram 71](#_Toc493758880)

[Figure 12 - OTA\_HotelSearchRQ class diagram 84](#_Toc493758881)

[Figure 13 - Transaction - Initiate-> Modify (Optional) -> End with Commit or Ignore 161](#_Toc493758882)

[Figure 14 - Initiate -> Time out on initiate 162](#_Toc493758883)

[Figure 15 - Initiate -> Modify -> Time out on Modify 163](#_Toc493758884)

[Figure 16 - Initiate -> Commit -> Time out on commit 164](#_Toc493758885)

[Figure 17 - Initiate -> Ignore -> Time out on ignore 165](#_Toc493758886)

[Figure 18 - Initiate -> Initiate (RetransmissionIndicator = false) 166](#_Toc493758887)

[Figure 19 - Initiate -> Commit -> Commit (RetransmissionIndicator = false) 167](#_Toc493758888)

[Figure 20 - Initiate -> Ignore -> Ignore (RetransmissionIndicator = false) 168](#_Toc493758889)

[Figure 21 - Commit (Without previous initiate) 168](#_Toc493758890)

[Figure 22 - OTA\_CancelRQ class diagram 178](#_Toc493758891)

[Figure 23 - OTA\_CancelRS class diagram 182](#_Toc493758892)

[Figure 24 - OTA\_HotelResModifyNotifRQ class diagram 189](#_Toc493758893)

[Figure 25 - OTA\_HotelResModifyNotifRS class diagram 193](#_Toc493758894)

# Index of tables

[Table 1 - Reservation Status 34](#_Toc493758823)

[Table 2 - Payment types 73](#_Toc493758824)

[Table 3 - Reservation Status 73](#_Toc493758825)

[Table 4 - Availability status for not available rates 80](#_Toc493758826)

[Table 5 - OTA\_HotelSearchRQ Schema 82](#_Toc493758827)

[Table 6 - OTA\_HotelSearchRS schema 89](#_Toc493758828)

[Table 7 - OTA\_HotelAvailRQ Schema 96](#_Toc493758829)

[Table 8 - OTA\_HotelAvailRS schema 121](#_Toc493758830)

[Table 9 - OTA\_HotelResRQ schema 141](#_Toc493758831)

[Table 10 - Supported codes for OTA\_HotelResRQ 148](#_Toc493758832)

[Table 11 - OTA\_HotelResRS schema 150](#_Toc493758833)

[Table 12 - Supported codes for OTA\_HotelResRS 152](#_Toc493758834)

[Table 13 - OTA\_HotelResModifyRQ transaction ResStatus 160](#_Toc493758835)

[Table 14 - OTA\_HotelResModifyRS transaction ResStatus 160](#_Toc493758836)

[Table 15 - OTA\_HotelResModifyRQ schema 170](#_Toc493758837)

[Table 16 - OTA\_CancelRQ schema 176](#_Toc493758838)

[Table 17 - OTA\_CancelRQ supported codes 177](#_Toc493758839)

[Table 18 - OTA\_CancelRS schema 180](#_Toc493758840)

[Table 19 - OTA\_CancelRS supported codes 181](#_Toc493758841)

[Table 20 - OTA\_HotelResModifyNotifRQ schema 187](#_Toc493758842)

[Table 21 - OTA\_HotelResModifyNotifRQ supported codes 188](#_Toc493758843)

[Table 22 - OTA\_HotelResModifyNotifRS schema 191](#_Toc493758844)

# Revision

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Changes | Change By |
|  |  |  |  |
|  |  |  |  |
| 2017-09-21 | 2.6 | - Added support for serach by RatePlanType in messages OTA\_HotelAvailRq – Section 7.2.2. (Please view Supported Codes in attachments)  - Added support for serach by RatePlanType in messages OTA\_HotelRatePlanRQ – Section 7.5.1. (Please view Supported Codes in attachments)  - Added new Codes for RatePlanType Chartering = 10001 and Exclusive = 10002. | Carlos Delgado |
| 2017-06-05 | 2.5 | - Add support for Rate Plan Type Code in messages OTA\_HotelAvailRS and OTA\_HotelRatePlanRQ. (Please view Supported Codes in attachments)  - Add support for new images at messages OTA\_HotelDescriptiveInfo, OTA\_HotelRatePlan, OTA\_HotelAvail. Please check section Images as Sytem information and schema at specified messages.  - Added Meal plan Name and Description to objects OTA\_HotelAvailRS and OTA\_RatePlanRS (Section 7);  - Change information about how to send On Request reservations in section 6.10. | Carlos Delgado |
| 2017-02-23 | 2.4 | - Added new Currencies to Currencies codes (Please view Supported Codes in attachments)  - Added new Languages in Languages code (Please view Supported Codes in attachments);  - Added Document ID to Guest profile in SendHotelRes (Section 7.3.1) | Carlos Delgado |
| 2017-02-10 | 2.3 | Added ISOCountry in POS object for Reservations requests in section Section 7.3.1.2  Added information about requests for cache systems in section 4 | Carlos Delgado |
| 2016-09-14 | 2.2 | Added information about Raterange in version Notes and HotelSearch Response.  Added information abou address line in HotelAvail Response message | Carlos Delgado |
| 2016-03-01 | 2.0.1 | Added information about version 2.0.1 in Section 1  Added Important note about how Pull Service manage releases and the version methodology  Added information about removed warning message in reservation response for deposit policies in section 1  Added information about correct using of warnings in Reservation Response at General Rules in section 7.3.1.3.2 | Carlos Delgado |
| 2016-01-30 | 2.0 | Added Information about version 2.0 – Section 1.1.1;  Added information about Booking on request – Section 6.10;  Added information about rates that are not for sale – Section 6.11;  Update objects for hotel avail messages on request – Section 7.2.2;  Update objects for hotel avail messages on response – Section 7.2.3;  Added information about Notify system section 3.2 and 5.8;  Added information about notify messages – section 7.3.5;  Added information about Search by currency – Section 7.2.2;  Added information about Search package net rates, net rates and commissionable rates – Section 7.2.2; | Carlos Delgado |
| 2015-09-10 | 1.9 | Added information about version 1.9  Added information about Rest services – Section 4  Added information about objects help pages – Section 4  Added information about search for modify – section 7.2.2  Added information about Modify a reservation – Section 7.3.2  Added new Offers Types – Section 6.5, section 7.2.3  Added Attach for Ota enum codes | Carlos Delgado |
| 2015-06-18 | 1.8 | Added Commission values on HotelAvailResponse – Section 7.2.3  Added Prices per pax on HotelAvailResponse – Section 7.2.3  Added new deposit policies rules (costs and deadline) – Sections 6.1, 7.1.3, 7.3.1.2  Added information about partner reservations number on HotelRes. Section 7.3.1  Added information about get reservation by partner reservation number.  Added annotations on this document with changes in version 1.8 | Carlos Delgado |
| 2015/04/21 | 1.7.1 | Add TPA\_Extesions.ApprovalInvoiced on 7.23 OTA\_HotelAvailRS Schema and general rules 7.f | Carlos Delgado |
| 2015/03/10 | 1.7 | Add best practices for paging on hotel avail on system information | Carlos Delgado |
| 2015/02/20 | 1.7 | Add Information about zones on HotelSearch  Update class diagrams for HotelSearch  Update general rules for hotelerach  Add Information about zones on HotelAvail  Add information about hotel position on Hotel Avail response  Update class diagrams for HotelAvail  Update general rules for hotelAvail  Add information about method GetZonesList | Carlos Delgado |
| 2014/10/31 | 1.6 | Change Information abou Agencies Office on GetAgenciesRequest  Update GetAgencies Diagram  Update GetAgencies Examples  Change information about agency office in Bookin section  Change information about agency offices in HotelAvail Section | Carlos Delgado |
| 2014/10/13 | 1.5.2 | Add information about Agencies Office on GetAgenciesRequest.  Add information about Logins types section 5.3 Add information about Operator requests  Add information about Self booking tools  Add information about Self booking tools, agencies requests  Add information about Self booking tools, corporates requests  Add information about services in hotel res request | Carlos Delgado |
| 2014/10/01 | 1.5.1 | Add information about new hotelsearch that allow to search all hotels in database.  Update hotelsearch examples  Add hotel search class diagram  Add hotel general rules | Carlos Delgado |
| 2014/09/15 | 1.5 | Add new type of rate IsPackage  Changed 7.2.3.1 Changed 7.2.3.2  Updated examples at 7.2.3.5  Add example at 7.2.3.6  Changed 7.3.1.2.1 Changed 7.3.1.2.2  Add one more example at 7.3.1.4.1  Changed Payment Types  Add 6.8 Allowed Payment Types | Carlos Delgado |
| 2014/07/17 | 1.4 | Add information about cancelation costs  Add information about Promotional codes  Add information about Group Codes  Add information about multiple rooms  Add Schema tables on HotelAvail, HotelRes, HotelResCancel  Add General Rules on HotelAvail, HotelRes, HotelResCancel  Add information about supported codes  Add information about class diagrams with drawings  Add more examples Refresh old examples | Carlos Delgado |
| 2014/05/20 | 1.3 | Add information how to get availability for rooms with different occupation | Carlos Delgado |
| 2014/04/23 | 1.2 | Add business Rules  Add Information about deposit policies  Add Information about Cancellation Policies  Add Information about Tax Policies  Add Information about Meal Plan Included  Add Information about Offers  Add information about Included Extras  Add information about Additional Extras | Carlos Delgado |
| 2014/26/03 | 1.1.1 | Add Information OTA\_Search without paging  Add Information OTA\_HotelAvail without paging  Add Information OTA\_Search in a radius  Add Information OTA\_HotelAvail in a radius  Add Information OTA\_HotelAvail optional best rates only | Carlos Delgado |
| 2014/01/10 | 1.1 | Add OTA\_HotelDescriptiveInfoRQ information | Carlos Delgado |
| 2013/07/11 | 1 | Initial Version | Carlos Delgado |

## Version notes

**IMPORTANT NOTE:** **Pull system use agile versioning strategy** that relies heavily on backward compatibility for environments that must tolerate frequent change. **This means that pull service can have optional new objects at every release**. **Partners should not have schema validation on wsdl, neither a static wsdl**. At new releases, **Omnibees team will assume that partner code is ready to ignore new added objects** for compatibility check.

### Version 2.6

**For partners** that already use versions **previous than 2.6** and do **not have interest in new features,** **no action is needed**.

* Add support for serach by RatePlanType in messages OTA\_HotelAvailRq – Section 7.2.2. (Please view Supported Codes in attachments)
* Add support for serach by RatePlanType in messages OTA\_HotelRatePlanRQ – Section 7.5.1. (Please view Supported Codes in attachments)
* Add new Codes for RatePlanType Chartering = 10001 and Exclusive = 10002.

### Version 2.5

**For partners** that already use versions **previous than 2.4** and use On Request reservations, must change the status of reservations for initial booking. The on request reservations will only be accepted if the initial request has Hold on TransactionActionType.

* Add support for Rate Plan Type Code in messages OTA\_HotelAvailRS and OTA\_HotelRatePlanRQ. (Please view Supported Codes in attachments)
* Add support for new images at messages OTA\_HotelDescriptiveInfo, OTA\_HotelRatePlan, OTA\_HotelAvail. Please check section Images as Sytem information and schema at specified messages.
* Added Meal plan Name and Description to objects OTA\_HotelAvailRS and OTA\_RatePlanRS (Section 7);
* Change information about how to send On Request reservations in section 6.10.

### Version 2.4

In version, 2.4 Omnibees added new Currencies and Languages. Partners should update service reference in order to get hotels with new currencies, this update is mandatory, because we will send this hotel by location, by id or in Hotel search method.

* Added new Currencies to Currencies codes, currency MAD = 116 changed to 117; (Please view Supported Codes in attachments)
* Added new Languages in Languages code, this implementation is under development and can be released in next releases, the objects are added in this version to minimize the impact(Please view Supported Codes in attachments);
* Added Document ID to Guest profile in SendHotelRes (Section 7.3.1);

### Version 2.3

**For partners** that already use versions **previous than 2.2** and do **not have interest in new features,** **no action is needed**.

* Added information for ISOCountry at POS in Hotel reservations. Partners can send information about the country code of the request party;
* Added information about endpoints for cache system. We detect that several requests in Omnibees live system are to fill partner’s cache system. Omnibees created a specific environment to handle this requests. Omnibees recommend partners that need to cache data, to changes cache requests to Omnibees Cache Endpoints.

### Version 2.2

**For partners** that already use versions **previous than 2.1** and do **not have interest in new features,** **no action is needed**. (Remember if partner soap parser cannot ignore new null objects, partner should check compatibility in cert environment)

* Added a new functionality that will inform the “From” price for each hotel. This message will inform partner that hotel can sell rooms from specified price per day. The Method is HotelSearch and in response, the object will be RateRange. The partner has to send Version 2.2 in message to get the new object.
* Added AddressLine in message HotelAvail. This property will give the information about hotel address.

### Version 2.1

Minor changes to optimize performance, no impact to partners.

### Version 2.0.1

This version is only a minor version of Version 2.0.

**For partners** that already use versions **previous than 1.9** and do **not have interest in new features,** **no action is needed**. (Remember if partner soap parser cannot ignore new objects, partner should check compatibility in cert environment)

* Warnings in Reservations response – In Version 2.0 Pull service added a warning message for Deposit Policies in Voucher Payment. **This Message is removed and no longer is on response.**

This message was a problem for some partners assume on code a warning for fail reservation request without check PMS\_ResStatusType and Reservation Number.

The partner should check the Reservation PMS\_ResStatusType on response, if status is not null partner should assume that reservation is made successful and check the current status (the status of reservation can be found at Section 6.0 Business Rules.

### Version 2.0

**For partners** that already use versions **previous than 1.9** and do **not have interest in new features,** **no action is needed**.

The new features are:

1. Notify reservations – Omnibees has now a notify system for pull reservations. This system will allow partners to receive notifications for changes in reservations;
2. Rates that are not avail – Search now has an option to return rates that are not available to sell. The rates that are not available to sell will indicate the reason for not be available to sell (HotelAvailRS);
3. Booking on request – API now allow search including days on request and allow to book reservations on request. This service can be used with pull notif to get reservations notifications (HotelAvailRS, HotelRes);
4. Search will inform room max occupancy (HotelAvailRS);
5. Available allotment can be found at day level (HotelAvailRS);
6. Room Amenities are now translated in requested language (HotelDescritiveInfo);
7. Property Amenities are now translated in requested language (HotelDescritiveInfo);
8. Filter HotelAvail responses by currency – HotelAvail request now has the option to filter by currency requested (Please view section 7.2.2 HotelAvailRQ, Schema and General Rules);
9. Filter Package net rates, Net rates, Commissionable Rates – Hotel avail request now has the option to filter by type of total amount (Please view section 7.2.2 HotelAvailRQ, Schema and General Rules);

#### Recommendations

New features are implemented in pull dev web app at pulldev.omnibees.com. Partner is free to use this application with certification credentials. This web application allow partner to test new features without code. Please contact support team to get help on web application.

For upgrade to version to version 2.0, partner should use certification environment to implement new features and to certify any requirements for new features.

### Version 1.9

**For partners** that already use versions **previous than 1.9** and do **not have interest in new features,** **no action is needed**.

For partners that what to upgrade to version 1.9 the changes in version 1.8 must be implemented before version 1.9.

The new features are:

1. Modify reservations – Pull api now allowed partners to modify reservations. Partners can search availability and prices to modify the reservation.
2. New Offers are now sent in availability and prices
   1. Discount offer;
   2. Stay discount with number of minimum nights;
   3. Stay discount with number of minimum and maximum nights

#### Recommendations

New features are implemented in pull dev web app at pulldev.omnibees.com. Partner is free to use this application with certification credentials. This web application allow partner to test new features without code. Please contact support team to get help on web application.

For upgrade to version to version 1.9, partner should use certification environment to implement new features and to certify any requirements for new features.

### Version 1.8

**For partners** that already use versions **previous than 1.8** and do **not have interest in new features,** **no action is needed**.

The new features are:

1. **Commission value** - OTA\_HotelAvail has now the commission value for rates rooms with amount including markup;
2. **Price per day per guest** – OTA\_HotelAvail has now information about day prices per guest grouped by ages;
3. **Price total per guest (with taxes)** – OTA\_HotelAvail has now information about total prices per guest Age qualifying code (Adult, child, Infant);
4. **Deposit policies with dead line** – Deposit policies has now information about dead line, the time before arrival that deposit policy will be applied;
5. **Deposit policies with values for deposit –** Deposit policies has now information about the value (amount, percent, nr nights) required for deposit for reservation;
6. **Deposit polices with guarantees by room –** Deposit policies has now information about the guarantees accepted for deposit in each room;
7. **Guarantees for Deposit accepted in reservation –** Reservation system now accepts Guarantee for deposit in reservation;
8. **Partner reservation number –** Partner can now send reservation number for mapping with Omnibees reservation number;
9. **Partner search booking with own reservation –** Partner can now send own reservation number to get one reservation.

To use version 1.8 the version property on requests has to be 1.8.

When change to 1.8 from previous version the payments types for reservation are e new object called Payment Policies. This implementation must be done in order to get payment types allowed for reservation.

#### Recommendations

1. Guid on correlation id – It is recommended to always send a new guid for searches in correlation uid.
2. Correlation id - We recommend that partners use the correlation id for reservation equal the search that has the result for reservation. Omnibees can track better the log messages;
3. Warning in reservation system- Please take care of warnings returned in reservation calls. This warnings are ota messages with the reason why the reservation cannot be completed. The reservation only has success when response has success tag and no warnings;
4. Retries in reservation system – Please do not retry to send reservation again when the response has warnings.

# Documentation Overview

The Pull system at Omnibees is OTA based version 2012B (Opentravel, 2013).

On next illustration are the mains Pull system and Notif messages.



Figure 1 - Main messages

On Messages section you have detailed information about the messages that you can use within web service:

1. **Ping** – This message allow to check connectivity between partner system and Omnibees Pull system;
2. **HotelSearch** – This messages provides information about active hotels. This information includes Hotel information, address, contacts, rating, check-in and check-out policies, geo location, room amenities and hotel amenities;
3. **HotelAvail** – This messages provides information about availabilities and prices about one specific hotel by hotel id, or about a range of hotels search with detailed criteria such as IATA code, Country code, state code, city code, rating, etc. The detail information to check you to filter and how to get codes is on Messages section.
4. **Booking Service** – This service is split in the following messages
   1. **SendHotelRes** – This message allow partner to send a reservation after use message HotelAvail. To prevent errors on message the use of HotelAvail before SendHotelRes is recommended;
   2. **SenHotelResModify** – This message allow partner to modify one booking. To prevent error on message the use of HotelAvail before SendHotelResModify is recommended;
   3. **SendHotelResCancel** – This message allow partner to cancel one booking para reservation number. To check authenticity of message a Verification fields is required. Message detail to get this information can be found at messages section;
   4. **GetBookings** – This messages provide information about single or multiple bookings. Filters to get bookings are allowed such as hotel id, last modify date from - to;
   5. **HotelResModifyNotif** – This messages allow partner to receive notification for changes in reservations. This is a simple message that allow partner do know that something is changed in reservation. After receive message partner can and should call Get Bookings with Omnibees reservation number to get details about current state of the reservation;
5. **GetHotelDescriptiveInfo** - This message provides information about hotel, rooms, multimedia, policies, etc.
6. **GetHotelRatePlans –** This message provide information about rate plans for one hotel.
7. **GetAgencies** – This message provides information about one agency or multiple agencies. This message allow to check agencies PCC codes. This requests includes information about agency offices;
8. **GetCorporates** – This message provides information about one Corporate or multiple corporates. This message allow to check corporates codes;
9. **GetCountryList** – This message provides information about system country codes;
10. **GetStateList** – This message provides information about states codes by country code;
11. **GetCityList** – This message provides information about cities codes by country code and state code.
12. **GetZonesList** – This message provides information about zones by Country code, StateSearchCode, ZoneSearchCode, CitiesDataUid.

**NOTE:** Examples on this document are just too see how objects works with wichother. You should not consider data in examples valid for any environment. If you try this examples in any environment it could not work because data configured in environment can be different. An example are the hotel ref that are diferent in all environments.

Please use contact support at integration.team@omnibees.com for any doubts.

# Protocols

## Pull Service

This section describes the protocol used to communicate with Omnibees Pull service.

The web service is on Windows Communication foundation (WCF), hosted on internet information Services (IIS).

Pull system uses WCF BasicHttpBinding.

The BasicHttpBinding uses HTTP as the transport for sending SOAP 1.1 messages. A service can use this binding to expose endpoints that conform to WS-I BP 1.1, such as those that ASMX clients consume. Similarly, a client can use the BasicHttpBinding to communicate with services exposing endpoints that conform to WS-I BP 1.1, such as ASMX Web services or services configured with the BasicHttpBinding (Microsoft, 2012).

A wsdl is supplied with all service contracts and data contracts.

In all requests, methods need a Login object to authenticate and use the service.

## Notify System

The notify system uses Rest message to communicate with partner. Partner should prepare the handler to receive Omnibees message and send back the response. The authentication username and password will be sent in HTTP Header.

# Connectivity Endpoints

This section describes the connectivity endpoints. The Omnibees Pull system uses different endpoints for test environment, certification environment and production/live environment.

**NOTE: Please use** [**http://pullcert.omnibees.com/Webapi/Help/**](http://pullcert.omnibees.com/Webapi/Help/) **to get detail information about the messages objects and enumerators that can be used in messages.**

## Endpoint URL for testing environment

<http://pulldev.omnibees.com>

At this environment the partner can access to tools to test Pull system. The partner will find documentation and xml examples at this environment.

The Partner is free to use this web application to test pull features without any implementation. This is very useful to get examples before start coding.

The credentials are the same used for dev/cert environment.

## Endpoint URL for dev/cert Environment

At this environment the partner will find the pull web services to star implementation and for certification. The environment for certification and development are the same. To certification Omnibees support team will ask xml for specify scenarios.

### SOAP Xml messages

<http://pullcert.omnibees.com>

### REST Json messages

<http://pullcert.omnibees.com/webapi>

## Endpoint URL for Production/Live environment

### SOAP Xml messages

<https://pull.omnibees.com>

### REST Json messages

<https://pullrest.omnibees.com>

## Endpoint URL for Production/Live environment for cache systems

This enpoints itend to Help partners that want to cache Omnibees data at Partners servers.

This eviroment will have better performance for cache systems, and can hadle several request at time.

### SOAP Xml messages

<https://pullcache.omnibees.com>

### REST Json messages

<https://pullrestcache.omnibees.com>

## Method

1. The partner ask credentials for dev/cert environment, and send the ip’s to whitelist at pull system. When credentials are supplied the partner should test it with service PingRQ. There should develop use pull API test all in dev environment.
2. When partner finish all the development, the partner should ask credentials, for certification. The certification will includes specific tests that will be check on Omnibees and Partner system.
3. When partner are certified, the Production/Online environment will be configured and the partner will receive the credentials for Production environment.

# System information

This section describes the system messaging.

## Login

The login object will be required in all messages. The user name and password need to be valid for success communication.

## Message

All message should have the Login object and the method object.

### Communication Success

When communication is done successfully you will receive a success tag.

<Success>

<ExtensionData />

</Success>

### Communication Success but fail on business rules

When something is wrong on message like missing tags, or business rules not valid, the response will have a success for communication messaging and warnings on object with OTA code and a description on what business rule is not sent.

<WarningsType>

<ExtensionData />

<Warnings>

<Warning>

<ExtensionData />

<Code>193</Code>

<Language>en</Language>

<ShortText>Cancellation process failed: Cannot cancel reservation. Verification data is not valid.</ShortText>

</Warning>

</Warnings>

</WarningsType>

On example above:

1. OTA code for warning: <Code>193</Code>
2. OTA message for warning: <ShortText> Cancellation process failed:
3. Omnibees custom message: Cannot cancel reservation. Verification data is not valid.</ShortText>

### Communication Fail

When communication fail, like some error on Omnibees pull system given by exception the response will have an error tag with id for communication with our support team.

<ErrorsType>

<ExtensionData />

<Errors>

<Error>

<ExtensionData />

<Code>448</Code>

<Language>en</Language>

<ShortText>System error: Internal system error. Please contact techincal support at integration.team@omnibees.com { echoToken = b7ff5191-38b1-4cd3-bc2f-e8676c9ef97e }</ShortText>

</Error>

</Errors>

</ErrorsType>

On example above:

1. OTA code for error: <Code>448</Code>
2. OTA message for error: <ShortText> System error:
3. Omnibees custom message: Internal system error. Please contact technical support at integration.team@omnibees.com { echoToken = b7ff5191-38b1-4cd3-bc2f-e8676c9ef97e }</ShortText>
4. Echo token to use in support: echoToken = b7ff5191-38b1-4cd3-bc2f-e8676c9ef97e

## Login Type

There are five types of login to make requests in Omnibees Pull Service:

1. Operator
2. Self-booking tool – Agencies
3. Self-booking tool – Agencies Offices
4. Self-booking tool – Corporates
5. Booking Engine

Please check the next section to understand the features and restrictions for each login type.

## Operator

### Login Object

As an operator the validation system check the login name and password.

For Operators the login object must be sent.

<Login>

<UserName>UserName</UserName>

<Password>Password</Password>

</Login>

### Payment Types

The allowed payment types that can be used on reservation is restricted to the response on hotel avail. In case of operator, the payment type that can appear on hotel avail are:

1. DirectBill = 2
2. Deposit = 8
3. CreditCard = 5
4. Voucher = 3
5. PrePay = 4 (this is a special case, not all operators can use this type of payment)

Please view section 6.8 to know more details about payment types.

## Self-Booking Tool

### Login

Self-booking tool login type uses tree level of authentication: username, password and Code.

The username and password must be sent in all requests

<Login>

<UserName>UserName</UserName>

<Password>Password</Password>

</Login>

In addition a **code must be sent for messages**:

1. HotelAvail;
2. SendHotelRes;
3. SendHotelResCancel;
4. GetBookings;

The code is unique for type of self-booking tool. Please see next sections according to self-booking tool type.

Self-booking tools **are not allowed** to use messages:

1. GetAgencies;
2. GetCorporates.

### Paymet Types

The allowed payment types that can be used on reservation is restricted to the response on hotel avail. In case of self-booking tool the payment type that can appear on hotel avail are:

1. DirectBill = 2
2. Deposit = 8
3. CreditCard = 5
4. Voucher = 3

Please view section 6.8 to know more details about payment types.

### Agency

For the agencies the code to send is PCC - pseudo city code.

This PCC will allow to match agency with username and password and validate the requests. The PCC code must be sent in messages: HotelAvail, SendHotelRes, SendHotelResCancel and GetBookings.

The PCC must be sent in POS object.

<POS>

<Sources>

<Source>

<PseudoCityCode>TESTP</PseudoCityCode>

</Source>

</Sources>

</POS>

### Agency Office

For the agencies office the code to send is Requestor ID with type CorporationRepresentative.

This code will allow to match agency office with username and password and validate the requests. The code must be sent in messages: HotelAvail, SendHotelRes, SendHotelResCancel and GetBookings.

The requestor ID must be sent in POS object.

<POS>

<Sources>

<Source>

<RequestorID>

<ID>TESTP|ABB</ID>

<Type>CorporationRepresentative</Type>

</RequestorID>

</Source>

</Sources>

</POS>

### Corporate

For the Corporates the code to send is Unique ID with type Company.

This code will allow to match company with username and password and validate the requests. The code must be sent in messages: HotelAvail, SendHotelRes, SendHotelResCancel and GetBookings.

1. **HotelAvail**

<HotelSearchCriteria>

<Criterion>

<ProfilesType>

<ProfileInfos>

<ProfileInfo>

<UniqueID>

<ID>ABB</ID>

<Type>Company</Type>

</UniqueID>

</ProfileInfo>

</ProfileInfos>

</ProfilesType>

...

</Criterion>

</HotelSearchCriteria>

1. **SendHotelRes**

<HotelReservation>

<ResGlobalInfo>

<HotelReservationsIDsType>

<HotelReservationIDs>

<HotelReservationID>

<ResID\_Type>Company</ResID\_Type>

<ResID\_Value>ABB</ResID\_Value>

</HotelReservationID>

</HotelReservationIDs>

</HotelReservationsIDsType>

...

</ResGlobalInfo>

..

</HotelReservation>

1. **SendHotelResCancel**

<UniqueID>

<UniqueID>

<ID>ABB</ID>

<Type>Company</Type>

</UniqueID>

..

</UniqueID>

1. **GetBookings**

<CompanyID>

<ID>ABB</ID>

<Type>Company</Type>

</CompanyID>

## Booking Engine

### Login Object

As a booking engine the validation system on check the login name and password.

For booking engine the login object must be sent.

<Login>

<UserName>UserName</UserName>

<Password>Password</Password>

</Login>

### Payment Types

The allowed payment types that can be used on reservation is restricted to the response on hotel avail. In can of operator the payment type that can appear on hotel avail are:

1. DirectBill = 2
2. Deposit = 8
3. CreditCard = 5
4. Voucher = 3

## Hotel Avail– Best Practices

### Filter

For make requests to hotel avail is required to use a filter by hotel code or by country and statecode. If Omnibees receives any request with filter, no availability will be response.

One of the following search type is mandatory.

#### Search availability by hotel

If partner need to make request to get availability just for one specific hotel then search by hotel id is recommended.

#### Search availability by Location

If partner need to make request for a specify location then the search must be by location.

**One** of following type is **mandatory** for location

1. Send Country code and state code and city code (Recommended)
2. Send Country code and state code
3. Send Country code and city code

**Optional types** for request on hotel availability

1. Send Country code and state code and city code and zone code

**Not allowed types** for request on hotel availability

1. Not send country code
2. Send only country code and zone code

### Best Only rates

If the partner only needs the lowest rate available for each hotel for the hotels price list in client website, then the tag BestOnly should be sent as true. With BestOnly = true Omnibees only returns lowest rate for each hotel.

If the client website allows to check mores rates for each hotel then the partner, after client enter on hotel detail, should sent a request to Omnibees with the hotel id and the tag BestOnly = false, so the response can has all the rates for the hotel.

### Paging

On messages Omnibees always recommended to use paging. The main idea of paging is to make small response so communication between server and client can be fast.

#### Paging according to client website (Recommended)



Figure 2 - Pagging according to client website

This is the basic example of paging. A website show 10 hotels per page. The hotel prices are loaded dynamically according to page the end user is requesting.

For each time the end user goes to next page the partner should make a request to Omnibees server with page the user is requested at partner website.

The partner should check tag MoreAvailable on the response. If the MoreAvailable tag is set to true the partner can continue increment pages. If MoreAvailable is set to false then the partner no need to increment the page and should not allow end user to change to next page, because no more hotels are available at Omnibees.

#### Load all data at once

When the partner needs to load all avail prices (according to requested filter) at once, then the partner should not use paging.

No need to used paging if the request is get all hotels in client website. But this can be an issue because the messages can be in order of Megabytes and the communication between two systems cannot be done fast enough before client gets a timeout.

## Notify Reservations

In version 2.0 a new features is introduced, the Notify system. The notify system will allow partners to get informed for any change in reservation already made in pull system.

The Notify system will notify partner in a URL that partner provide to Omnibees at the beginning of integration. Partner will need to have a rest API to handle notifications.

The objects must equal to objects expected by Omnibees. This objects description can be found at section 7 – HotelResModifyNotif.

### Reservation Status

The reservation notification will have the reservation number and the current status of a reservation. The status can be:

|  |  |
| --- | --- |
| Code | Description |
| Requested | The reservation is on “On Request” state, waiting for hotel to approve. |
| Reserved | The reservation is Booked. |
| RequestDenied | The hotel did not approved the “On request” reservation. |
| RequestAccepted | The hotel approved the “On request” reservation. |
| Cancelled | Reservation is cancelled. |
| Waitlisted | Reservation is in pending state and is waiting for some other action. |
| Modified | Reservation is modified and is booked. |

Table 1 - Reservation Status

## Images

### OTA\_HotelDescriptiveInfo

To get images the object OTA\_HotelDescriptiveInfoRQ should have the flag to get images equal to true:

<d4p1:MultimediaObjects>

<d4p1:SendData>true</d4p1:SendData>

</d4p1:MultimediaObjects>

#### Properties

All images from properties goes in response at object:

HotelInfo.Descriptions.MultimediaDescriptionsType.MultimediaDescriptions

This is an array of descriptions for the property. To get the property images the partner should chose objects that meet the condition:

1. InfoCode = Marketing;
2. AdditionalDetailCode = PropertyDescription;
3. ImageItemsType not null

Example:

<HotelInfo>

...

<Descriptions>

<MultimediaDescriptionsType>

<MultimediaDescriptions>

<MultimediaDescription>

<AdditionalDetailCode>PropertyDescription</AdditionalDetailCode>

<ImageItemsType>

<ImageItems>

<ImageItem>

<Name>9804.jpg</Name>

<URL>

<Address>http://media.omnibees.com/Images/1814/Property/22622.jpg</Address>

<URLType>Image</URLType>

</URL>

</ImageItem>

</ImageItems>

</ImageItemsType>

<InfoCode>Marketing</InfoCode>

</MultimediaDescription>

</MultimediaDescriptions>

</MultimediaDescriptionsType>

</Descriptions>

...

</HotelInfo>

#### Rooms

All the room images goes in object:

FacilityInfo.GuestRoomsType.GuestRoom.MultimediaDescriptionsType.MultimediaDescriptions

This is an array of descriptions for the room. To get the images the partner should choose objects that meet the condition:

1. InfoCode = Marketing;
2. AdditionalDetailCode = RoomInformation;
3. ImageItemsType not null

Example:

<FacilityInfo>

<GuestRoomsType>

<GuestRooms>

<GuestRoom>

<MultimediaDescriptionsType>

<MultimediaDescriptions>

<MultimediaDescription>

<AdditionalDetailCode>RoomInformation</AdditionalDetailCode>

<ImageItemsType>

<ExtensionData />

<ImageItems>

<ImageItem>

<Name>thS96MC9OZ.jpg</Name>

<URL>

<Address>http://media.omnibees.com/Images/1814/RoomTypes/33117.jpg</Address>

<URLType>Image</URLType>

</URL>

</ImageItem>

</ImageItems>

</ImageItemsType>

<InfoCode>Marketing</InfoCode>

</MultimediaDescription>

</MultimediaDescriptions>

</MultimediaDescriptionsType>

</GuestRoom>

</GuestRooms>

</GuestRoomsType>

</FacilityInfo>

#### Attractions

All the attraction images goes in object:

AreaInfo. AttractionsType.Attraction.MultimediaDescriptionsType.MultimediaDescriptions

This is an array of descriptions for the attraction. To get the images the partner should choose objects that meet the condition:

1. InfoCode = Attractions;
2. AdditionalDetailCode = MiscellaneousInformation;
3. ImageItemsType not null

Example:

<AreaInfo>

<AttractionsType>

<Attractions>

<Attraction>

<MultimediaDescriptionsType>

<MultimediaDescriptions>

<MultimediaDescription>

<AdditionalDetailCode>MiscellaneousInformation</AdditionalDetailCode>

<ImageItemsType>

<ImageItems>

<ImageItem>

<Name>106610.jpg</Name>

<URL>

<Address>http://media.omnibees.com/Images/1814/Attractions/23105.jpg</Address>

<URLType>Image</URLType>

</URL>

</ImageItem>

</ImageItems>

</ImageItemsType>

<InfoCode>Attractions</InfoCode>

</MultimediaDescription>

</MultimediaDescriptions>

</MultimediaDescriptionsType>

</Attraction>

</Attractions>

</AttractionsType>

</AreaInfo>

### OTA\_HotelRatePlan

To get images the object OTA\_ HotelRatePlan should have the flag to get images equal to true:

<d4p1:TPA\_Extensions>

<d4p1:MultimediaObjects>

<d4p1:SendData>true</d4p1:SendData>

</d4p1:MultimediaObjects>

</d4p1:TPA\_Extensions>

#### MealPlan (Boardtype extra), RateplanInclusions (included extras), services (optional extras)

All images from objects MealPlanIncluded, RatePlanInclusion, Services are in the object:

RatePlans.TPA\_Extensions.MultimediaDescriptionsType

To get images from specific objects, the id should be used with the following codes are used:

1. MealPlanIncluded:
   1. ID = MealPlanIncluded.ID
   2. InfoCode = Marketing;
   3. AdditionalDetailCode = ServiceInformation;
   4. ImageItemsType not null

Example:

<a:MealsIncluded>

<a:ID>2445</a:ID>

</a:MealsIncluded>

…  
<a:MultimediaDescription>

<a:AdditionalDetailCode>ServiceInformation</a:AdditionalDetailCode>

<a:ID>2445</a:ID>

<a:ImageItemsType>

<a:ImageItems>

<a:ImageItem>

<a:Name>half-board-meals-offer-p.jpg</a:Name>

<a:URL>

<a:Address>http://media.omnibees.com/Images/1814/Extras/23108.jpg</a:Address>

<a:URLType>Image</a:URLType>

</a:URL>

</a:ImageItem>

</a:ImageItems>

</a:ImageItemsType>

<a:InfoCode>Marketing</a:InfoCode>

</a:MultimediaDescription>

1. RatePlanInclusions
   1. ID = RatePlanInclusion.ID
   2. InfoCode = Marketing;
   3. AdditionalDetailCode = ServiceInformation;
   4. ImageItemsType not null

Example

<a:RatePlanInclusion>

<a:ID>2444</a:ID>

</a:RatePlanInclusion>  
…  
<a:MultimediaDescription>

<a:AdditionalDetailCode>ServiceInformation</a:AdditionalDetailCode>

<a:ID>2444</a:ID>

<a:ImageItemsType>

<a:ImageItems>

<a:ImageItem>

<a:Name>champagnse.jpg</a:Name>

<a:URL>

<a:Address>http://media.omnibees.com/Images/1814/Extras/23109.jpg</a:Address>

<a:URLType>Image</a:URLType>

</a:URL>

</a:ImageItem>

</a:ImageItems>

</a:ImageItemsType>

<a:InfoCode>Marketing</a:InfoCode>

</a:MultimediaDescription>

1. Service
   1. ID = Service.ID
   2. InfoCode = Marketing;
   3. AdditionalDetailCode = ServiceInformation;
   4. ImageItemsType not null

Example:

<a:Service>

<a:ID>2444</a:ID>

</a:Service>  
…  
<a:MultimediaDescription>

<a:AdditionalDetailCode>ServiceInformation</a:AdditionalDetailCode>

<a:ID>2444</a:ID>

<a:ImageItemsType>

<a:ImageItems>

<a:ImageItem>

<a:Name>champagnse.jpg</a:Name>

<a:URL>

<a:Address>http://media.omnibees.com/Images/1814/Extras/23109.jpg</a:Address>

<a:URLType>Image</a:URLType>

</a:URL>

</a:ImageItem>

</a:ImageItems>

</a:ImageItemsType>

<a:InfoCode>Marketing</a:InfoCode>

</a:MultimediaDescription>

#### Packages

All the packages images goes in object:

RatePlans.TPA\_Extensions.MultimediaDescriptionsType

To get images from specific objects, the id should be used with the following codes are used:

1. HotelRatePlan:
   1. ID = RatePlanID.ID
   2. InfoCode = Marketing;
   3. AdditionalDetailCode = PackageInformation;
   4. ImageItemsType not null

Example:

<a:HotelRatePlan>

<a:RatePlanID>33638</a:RatePlanID>

<a:RatePlanTypeCode>11</a:RatePlanTypeCode>

</a:HotelRatePlan>  
…  
<a:MultimediaDescription>

<a:AdditionalDetailCode>PackageInformation</a:AdditionalDetailCode>

<a:ID>33638</a:ID>

<a:ImageItemsType>

<a:ImageItems>

<a:ImageItem>

<a:Name>zoomarine.jpg</a:Name>

<a:URL>

<a:Address>http://media.omnibees.com/Images/1987/Packages/23028.jpg</a:Address>

<a:URLType>Image</a:URLType>

</a:URL>

</a:ImageItem>

</a:ImageItems>

</a:ImageItemsType>

<a:InfoCode>Marketing</a:InfoCode>

</a:MultimediaDescription>

### OTA\_HotelAvail

To get images the object OTA\_ HotelAvail should have the flag to get images equal to true:

<d4p1:TPA\_Extensions>

<d4p1:MultimediaObjects>

<d4p1:SendData>true</d4p1:SendData>

</d4p1:MultimediaObjects>

</d4p1:TPA\_Extensions>

All the images in HotelAvail goes in object:

TPA\_Extensions.MultimediaDescriptionsType

To get images from specific objects, the id should be used with the following codes are used:

#### Properties

1. ID = HotelRef.HotelCode
2. InfoCode = Marketing;
3. AdditionalDetailCode = PropertyDescription;
4. ImageItemsType not null

Example:

<a:MultimediaDescription>

<a:AdditionalDetailCode>PropertyDescription</a:AdditionalDetailCode>

<a:ID>1814</a:ID>

<a:ImageItemsType>

<a:ImageItems>

<a:ImageItem>

<a:Name>hotel 2.jpg</a:Name>

<a:URL>

<a:Address>http://media.omnibees.com/Images/1814/Property/22631.jpg</a:Address>

<a:URLType>Image</a:URLType>

</a:URL>

</a:ImageItem>

</a:ImageItems>

</a:ImageItemsType>

<a:InfoCode>Marketing</a:InfoCode>

</a:MultimediaDescription>

#### Rooms

1. ID = RoomType.RoomID
2. InfoCode = Marketing;
3. AdditionalDetailCode = RoomInformation;
4. ImageItemsType not null

Example:

<a:MultimediaDescription>

<a:AdditionalDetailCode> RoomInformation</a:AdditionalDetailCode>

<a:ID>1814</a:ID>

<a:ImageItemsType>

<a:ImageItems>

<a:ImageItem>

<a:Name>hotel 2.jpg</a:Name>

<a:URL>

<a:Address>http://media.omnibees.com/Images/1814/RoomTypes/33117.jpg</a:Address>

<a:URLType>Image</a:URLType>

</a:URL>

</a:ImageItem>

</a:ImageItems>

</a:ImageItemsType>

<a:InfoCode>Marketing</a:InfoCode>

</a:MultimediaDescription>

#### MealPlan (Boardtype extra), RateplanInclusions (included extras), services (optional extras)

To get images from specific objects, the id should be used with the following codes are used:

1. MealPlanIncluded:
   1. ID = MealPlanIncluded.ID
   2. InfoCode = Marketing;
   3. AdditionalDetailCode = ServiceInformation;
   4. ImageItemsType not null

Example:

<a:MealsIncluded>

<a:ID>2445</a:ID>

</a:MealsIncluded>

…  
<a:MultimediaDescription>

<a:AdditionalDetailCode>ServiceInformation</a:AdditionalDetailCode>

<a:ID>2445</a:ID>

<a:ImageItemsType>

<a:ImageItems>

<a:ImageItem>

<a:Name>half-board-meals-offer-p.jpg</a:Name>

<a:URL>

<a:Address>http://media.omnibees.com/Images/1814/Extras/23108.jpg</a:Address>

<a:URLType>Image</a:URLType>

</a:URL>

</a:ImageItem>

</a:ImageItems>

</a:ImageItemsType>

<a:InfoCode>Marketing</a:InfoCode>

</a:MultimediaDescription>

1. RatePlanInclusions
   1. ID = RatePlanInclusion.ID
   2. InfoCode = Marketing;
   3. AdditionalDetailCode = ServiceInformation;
   4. ImageItemsType not null

Example

<a:RatePlanInclusion>

<a:ID>2444</a:ID>

</a:RatePlanInclusion>  
…  
<a:MultimediaDescription>

<a:AdditionalDetailCode>ServiceInformation</a:AdditionalDetailCode>

<a:ID>2444</a:ID>

<a:ImageItemsType>

<a:ImageItems>

<a:ImageItem>

<a:Name>champagnse.jpg</a:Name>

<a:URL>

<a:Address>http://media.omnibees.com/Images/1814/Extras/23109.jpg</a:Address>

<a:URLType>Image</a:URLType>

</a:URL>

</a:ImageItem>

</a:ImageItems>

</a:ImageItemsType>

<a:InfoCode>Marketing</a:InfoCode>

</a:MultimediaDescription>

1. Service
   1. ID = Service.ID
   2. InfoCode = Marketing;
   3. AdditionalDetailCode = ServiceInformation;
   4. ImageItemsType not null

Example:

<a:Service>

<a:ID>2444</a:ID>

</a:Service>  
…  
<a:MultimediaDescription>

<a:AdditionalDetailCode>ServiceInformation</a:AdditionalDetailCode>

<a:ID>2444</a:ID>

<a:ImageItemsType>

<a:ImageItems>

<a:ImageItem>

<a:Name>champagnse.jpg</a:Name>

<a:URL>

<a:Address>http://media.omnibees.com/Images/1814/Extras/23109.jpg</a:Address>

<a:URLType>Image</a:URLType>

</a:URL>

</a:ImageItem>

</a:ImageItems>

</a:ImageItemsType>

<a:InfoCode>Marketing</a:InfoCode>

</a:MultimediaDescription>

#### Packages

To get images from specific objects, the id should be used with the following codes are used:

1. HotelRatePlan:
   1. ID = RatePlanID.ID
   2. InfoCode = Marketing;
   3. AdditionalDetailCode = PackageInformation;
   4. ImageItemsType not null

Example:

<a:HotelRatePlan>

<a:RatePlanID>33638</a:RatePlanID>

<a:RatePlanTypeCode>11</a:RatePlanTypeCode>

</a:HotelRatePlan>  
…  
<a:MultimediaDescription>

<a:AdditionalDetailCode>PackageInformation</a:AdditionalDetailCode>

<a:ID>33638</a:ID>

<a:ImageItemsType>

<a:ImageItems>

<a:ImageItem>

<a:Name>zoomarine.jpg</a:Name>

<a:URL>

<a:Address>http://media.omnibees.com/Images/1987/Packages/23028.jpg</a:Address>

<a:URLType>Image</a:URLType>

</a:URL>

</a:ImageItem>

</a:ImageItems>

</a:ImageItemsType>

<a:InfoCode>Marketing</a:InfoCode>

</a:MultimediaDescription>

# Business Rules

## Deposit Policies

Deposit policies are in OTA\_HotelAvailRS to allow to know what kind of deposit is allowed for reservations.

For reservation Omnibees will only accepted deposit that are valid for the rate plan requested.

Please view section 7.2.3 and section 7.3.1 to know more about the objects.

Note: When the request for reservation contains deposit policy credit card, the payment method has to be credit card. Omnibees will use payment method credit card to save information on deposit.

### For OTA\_HotelAvailRS

#### Class Diagram

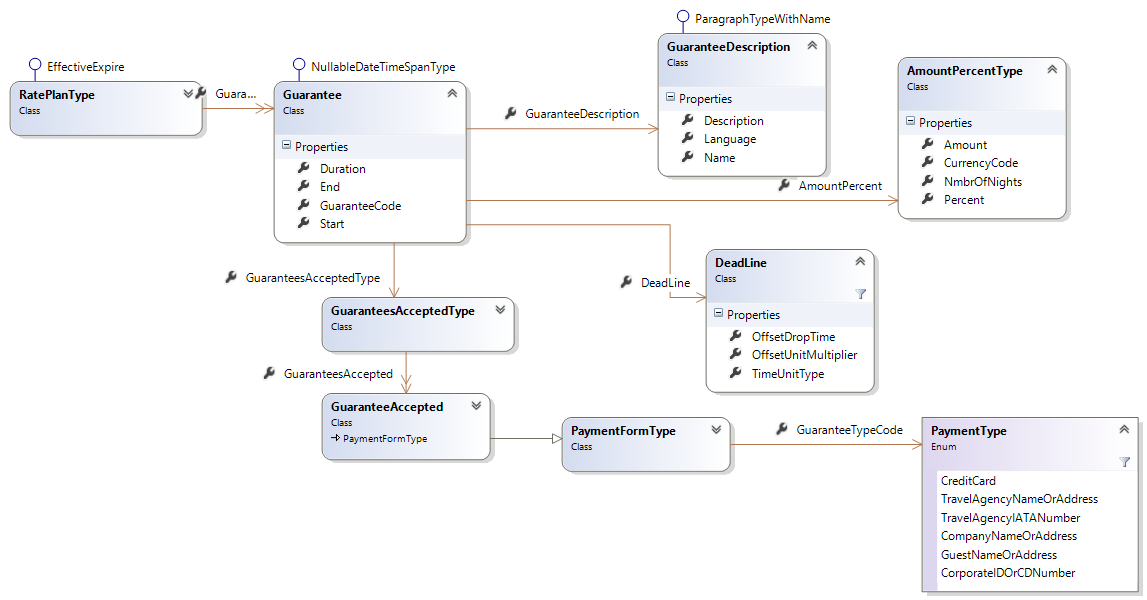


Figure 3 - Deposit Policies Class Diagram

#### Example

<Guarantees>

<Guarantee>

<AmountPercent>

<Amount>0</Amount>

<CurrencyCode>EUR</CurrencyCode>

<NmbrOfNights>2</NmbrOfNights>

<Percent>0</Percent>

</AmountPercent>

<DeadLine>

<OffsetDropTime>BeforeArrival</OffsetDropTime>

<OffsetUnitMultiplier>0</OffsetUnitMultiplier>

<TimeUnitType>Day</TimeUnitType>

</DeadLine>

<End>2015-06-25T00:00:00</End>

<GuaranteeCode>2481</GuaranteeCode>

<GuaranteeDescription>

<Description>Deposito Custo 5</Description>

<Language>en</Language>

<Name>Deposito Custo 5</Name>

</GuaranteeDescription>

<GuaranteesAcceptedType>

<GuaranteesAccepted>

<GuaranteeAccepted>

<GuaranteeTypeCode>CreditCard</GuaranteeTypeCode>

<RPH>0</RPH>

</GuaranteeAccepted>

<GuaranteeAccepted>

<GuaranteeTypeCode>CompanyNameOrAddress</GuaranteeTypeCode>

<RPH>0</RPH>

</GuaranteeAccepted>

</GuaranteesAccepted>

</GuaranteesAcceptedType>

<Start>2015-06-19T00:00:00</Start>

</Guarantee>

</Guarantees>

### For OTA\_HotelResRQ

When send a reservation request, the message has to have filled rateplan guarantees according to deposit policy returned in OTA\_HotelAvailResp.

The information what deposit guarantees is wanted for reservation has to be sent in object DepositPayments.

The deposit payment must be sent in each roomstay object.

When the deposit payment is credit card, the payment method must be credit card and valid information must be sent. Please view Section Payment policies at Business Rules.

The Guarante is the value for Guarantee type, for example: Iata number, agency name, etc…

Example:

<RoomStaysType>

<RoomStays>

<RoomStay>

<Guarantees>

<Guarantee>

<GuaranteesAcceptedType>

<GuaranteesAccepted>

<GuaranteeAccepted>

<GuaranteeID>Company Name</GuaranteeID>

<GuaranteeTypeCode>CompanyNameOrAddress</GuaranteeTypeCode>

</GuaranteeAccepted>

</GuaranteesAccepted>

</GuaranteesAcceptedType>

</Guarantee>

</Guarantees>

</RoomStay>

</RoomStays>

</RoomStaysType>

…

## Cancelation Policies

Cancelation policies are sent as parameterized object.

At hotelavail response one cancelation policy will be sent, and allways will be sent just one cancelation policy per rateplan.

The response for cancelation policies can be:

When cancelation is set for the rate the response contains the number of days before arrival the rule applies the cancelation costs info.

The number of days the cancelation cost applies is sent in deadline object.

When filled dealline object represents the days before arrival the cancelation policies are applied. Outside this days the cancelation has 0% of cost.  
The cancelaltion costs can be set as Cost or number of nights:

* 1. Cost
     1. Can be sent as percentage of total value, sent in tag AmountPercent.Percent
     2. Can be sent as value sent in AmountPercent.Amount
  2. Number of nights

Sent in tag AmountPercent.NmbrOfNights

* 1. NonRefundable Rates – The cancelation has cost of 100% of reservation total.

Sent NonRefundable = true

### Class Diagram

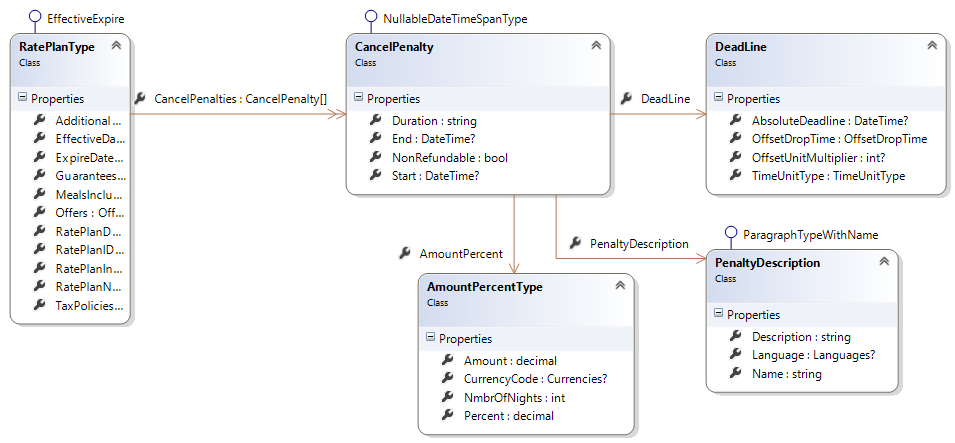


Figure 4 - Cancelation Policy class diagram

### Examples

#### Cancelation of non refundable rates with 100% of cost

This example is for cancelation policy valid for all periods and indicates that cancelation as a cost of total value.

<CancelPenalties>

<CancelPenalty>

<NonRefundable>true</NonRefundable>

<PenaltyDescription>

<Description>Rate non refundable</Description>

<Language xsi:nil="true" />

<Name>Cancelation policy</Name>

</PenaltyDescription>

</CancelPenalty>

</CancelPenalties>

#### Cost as number of nights

This Example is for a cancelation policies that applies in period from 2014-04-23 to 2014-04-30, in one day before arrival. When this condition applies, the cost for cancelation will be 3 nights.

<CancelPenalties>

<CancelPenalty>

<ExtensionData />

<AmountPercent>

<ExtensionData />

<Amount>0</Amount>

<CurrencyCode>EUR</CurrencyCode>

<NmbrOfNights>3</NmbrOfNights>

<Percent>0</Percent>

</AmountPercent>

<DeadLine>

<ExtensionData />

<AbsoluteDeadline xsi:nil="true" />

<OffsetDropTime>BeforeArrival</OffsetDropTime>

<OffsetUnitMultiplier>1</OffsetUnitMultiplier>

<TimeUnitType>Day</TimeUnitType>

</DeadLine>

<End>2014-04-30T00:00:00</End>

<NonRefundable>false</NonRefundable>

<PenaltyDescription>

<ExtensionData />

<Description>Allowed cancelation within 1 days before arrival with 3 nights of total cost</Description>

<Language xsi:nil="true" />

<Name>Cancelation policy</Name>

</PenaltyDescription>

<Start>2014-04-23T00:00:00</Start>

</CancelPenalty>

</CancelPenalties>

#### Cost as Value

This example is for cancelation policy valid for all periods and indicates that cancelation has a cost of 25 EUR.

<CancelPenalties>

<CancelPenalty>

<AmountPercent>

<Amount>25.00</Amount>

<CurrencyCode>EUR</CurrencyCode>

<NmbrOfNights>0</NmbrOfNights>

<Percent>0</Percent>

</AmountPercent>

<DeadLine>

<OffsetDropTime>BeforeArrival</OffsetDropTime>

<OffsetUnitMultiplier>5</OffsetUnitMultiplier>

<TimeUnitType>Day</TimeUnitType>

</DeadLine>

<NonRefundable>false</NonRefundable>

<PenaltyDescription>

<Description>Allowed cancelation within 5 days before arrival with 25 Euro</Description>

<Language xsi:nil="true" />

<Name>Cancelation policy</Name>

</PenaltyDescription>

</CancelPenalty>

</CancelPenalties>

#### Cost as Percentage

This example is for cancelation policy valid for all periods and indicates that cancelation has a cost of 25% of total cost.

<CancelPenalties>

<CancelPenalty>

<AmountPercent>

<Amount>0</Amount>

<CurrencyCode>EUR</CurrencyCode>

<NmbrOfNights>0</NmbrOfNights>

<Percent>25.00</Percent>

</AmountPercent>

<DeadLine>

<AbsoluteDeadline xsi:nil="true" />

<OffsetDropTime>BeforeArrival</OffsetDropTime>

<OffsetUnitMultiplier>5</OffsetUnitMultiplier>

<TimeUnitType>Day</TimeUnitType>

</DeadLine>

<End>2014-04-30T00:00:00</End>

<NonRefundable>false</NonRefundable>

<PenaltyDescription>

<Description>Allowed cancelation within 5 days before arrival with 25% of total cost</Description>

<Language xsi:nil="true" />

<Name>Cancelation policy</Name>

</PenaltyDescription>

<Start>2014-04-23T00:00:00</Start>

</CancelPenalty>

</CancelPenalties>

## Tax Policies

Tax policies are sent as parameterized object.

The tag amount after tax is the total of amount before tax plus the tax on tax policies object.

A tax Policy can be a value or percentage and can be charged as:

* Per Night per room
* Per Night per Person
* Per Stay per room
* Per Stay per person

### Class Diagram

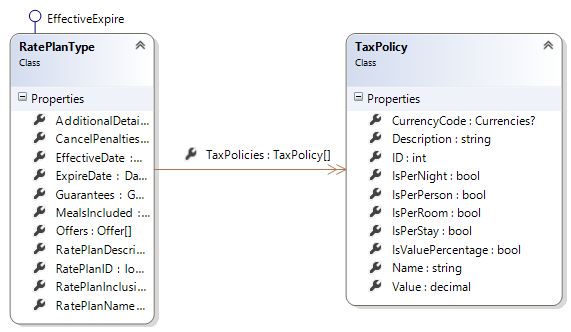


Figure 5 - Tax Policies class Diagram

### Examples

#### Tax Policy defined per stay per room in percentage

<TaxPolicies>

<TaxPolicy>

<ExtensionData />

<CurrencyCode>EUR</CurrencyCode>

<Description>Tax service</Description>

<ID>585</ID>

<IsPerNight>false</IsPerNight>

<IsPerPerson>false</IsPerPerson>

<IsPerRoom>true</IsPerRoom>

<IsPerStay>true</IsPerStay>

<IsValuePercentage>true</IsValuePercentage>

<Name>Tax service</Name>

<Value>5.00</Value>

</TaxPolicy>

</TaxPolicies>

#### Tax policy defined per stay per room defined in value

<TaxPolicies>

<TaxPolicy>

<ExtensionData />

<CurrencyCode>EUR</CurrencyCode>

<Description>Tax service</Description>

<ID>587</ID>

<IsPerNight>false</IsPerNight>

<IsPerPerson>false</IsPerPerson>

<IsPerRoom>true</IsPerRoom>

<IsPerStay>true</IsPerStay>

<IsValuePercentage>false</IsValuePercentage>

<Name>Tax service</Name>

<Value>5.00</Value>

</TaxPolicy>

</TaxPolicies>

#### Tax policy define per night per person in percentage

<TaxPolicies>

<TaxPolicy>

<ExtensionData />

<CurrencyCode>EUR</CurrencyCode>

<Description>Tax service</Description>

<ID>586</ID>

<IsPerNight>true</IsPerNight>

<IsPerPerson>true</IsPerPerson>

<IsPerRoom>false</IsPerRoom>

<IsPerStay>false</IsPerStay>

<IsValuePercentage>true</IsValuePercentage>

<Name>Tax service</Name>

<Value>5.00</Value>

</TaxPolicy>

</TaxPolicies>

## Meal Plan Included

Meal plan is sent as parameterized object.

Is sent on MealsIncluded object. The mealplan on this tag is included on total price.

If need to upgrade board and if it is available to rate it will sent as additional extras (service).

Available meal plans code are:

* BedBreakfast;
* HalfBoard;
* FullBoard;
* AllInclusive;

### Class Diagram

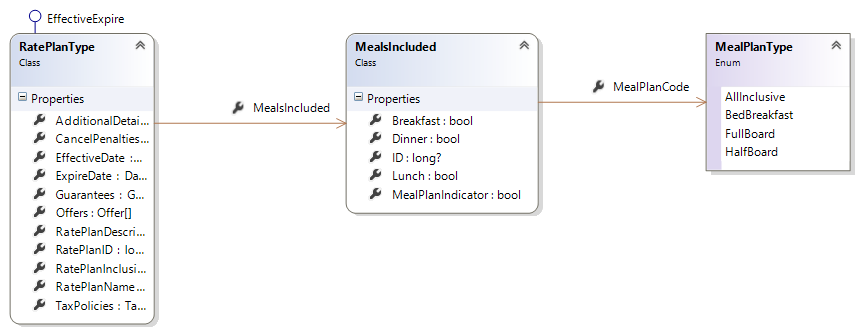


Figure 6 - Meal plan class diagram

### Example

<MealsIncluded>

<Breakfast>true</Breakfast>

<Dinner>false</Dinner>

<ID>2443</ID>

<Lunch>false</Lunch>

<MealPlanCode>BedBreakfast</MealPlanCode>

<MealPlanIndicator>true</MealPlanIndicator>

</MealsIncluded>

## Offers

Offer is sent as parameterized object.

The Total price includes any offer that rate can have.

The detailed info about offers are in Offers Object.

An offer can be available for just a period of time. If this period is not defined then the offer is always available if it is associated in rate.

Offer can be cumulative or non-cumulative:

1. **Cumulative offers** – When offers are cumulative the value for total price will apply one offer to the total price then you have a new price named n2. When you have more compatible offers then total next price will be this offer applied to n2. Example:

Total

Offer01

Offer02

Offer03

Total Price = ((Total – Offer 01) – Offer02) – Offer 03

1. **Non-cumulative offer** – When offer is non-cumulative only that offer can be applied to rate.

**Types of offers**

1. Early Booking – This offer will apply a discount if booking are made up to the number of days earlier than check in date.
2. Last Minute Booking - This offer will apply a discount if booking are made up to the number of days earlier than check in date.
3. Free Nights – This offer will apply when reservation number of night is the same defined in offer rule. This offer can give free nights at the beginning of the stay or ate the end of stay depending on discount pattern.
4. Discount – This offer will apply a direct discount in percentage.
5. Stay Discount – This offer will apply a discount in percentage based on required minimum of nights or/and a maximum number of nights.

### Class Diagram



Figure 7 - Offers class diagram

### Examples

#### Cumulative Offers

Two compatible offers early booking, one with 50 days advance and other with 90 days with advance.

<Offers>

<Offer>

<CompatibleOffer>

<CompatibleOffer>

<IncompatibleOfferIndicator>false</IncompatibleOfferIndicator>

<OfferRPH>

<string>972#0</string>

</OfferRPH>

</CompatibleOffer>

</CompatibleOffer>

<Description>

<Description>EarlyBooking 2</Description>

</Description>

<Discount>

<ChargeUnitCode>PerRoomPerStay</ChargeUnitCode>

<Percent>25</Percent>

</Discount>

<OfferCode>971#0</OfferCode>

<OfferRules>

<OfferRule>

<MinAdvancedBookingOffset>50</MinAdvancedBookingOffset>

</OfferRule>

</OfferRules>

<RPH>971#0</RPH>

</Offer>

<Offer>

<ExtensionData />

<CompatibleOffer>

<CompatibleOffer>

<IncompatibleOfferIndicator>false</IncompatibleOfferIndicator>

<OfferRPH>

<string>971#0</string>

</OfferRPH>

</CompatibleOffer>

</CompatibleOffer>

<Description>

<Description>Early booking 3</Description>

</Description>

<Discount>

<ChargeUnitCode>PerRoomPerStay</ChargeUnitCode>

<Percent>60</Percent>

</Discount>

<OfferCode>972#0</OfferCode>

<OfferRules>

<OfferRule>

<MinAdvancedBookingOffset>90</MinAdvancedBookingOffset>

</OfferRule>

</OfferRules>

<RPH>972#0</RPH>

</Offer>

</Offers>

#### Non-cumulative offers

An offer available only for October with 10 percent discount with 2 days in advance.

<Offers>

<Offer>

<Description>

<Description>Early booking</Description>

</Description>

<Discount>

<ChargeUnitCode>PerRoomPerStay</ChargeUnitCode>

<Percent>10</Percent>

</Discount>

<OfferCode>973#0</OfferCode>

<OfferRules>

<OfferRule>

<DateRestriction>

<DateRestriction>

<ExtensionData />

<End>2014-10-31T00:00:00</End>

<Start>2014-10-01T00:00:00</Start>

</DateRestriction>

</DateRestriction>

<MinAdvancedBookingOffset>2</MinAdvancedBookingOffset>

</OfferRule>

</OfferRules>

<RPH>973#0</RPH>

</Offer>

</Offers>

#### Early booking

An offer available all periods with 10 percent discount with 2 days in advance.

<Offers>

<Offer>

<Description>

<Description>Early booking</Description>

</Description>

<Discount>

<ChargeUnitCode>PerRoomPerStay</ChargeUnitCode>

<Percent>10</Percent>

</Discount>

<OfferCode>975#0</OfferCode>

<OfferRules>

<OfferRule>

<MinAdvancedBookingOffset>2</MinAdvancedBookingOffset>

</OfferRule>

</OfferRules>

<RPH>973#0</RPH>

</Offer>

</Offers>

#### Last Minute

Last minute offer with 10% discount if booking is made 5 days max offset advance booking from check-in date.

<Offers>

<Offer>

<Description>

<Description>Last Minute</Description>

</Description>

<Discount>

<ChargeUnitCode>PerRoomPerStay</ChargeUnitCode>

<Percent>10</Percent>

</Discount>

<OfferCode>974#0</OfferCode>

<OfferRules>

<OfferRule>

<MaxAdvancedBookingOffset>5</MaxAdvancedBookingOffset>

</OfferRule>

</OfferRules>

<RPH>974#0</RPH>

</Offer>

</Offers>

#### Free nights

Free nights offer for bookings with 5 nights. Offer for last 3 nights.

<Offers>

<Offer>

<Description>

<Description>Free nights</Description>

</Description>

<Discount>

<ChargeUnitCode>PerRoomPerStay</ChargeUnitCode>

<DiscountPattern>00111</DiscountPattern>

<NightsDiscounted>3</NightsDiscounted>

<NightsRequired>5</NightsRequired>

</Discount>

<OfferCode>975#0</OfferCode>

<RPH>975#0</RPH>

</Offer>

</Offers>

#### Discount

Applied 10% of discount that can only be applied from 2015-09-13 to 2015-09-19

<Offers>

<Offer>

<Description>

<Description>Discount</Description>

<Language>en</Language>

</Description>

<Discount>

<ChargeUnitCode>PerRoomPerStay</ChargeUnitCode>

<Percent>10</Percent>

</Discount>

<OfferCode>11287#0</OfferCode>

<OfferRules>

<OfferRule>

<DateRestriction>

<DateRestriction>

<End>2015-09-19T00:00:00</End>

<Start>2015-09-13T00:00:00</Start>

</DateRestriction>

</DateRestriction>

</OfferRule>

</OfferRules>

<RPH>11287#0</RPH>

</Offer>

</Offers>

#### Stay Discount – Required minimum of night for discount

Discount of 25 % for reservation with 5 nights of minimum length of stay.

<Offers>

<Offer>

<Description>

<Description>Stay discount min nights</Description>

<Language>en</Language>

</Description>

<Discount>

<ChargeUnitCode>PerRoomPerStay</ChargeUnitCode>

<Percent>25</Percent>

</Discount>

<OfferCode>11288#0</OfferCode>

<OfferRules>

<OfferRule>

<LengthsOfStay>

<ArrivalDateBased>true</ArrivalDateBased>

<LengthsOfStay>

<LengthOfStay>

<MinMaxMessageType>SetMinLOS</MinMaxMessageType>

<Time>5</Time>

<TimeUnit>Day</TimeUnit>

</LengthOfStay>

</LengthsOfStay>

</LengthsOfStay>

</OfferRule>

</OfferRules>

<RPH>11288#0</RPH>

</Offer>

</Offers>

#### Stay Discount – Required minimum of nights and no more than maximum of nights to get discount

Discount of 25 % for reservation with 3 nights of minimum length of stay and 7 nights of maximum length of stay.

<Offers>

<Offer>

<Description>

<Description>Stay discount min/max nights</Description>

<Language>en</Language>

</Description>

<Discount>

<ChargeUnitCode>PerRoomPerStay</ChargeUnitCode>

<Percent>30</Percent>

</Discount>

<OfferCode>11289#0</OfferCode>

<OfferRules>

<OfferRule>

<LengthsOfStay>

<ArrivalDateBased>true</ArrivalDateBased>

<LengthsOfStay>

<LengthOfStay>

<MinMaxMessageType>SetMinLOS</MinMaxMessageType>

<Time>3</Time>

<TimeUnit>Day</TimeUnit>

</LengthOfStay>

<LengthOfStay>

<MinMaxMessageType>SetMaxLOS</MinMaxMessageType>

<Time>7</Time>

<TimeUnit>Day</TimeUnit>

</LengthOfStay>

</LengthsOfStay>

</LengthsOfStay>

</OfferRule>

</OfferRules>

<RPH>11289#0</RPH>

</Offer>

</Offers>

## Included Extras

Included extras are sent in RatePlanInclusion. This extras are included in price. It’s very important to send back this object in reservation so the service can fill correct data on reservation.

### Class Diagram

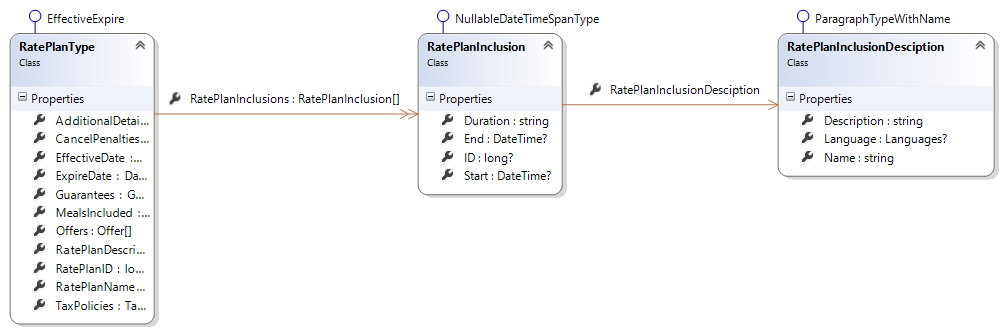


Figure 8 - Rate Plan inclusions class diagram

### Example

<RatePlanInclusions>

<RatePlanInclusion>

<End>2014-05-31T00:00:00</End>

<ID>2444</ID>

<RatePlanInclusionDesciption>

<Description>French 1980</Description>

<Language>en</Language>

<Name>Champagnee</Name>

</RatePlanInclusionDesciption>

<Start>2014-05-01T00:00:00</Start>

</RatePlanInclusion>

</RatePlanInclusions>

## Additional Extras

Additional extras are sent apart from rate. The partner can choose what items to apply.

The total price on reservation should not include the total value of service. Our web service will make the calculation.

Just need to sent back the object and inform the quantity. The quantity only applies to Per Stay prices. The partner can scheduled the Service by sending TPA\_Extensions.

To know what service can be applied to rate room, the service RPH should be used.

### Class Diagram

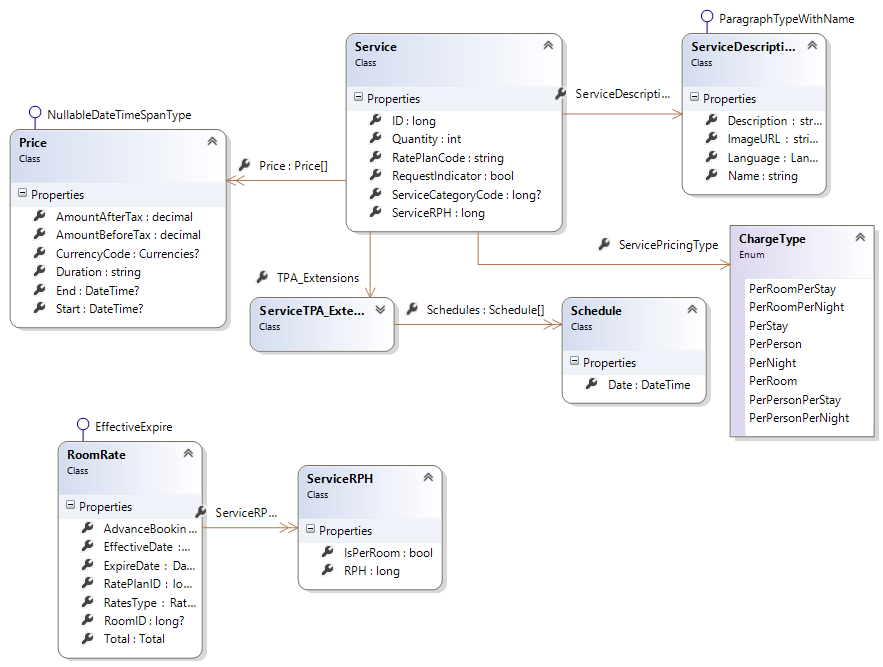


Figure 9 - Additional Extras class diagram

### Example

<RoomRate>

<RatePlanID>12274</RatePlanID>

<RoomID>5825</RoomID>

Only Services with RPH: 2444 and 2445 can be applied to rate 12274 and room 5825

<ServiceRPHs>

<ServiceRPH>

<IsPerRoom>true</IsPerRoom>

<RPH>2444</RPH>

</ServiceRPH>

<ServiceRPH>

<IsPerRoom>true</IsPerRoom>

<RPH>2445</RPH>

</ServiceRPH>

</ServiceRPHs>

</RoomRate>

<Services>

<Service>

<ID>2443</ID>

Service with RPH: **2443 cannot be applied** to rate 12274 and room 5825

<Price>

<Price>

<AmountAfterTax>10.00</AmountAfterTax>

<AmountBeforeTax>10.00</AmountBeforeTax>

<CurrencyCode>EUR</CurrencyCode>

</Price>

</Price>

<ServiceCategoryCode>1</ServiceCategoryCode>

<ServiceDescription>

<Language>en</Language>

<Name>Pequeno Almoçoo</Name>

</ServiceDescription>

<ServicePricingType>PerPersonPerNight</ServicePricingType>

<ServiceRPH>2443</ServiceRPH>

</Service>

Service with RPH: 2444 can be applied to rate 12274 and room 5825

<Service>

<ID>2444</ID>

<Price>

<Price>

<AmountAfterTax>400.00</AmountAfterTax>

<AmountBeforeTax>400.00</AmountBeforeTax>

<CurrencyCode>EUR</CurrencyCode>

</Price>

</Price>

<ServiceDescription>

<Description>Fraçês de 1980</Description>

<ImageURL>http://media.omnibees.loc/Images/1814/Extras/22709.png</ImageURL>

<Language>en</Language>

<Name>Champagnee</Name>

</ServiceDescription>

<ServicePricingType>PerRoomPerStay</ServicePricingType>

<ServiceRPH>2444</ServiceRPH>

</Service>

Service with RPH: 2445 can be applied to rate 12274 and room 5825

<Service>

<ID>2445</ID>

<Price>

<Price>

<AmountAfterTax>60.00</AmountAfterTax>

<AmountBeforeTax>60.00</AmountBeforeTax>

<CurrencyCode>EUR</CurrencyCode>

<End xsi:nil="true" />

<Start xsi:nil="true" />

</Price>

</Price>

<ServiceCategoryCode>2</ServiceCategoryCode>

<ServiceDescription>

<ExtensionData />

<Description>Half board</Description>

<Language>en</Language>

<Name>Half board</Name>

</ServiceDescription>

<ServicePricingType>PerRoomPerNight</ServicePricingType>

<ServiceRPH>2445</ServiceRPH>

</Service>

</Services>

## Payment Policies

### For OTA\_HotelAvailRS (Object used to search availability and prices)

In version previous than 1.8 allowed payment types was sent in GuaranteesAcceptedType, in RatePlanType.

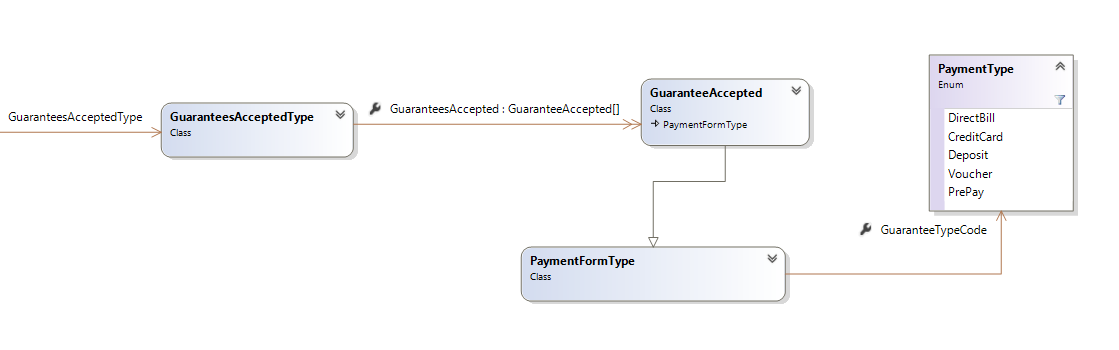


Figure 10 - Payment methods before version 1.8 class diagram

For version 1.8 Payment policies are now sent in PaymentPolicies object in RatePlan object. If partner change from old version this change must be implemented. If partner do not change do version 1.8 not action is needed.

#### Class Diagram

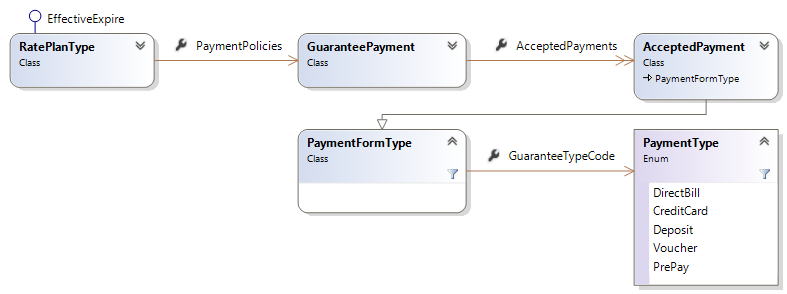


Figure 11 - Payment methods after version 1.8 class diagram

#### Example

<PaymentPolicies>

<AcceptedPayments>

<AcceptedPayment>

<GuaranteeTypeCode>Voucher</GuaranteeTypeCode>

</AcceptedPayment>

<AcceptedPayment>

<GuaranteeTypeCode>CreditCard</GuaranteeTypeCode>

</AcceptedPayment>

</AcceptedPayments>

</PaymentPolicies>

### For OTA\_HotelResRQ

After search, partner can send one of the payment type sent in OTA\_HotelAvailRS for rateplan that will be used in reservation request.

Note: The only payment allow for reservation is the payments returned on search. For bookings with more than one room with different rate plans, the payment that will be used on reservation has to be allowed in all rate plans. Please make sure that payment type used for reservation is allowed in all rate plans.

The Object used for sending payment type in reservation is

HotelReservation.ResGlobalInfo.Guarantee.GuaranteesAcceptedType.GuaranteesAccepted. GuaranteeAccepted

Example:

<HotelReservation>

<ResGlobalInfo>

<Guarantee>

<GuaranteesAcceptedType>

<GuaranteesAccepted>

<GuaranteeAccepted>

<GuaranteeTypeCode>CreditCard</GuaranteeTypeCode>

<PaymentCard>

<CardCode>Visa</CardCode>

<CardHolderName>Consumer</CardHolderName>

<CardNumber>4111111111111111</CardNumber>

<EffectiveDate>2014-01-01T00:00:00</EffectiveDate>

<ExpireDate>2016-06-01T00:00:00</ExpireDate>

<SeriesCode>737</SeriesCode>

</PaymentCard>

<RPH>0</RPH>

</GuaranteeAccepted>

</GuaranteesAccepted>

</GuaranteesAcceptedType>

</Guarantee>

### Codes that can be returned on search

|  |  |  |  |
| --- | --- | --- | --- |
| OTA Code List Name | Code Value | Code Name | Where Used |
| PMT | 2 | DirectBill | GuaranteeTypeCode |
| PMT | 5 | CreditCard | GuaranteeTypeCode |
| PMT | 8 | Deposit | GuaranteeTypeCode |
| PMT | 3 | Voucher | GuaranteeTypeCode |
| PMT | 4 | PrePay | GuaranteeTypeCode |
|  |  |  |  |

Table 2 - Payment types

Please check section 5.3 to know about login types. Payment Methods can be different according to login type.

## Reservation Status

The reservation status of a reservation. The status can be:

|  |  |
| --- | --- |
| Code | Description |
| Requested | The reservation is on “On Request” state, waiting for hotel to approve. |
| Reserved | The reservation is Booked. |
| RequestDenied | The hotel did not approved the “On request” reservation. |
| RequestAccepted | The hotel approved the “On request” reservation. |
| Cancelled | Reservation is cancelled. |
| Waitlisted | Reservation is in pending state and is waiting for some other action. |
| Modified | Reservation is modified and is booked. |

Table 3 - Reservation Status

## Booking on Request

When search is in a data range that there is no available allotment, hotel can decide if accept the booking or not.

By default search do not return rates on request so partner has to send the indication to return rates on request.

To make a reservation that can be on request the steps are:

* + 1. Search (HotelAvail) indicating the service to return rates that can be on request;
    2. Choose the rate and room and send to reservation system (HotelRes) in a normal request. Partner has to generate a Guid for transaction id. This transaction id will be used on further messages (modify, cancel);
    3. Prepare on web service to receive reservations notifications from Pull Notify system. Optional the partner can call GetBookings to get the current state of booking;

After hotel accept the reservation on request the reservation will be on **RequestAccepted** status;

Optional hotel can denied the booking on request and the reservation will be on **RequestDenied**

* + 1. After reservation in on RequestAccpeted status partner can accepted or ignore booking or ignore:

1. To accepted booking partner has to send a modify request and send the reservation number the transaction id (sent previous in HotelRes, or by getting with GetBookings). The modify will has ResStatus on principal object equals to “Commit”;
2. To ignore the booking simple send a cancel request, no charge will be added because cancelation. When reservation is not on status Booked, Modified, no charge will be considered in cancelation;

### Search (Hotel Avail) including “On Request” rates

In HotelAvail request object the OnRequestInd must be sent true to get on request messages. Note: Hotel must have configured that on request reservation is accepted on Omnibees, otherwise no On request message will be visible.

<ota\_HotelAvailRQ>

...

<d4p1:OnRequestInd>true</d4p1:OnRequestInd>

..

<d4p1:Version>2.0</d4p1:Version>

...

</ota\_HotelAvailRQ>

At HotelAvail response on request can be found at RoomRate Availability Level. The availability status will be OnRequest. The WaningRPH is a reference for Warnings object, it will have a friendly message.

...

<a:RoomStaysType>

<a:RoomStays>

<a:RoomStay>

<a:RoomRates>

<a:RoomRate>

<a:Availability>

<a:RoomRateAvailability>

<a:AvailabilityStatus>OnRequest</a:AvailabilityStatus>

<a:WarningRPH>507</a:WarningRPH>

</a:RoomRateAvailability>

</a:Availability>

...

</a:RoomRate>

</a:RoomRates>

...

</a:RoomStay>

</a:RoomStays>

</a:RoomStaysType>

...

<a:WarningsType>

<a:Warnings>

<a:Warning>

<a:Code>507</a:Code>

<a:Language>en</a:Language>

<a:RPH>507</a:RPH>

<a:ShortText>Room type on request</a:ShortText>

<a:Type>3</a:Type>

</a:Warning>

</a:Warnings>

</a:WarningsType>

### HotelRes Status

For sending a reservation no special item is need for on request booking. Partner will send TransactionActionType Hold for HotelReservationObject and a TransactionIndetifier. Transaction Identifier must be a GUID and will be used for modify or cancel a reservation.

<d4p1:TransactionIdentifier>e7d841e8-e687-4aa8-b36f-87d6266a70b9</d4p1:TransactionIdentifier>

<d4p1:HotelReservationsType>

<d4p1:HotelReservations>

<d4p1:HotelReservation>

...

<d4p1:ResStatus>

<d4p1:TransactionActionType>Hold</d4p1:TransactionActionType>

</d4p1:ResStatus>

...

</d4p1:HotelReservation>

</d4p1:HotelReservations>

</d4p1:HotelReservationsType>

In response partner will receive a Requested status for bookings that are on request.

<d4p1:TransactionIdentifier>e7d841e8-e687-4aa8-b36f-87d6266a70b9</d4p1:TransactionIdentifier>

<d4p1:HotelReservationsType>

<d4p1:HotelReservations>

<d4p1:HotelReservation>

...

<d4p1:ResStatus>

<d4p1:PMS\_ResStatusType>Requested</d4p1:PMS\_ResStatusType>

<d4p1:TransactionActionType>Hold</d4p1:TransactionActionType>

</d4p1:ResStatus>

...

</d4p1:HotelReservation>

</d4p1:HotelReservations>

</d4p1:HotelReservationsType>

### Reservation that is on status RequestAccepted

When hotel accept reservation that is on status RequestAccepted, partner can accept or ignore reservation.

<d4p1:ResStatus>

<d4p1:PMS\_ResStatusType>RequestAccepted</d4p1:PMS\_ResStatusType>

</d4p1:ResStatus>

In this reservation status partner can accept or ignore reservation.

### Accept reservation

To accepted reservation partner has to send a Modify request with Commit in transaction status. The Transaction id sent previous in hotel res must be sent on Commit.

<ota\_HotelResModifyRQ>

<d4p1:EchoToken>e7d541e8-e657-4aa8-b36f-87d6266a8751</d4p1:EchoToken>

<d4p1:HotelResModifiesType>

<d4p1:HotelResModifies>

<d4p1:HotelReservation>

<d4p1:UniqueID>

<d4p1:ID>RES003331-1881</d4p1:ID>

<d4p1:Type>Reservation</d4p1:Type>

</d4p1:UniqueID>

</d4p1:HotelReservation>

</d4p1:HotelResModifies>

</d4p1:HotelResModifiesType>

<d4p1:PrimaryLangID>en</d4p1:PrimaryLangID>

<d4p1:ResStatus>Commit</d4p1:ResStatus>

<d4p1:RetransmissionIndicator>false</d4p1:RetransmissionIndicator>

<d4p1:Target>Test</d4p1:Target>

<d4p1:TimeStamp>2016-01-28T18:30:32.5654219Z</d4p1:TimeStamp>

<d4p1:TransactionIdentifier>e7d841e8-e687-4aa8-b36f-87d6266a70b9</d4p1:TransactionIdentifier>

<d4p1:Version>2.0</d4p1:Version>

</ota\_HotelResModifyRQ>

### Ignore Reservation

To ignore a reservation partner ha to call Cancel Reservation in normal request. The transaction id sent previous in hotel res must be sent on Cancel.

<ota\_CancelRQ>

<d4p1:EchoToken>e7d541e8-e657-4aa8-b36f-87d6266a8751</d4p1:EchoToken>

<d4p1:PrimaryLangID>en</d4p1:PrimaryLangID>

<d4p1:Target>Test</d4p1:Target>

<d4p1:TimeStamp>2016-01-28T00:00:00Z</d4p1:TimeStamp>

<d4p1:TransactionIdentifier>e7d841e8-e687-4aa8-b36f-87d6266a70b9</d4p1:TransactionIdentifier>

<d4p1:UniqueID>

<d4p1:UniqueID>

<d4p1:ID>RES003331-1881</d4p1:ID>

<d4p1:Type>Reservation</d4p1:Type>

</d4p1:UniqueID>

</d4p1:UniqueID>

<d4p1:Verification>

<d4p1:Email>email@visualforma.pt</d4p1:Email>

<d4p1:HotelRef>

<d4p1:HotelCode>1881</d4p1:HotelCode>

</d4p1:HotelRef>

<d4p1:ReservationTimeSpan>

<d4p1:End>2016-01-31T00:00:00</d4p1:End>

<d4p1:Start>2016-01-29T00:00:00</d4p1:Start>

</d4p1:ReservationTimeSpan>

</d4p1:Verification>

<d4p1:Version>2.0</d4p1:Version>

</ota\_CancelRQ>

### Reservation that is on status RequestDenied

When hotel rejects reservation on request reservation goes to status RequestDenied.

<d4p1:ResStatus>

<d4p1:PMS\_ResStatusType> RequestDenied</d4p1:PMS\_ResStatusType>

</d4p1:ResStatus>

## Rates that are not available for sell

From version 2.0 partner can now ask to include rates that are not for sale. This way client can know if can change something to get results for sale. For example Minum Length of stay not ok.

To make a request to include rates that are not for sale, the tag **AvailRatesOnly = false**.

The response will have multiple roomrates. Remember that each Roomstay are for each hotel (in case of multime hotels request), and RoomRate is a combination of RatePlan and Roomtype.

To see that rates is not avail for sale partner has to check Availability Status, and cross reference with warning to get more details.

<a:RoomRate>

<a:Availability>

<a:RoomRateAvailability>

<a:AvailabilityStatus>ClosedOut</a:AvailabilityStatus>

<a:WarningRPH>575</a:WarningRPH>

</a:RoomRateAvailability>

</a:Availability>

</a:RoomRate>

<a:WarningsType>

<a:Warnings>

<a:Warning>

<a:Code>575</a:Code>

<a:Language>en</a:Language>

<a:RPH>575</a:RPH>

<a:ShortText>Closed restriction</a:ShortText>

<a:Type>3</a:Type>

</a:Warning>

</a:Warnings>

</a:WarningsType>

The Possible values for rates that are not for sale are:

|  |  |  |  |
| --- | --- | --- | --- |
| AvailabilityStatus | Warning Code | Waning text | Detail |
| AvailableForSale |  | No Warning |  |
| NoAvailability | 427 | No Rooms Available For Requested Dates | Allotment not available at Hotel |
| OnRequest | 507 | Room Type On Request | Allotment not available at Hotel, but hotel accept on request |
| ClosedOut | 575 | Closed Restriction | At least one day is closed |
| OtherAvailable | 346 | Closed To Arrivals | Check in date not allow reservation |
| OtherAvailable | 563 | Closed To Departure | Check out date not allow reservation |
| UnableToProcess | 51 | Prices Not Yet Released | Prices not loaded in Hotel system |
| Exclusive | 169 | Agent Code Invalid | Agent Code is Invalid or rate is exclusive for agencies |
| Exclusive | 569 | Group Booking Required | Group Code is Invalid or rate is exclusive for groups |
| Exclusive | 536 | Invalid Promotion Code | Promotional code is invalid or rate is exclusive for promotional codes |
| OtherAvailable | 397 | Invalid Number Of Adults | Number of adults cannot go to room |
| OtherAvailable | 138 | Child Numbers Occupancy Mismatch | Number of child’s cannot go to room |
| LOS\_Restricted | 409 | Maximum Length Of Stay Restriction | Maximum Length of stay is less than total of days |
| OtherAvailable | 142 | Occupancy Rules Not Fulfilled | Number of persons is higher than expected |
| LOS\_Restricted | 411 | Minimum Length Of Stay Restriction | Minimum length of stay is higher than total of days |
| RatePlanClosed | 441 | Requested Rate Not Available For Entire Stay | Rate is not active at the hotel |
| RatePlanClosed | 404 | Invalid Start End Date Combination | Rate is not for sale with at this time |
| LOS\_Restricted | 109 | Cannot Book Arrival Too Close | Release days make check in too close for book |
| LOS\_Restricted | 560 | Stay Not Compliant With Availability Restrictions | Stay is not the same number of days that hotel expects |

Table 4 - Availability status for not available rates

# Messages

**Files are in a zip at the attachments section.**

## Hotel Search

### Description

This messages provides information about active hotels. This information includes Hotel information, address, contacts, rating, check-in and check-out policies, geo location, room amenities and hotel amenities.

This method can response requests with paging or without pagging. Section 6.1.2.1 and Section 6.1.2.2. To use paging simple fill the page number and page size you need. If no need to use paging page number and page size has to passed as null.

When use paging you can know if more hotels are available by checking MoreIndicator property.

To count how many hotels are return just check array length on PropertiesType.Properties.Property[]. If you are paging the return hotel are you page size if moreindicator is true.

For search hotel in a city in a radius the object Radius need to be filled. Section 6.1.2.3

### OTA\_HotelSearchRQ

#### Schema

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Level | Occurrence | Element | Property | Mandatory (M)/ Optional (O) | Sample Usage |
| 0 | 1 | OTA\_HotelAvailRQ |  |  |  |
|  |  |  | Echotoken | M | c2ab0184-e9ac-4eb7-a9e1-7f1e8df2b66f |
|  |  |  | MaxResponses | O | 50 |
|  |  |  | PageNumber | O | 1 |
|  |  |  | PrimaryLanID | O | En |
|  |  |  | TimeStamp | M | 2014-06-23T15:04:16.0294414Z |
|  |  |  | Target | M | Test |
|  |  |  | Version | M | 1 |
|  |  |  |  |  |  |
| 1 | 1 | HotelSearchCriteria |  | M |  |
| 2 | 1 | HotelSearchCriterion |  | M |  |
| 3 | 1 | HotelRef |  | M |  |
|  |  |  | ChainCode | O | null |
|  |  |  | HotelCode | M | 1053 |
| 3 | 0..1 | AddressType |  | O |  |
|  |  |  | CountryCode | O | 40 |
|  |  |  | StateProvCode | O | 6362053 |
|  |  |  | CityCode | O | 2851556 |
|  |  |  | ZoneCode | O | 2254852 |
| 3 | 0..1 | Location |  | O |  |
|  |  |  | LocationCode | M | FAO |
|  |  |  | CodeContext | M | IATA |

Table 5 - OTA\_HotelSearchRQ Schema

#### General Rules

1. Echotoken – A Guid must be sent in every request. Must be unique per request;
2. When using paging must sent the current page and page size
   1. PageNumber – Indicate the current page requested
   2. Maxresponses – indicate the page size
3. PrimaryLangID – Indicates the language that responses should be sent
4. Timestamp – Must send allways the now date in utc time.
5. Version – Version of the service.
6. HotelRef
   1. HotelCode – Must send hotelcode if only need rates for one specific hotel;
   2. ChainCode – Used to specify hotels chain code.
7. AddressType
   1. CountryCode – The country uid can be obtained in GetCitiesList method
   2. StatesProvCode – The stateprovcode can be obtained in GetStateList method by passing countryCode
   3. CityCode – The citycode can be obtained in GetCities method by CountryCode and StateSearchCode.
   4. ZoneCode – The zonecode can be obtained in GetZonesList by CountryUid, StateSearchCode, ZoneSearchCode, CitiesDataUid
8. Location – Used to search hotels by an IATA code

#### Class Diagram

##### Image

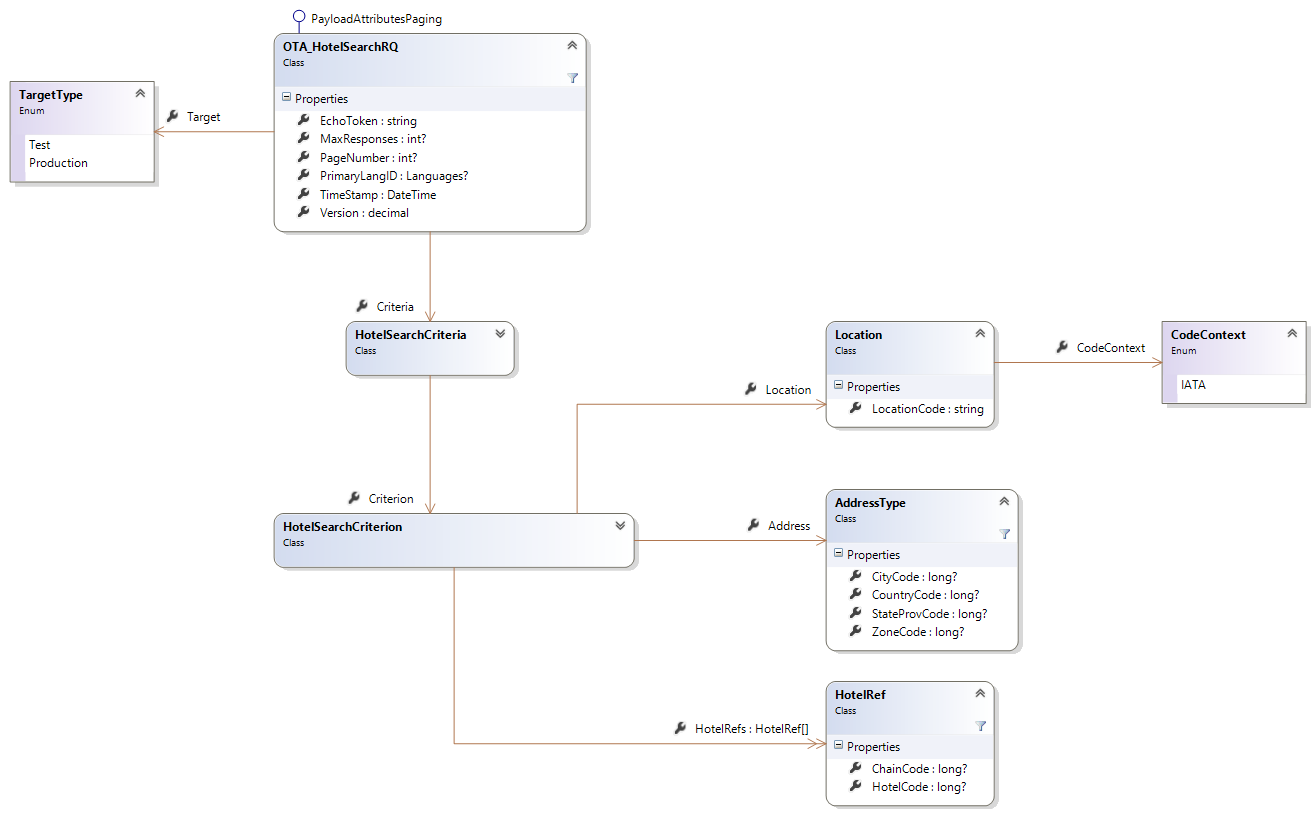


Figure 12 - OTA\_HotelSearchRQ class diagram

#### XML Requests

##### Search hotels with paging

<OTA\_HotelSearchRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<Criteria>

<Criterion />

</Criteria>

<EchoToken>aaac089f-4c3f-4a8a-85d6-2192a198574b</EchoToken>

<MaxResponses>50</MaxResponses>

<PageNumber>1</PageNumber>

<PrimaryLangID>en</PrimaryLangID>

<Target>Test</Target>

<TimeStamp>2014-03-26T09:58:13.6386147Z</TimeStamp>

<Version>1</Version>

</OTA\_HotelSearchRQ>

##### Search hotels without paging

<OTA\_HotelSearchRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<Criteria>

<Criterion />

</Criteria>

<EchoToken>aaac089f-4c3f-4a8a-85d6-2192a198574b</EchoToken>

<MaxResponses xsi:nil="true" />

<PageNumber xsi:nil="true" />

<PrimaryLangID>en</PrimaryLangID>

<Target>Test</Target>

<TimeStamp>2014-03-26T09:58:13.6386147Z</TimeStamp>

<Version>1</Version>

</OTA\_HotelSearchRQ>

##### Search hotel by city UID

<OTA\_HotelSearchRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<Criteria>

<Criterion>

<Address>

<CityCode>509072</CityCode>

<CountryCode>157</CountryCode>

<StateProvCode>3017993</StateProvCode>

</Address>

</Criterion>

</Criteria>

<EchoToken>7955c0a0-2672-4690-ba8e-e73e2e64857b</EchoToken>

<MaxResponses xsi:nil="true" />

<PageNumber xsi:nil="true" />

<PrimaryLangID>en</PrimaryLangID>

<Target>Test</Target>

<TimeStamp>2014-03-26T11:07:56.1859062Z</TimeStamp>

<Version>1</Version>

</OTA\_HotelSearchRQ>

##### Search an hotel by IATA Code

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

<OTA\_HotelSearchRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<Criteria>

<Criterion>

<Location>

<CodeContext>IATA</CodeContext>

<LocationCode>FAO</LocationCode>

</Location>

</Criterion>

</Criteria>

<EchoToken>5bc9fec9-ff9b-4dd0-bb50-926211ce85fd</EchoToken>

<MaxResponses>50</MaxResponses>

<PageNumber>1</PageNumber>

<PrimaryLangID>en</PrimaryLangID>

<Target>Test</Target>

<TimeStamp>2013-07-10T13:23:34.5375243Z</TimeStamp>

<Version>1</Version>

</OTA\_HotelSearchRQ>

##### Search by Country/state/city code/Zone code

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

<OTA\_HotelSearchRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<Criteria>

<Criterion>

<Address>

<CityCode>509072</CityCode>

<CountryCode>157</CountryCode>

<StateProvCode>3017993</StateProvCode>

<ZoneCode>5588584</ZoneCode>

</Address>

</Criterion>

</Criteria>

<EchoToken>29523f10-eef8-4811-8cd4-85b2eb09c2ca</EchoToken>

<MaxResponses>50</MaxResponses>

<PageNumber>1</PageNumber>

<PrimaryLangID>en</PrimaryLangID>

<Target>Test</Target>

<TimeStamp>2013-07-10T13:30:18.6129035Z</TimeStamp>

<Version>1</Version>

</OTA\_HotelSearchRQ>

##### Search all hotels in database

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

<OTA\_HotelSearchRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<Criteria>

<AvailableOnlyIndicator>false</AvailableOnlyIndicator>

<Criterion>

</Criterion>

</Criteria>

<EchoToken>a5dfe129-f72b-492c-b5fc-70d63b4a7201</EchoToken>

<MaxResponses>50</MaxResponses>

<PageNumber>1</PageNumber>

<PrimaryLangID>en</PrimaryLangID>

<Target>Test</Target>

<TimeStamp>2014-10-13T10:38:39.3646409Z</TimeStamp>

<Version>1.0</Version>

</OTA\_HotelSearchRQ>

##### Search only hotels that have rates to sell

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

<OTA\_HotelSearchRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<Criteria>

<AvailableOnlyIndicator>true</AvailableOnlyIndicator>

<Criterion>

</Criterion>

</Criteria>

<EchoToken>a5dfe129-f72b-492c-b5fc-70d63b4a7201</EchoToken>

<MaxResponses>50</MaxResponses>

<PageNumber>1</PageNumber>

<PrimaryLangID>en</PrimaryLangID>

<Target>Test</Target>

<TimeStamp>2014-10-13T10:38:39.3646409Z</TimeStamp>

<Version>1.0</Version>

</OTA\_HotelSearchRQ>

### OTA\_HotelSearchRS

#### Schema

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Level | Occurrence | Element | Property | Mandatory (M)/ Optional (O) | Sample Usage |
| 0 | 1 | OTA\_HotelAvailRS |  |  |  |
|  |  |  | Echotoken | M | c2ab0184-e9ac-4eb7-a9e1-7f1e8df2b66f |
|  |  |  | MaxResponses | O | 50 |
|  |  |  | PageNumber | O | 1 |
|  |  |  | PrimaryLanID | O | En |
|  |  |  | TimeStamp | M | 2014-06-23T15:04:16.0294414Z |
|  |  |  | Target | M | Test |
|  |  |  | Version | M | 1 |
|  |  |  |  |  |  |
| 1 | 1 | HotelSearchCriteria |  | M | Same as request |
|  |  |  |  |  |  |
| 1 | 0.1 | PropertiesType |  | O |  |
|  |  |  | MoreIndicator | M | true |
| 2 | 1..N | Property |  | M |  |
|  |  |  | CurrencyCode | O | EUR |
|  |  |  |  |  |  |
| 3 | 1 | HotelRef |  | M |  |
|  |  |  | BrandName | O | Null |
|  |  |  | ChainCode | O | OB |
|  |  |  | HotelCode | M | 1814 |
|  |  |  | HotelName | M | BigBullEuro |
|  |  |  |  |  |  |
| 3 | 0..1 | Award |  | O |  |
|  |  |  | Rating | M | 4 |
|  |  |  |  |  |  |
| 3 | 0..1 | AddressType |  | O |  |
|  |  |  | AddressLine | O | Benavente |
|  |  |  | CityCode | O | 2851556 |
|  |  |  | CityName | O | Sao Paulo |
|  |  |  | CountryCode | O | 40 |
|  |  |  | CountryName | O | Brasil |
|  |  |  | PostalCode | O | 9125-067 |
|  |  |  | StateProv | O | Sao Paulo |
|  |  |  | StateProvCode | O | 6362053 |
|  |  |  |  |  |  |
| 3 | 0..1 | Policy |  | O |  |
|  |  |  | CheckInTime | M | 9:00 |
|  |  |  | CheckOutTime | M | 12:00 |
|  |  |  |  |  |  |
| 3 | 0..1 | PositionType |  | O |  |
|  |  |  | Latitude | M | 38.76306 |
|  |  |  | Longitude | M | -8.85722 |
|  |  |  |  |  |  |
| 3 | 0..1 | ContactNumbersType |  | O |  |
| 4 | 1..3 | ContactNumbers |  | M |  |
|  |  |  | ShareSyncInd | M | No |
|  |  |  | Share ISOCountry Ind | M | No |
|  |  |  | PhoneNumber | M | +351895478954 |
|  |  |  | PhoneTechType | M | Fax (PPT codes) |
|  |  |  | PhoneLocationType | M | Allways PropertyDirect |
|  |  |  |  |  |  |
| 3 | 0..1 | VendorMessageType |  | O |  |
| 4 | 1..N | VendorMessage |  | M |  |
|  |  |  | Description | M | Fica situado na costa Sul da Ilha da Madeira, |
|  |  |  |  |  |  |
| 3 | 0..1 | RateRange |  | O |  |
|  |  |  | MinRate | M | 80 |
|  |  |  | CurrencyCode | M | EUR |
|  |  |  |  |  |  |
| 1 | 0..1 | Sucess |  | O |  |
|  |  |  |  |  |  |
| 1 | 0..1 | WarningsType |  | O |  |
| 2 | 1..N | Warning |  | M |  |
|  |  |  | Code | M | 450 |
|  |  |  | Language | M | En |
|  |  |  | ShortText | M | Warning Text |
|  |  |  |  |  |  |
| 1 | 0..1 | ErrorsType |  | O |  |
| 2 | 1..N | Error |  | M |  |
|  |  |  | Code | M | 448 |
|  |  |  | Language | M | en |
|  |  |  | ShortText | M | System error: Internal system error. Please contact techincal support at integration.team@omnibees.com { echoToken = 57fde0db-32ed-40c5-bf88-e19a7601472a } |

Table 6 - OTA\_HotelSearchRS schema

General Rules

1. Echotoken – A Guid must be sent in every request. Must be unique per request;
2. When using paging must sent the current page and page size
   1. PageNumber – Indicate the current page requested
   2. Maxresponses – indicate the page size
3. PrimaryLangID – Indicates the language that responses should be sent
4. Timestamp – Must send allways the now date in utc time.
5. Version – Version of the service;
6. PropertiesType
   1. MoreIndicator – Indicates that more items are available;
7. Property
   1. CurrencyCode - The code specifying a monetary unit. Use ISO 4217, three alpha code.
   2. HotelRef
      1. BrandName – The name of a brand of hotels (e.g., Courtyard, Hampton Inn).
      2. HotelName – A property name of the hotel;
      3. HotelCode – Must send hotelcode if only need rates for one specific hotel;
      4. ChainCode – Used to specify hotels chain code.
   3. Award
      1. Rating - The actual award or rating received by the hotel facility.
   4. AddressType
      1. AddressLine – Hotel address;
      2. CityCode – City code;
      3. CityName – City (e.g., Dublin), town, or postal station;
      4. CountryCode – Country code;
      5. CountryName – Country name;
      6. PostalCode – Post Office code number;
      7. StateProv – State code;
      8. StateProvCode – State or Province name;
   5. Policy
      1. CheckInTime – The checkin time required by this hotel for a room stay.
      2. CheckOutTime – The checkout time required by this hotel for a room stay.
   6. PositionType
      1. Latitude – The measure of the angular distance on a meridian north or south of the equator;
      2. Longitude – The measure of the angular distance on a meridian east or west of the prime meridian;
   7. TelephoneInfo
      1. ShareSyncInd – Permission for sharing data for synchronization of information held by other travel service providers;
      2. ShareMarketInd – Permission for sharing data for marketing purposes;
      3. PhoneNumber – Telephone number assigned to a single location;
      4. PhoneTechType – Indicates type of technology associated with this telephone number;
      5. PhoneLocationType – Describes the location of the phone.
   8. VendorMessage
      1. Description - Description in plain text format.
   9. RateRange – Used to specify the best price in hotel
      1. MinRate – Decimal with value of min rate found
      2. CurrencyCode - An ISO 4217 alpha character code that specifies a monetary unit
8. Success – Used to specify that communication from service was handled without errors. The success tag can be return with Warnings.
9. Warnings – Used to specify the request cannot be satisfied because of business rules. The description and code should be used to detect the problem.
10. Errors – Used to specify that the request cannot be satisfied because of an internal error. Echo token should be send to Omnibees support to check what the problem was.

#### Class Diagram

##### Zip



#### XML Response example

##### Single hotel response



##### Multiple hotel response



## Availability and prices Service

### Description

This messages provides information about availabilities and prices about one specific hotel by hotel id, or about a range of hotels search with detailed criteria such as IATA code, Country code, state code, city code, rating, etc. The detail information to check you to filter and how to get codes is on Messages section.

This method can response requests with **paging or without paging**. **Section 6.2.2.1** and **Section 6.2.2.2**. To use paging, simple fill the page number and page size you need. If don´t need to use paging, the tags page number and page size has to passed as null.

When paging is used you can know if more hotels are available by checking RoomStaysType.MoreIndicator property.

To count how many rates are returned just check array length on RoomStaysType. RoomStays.

For serch **best rates for hotels** send property bestonly = true.

Notes:

1. When childs is included in search child ages must be provided.
2. Country/states/cities codes can be obtained using service GetCountryList, GetStateList and GetCityList.

### OTA\_HotelAvailRQ

#### Schema

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Level | Occurrence | Element | Property | Mandatory (M)/ Optional (O) | Sample Usage |
| 0 | 1 | OTA\_HotelAvailRQ |  |  |  |
|  |  |  | MaxResponses | O | 50 |
|  |  |  | PageNumber | O | 1 |
|  |  |  | Echotoken | M | c2ab0184-e9ac-4eb7-a9e1-7f1e8df2b66f |
|  |  |  | TimeStamp | M | 2014-06-23T15:04:16.0294414Z |
|  |  |  | Target | M | Test |
|  |  |  | Version | M | 1 |
|  |  |  | PrimaryLanID | O | En |
|  |  |  | AvailRatesOnly | O | True |
|  |  |  | BestOnly | O | True, False |
|  |  |  | OnRequestInd | O | True |
|  |  |  | IsModify | O | true |
|  |  |  | RequestedCurrency | O | true |
|  |  |  |  |  |  |
| 1 | 0..1 | POSType |  | O |  |
| 2 | 1 | Source |  | M |  |
|  |  |  | PseudoCityCode | O | TESTP |
| 3 | 0..1 | RequestorID |  | O |  |
|  |  |  | ID | M | TESTP|ABB |
|  |  |  | Type | M | UID Code (Only 3 – CorporationRepresentative is valid) |
|  |  |  |  |  |  |
| 1 | 1 | HotelSearchCriteria |  | M |  |
| 2 | 1 | HotelSearchCriterion |  | M |  |
| 3 | 1..n | HotelRef |  | M |  |
|  |  |  | HotelCode | M | 1053 |
|  |  |  | ChainCode | O | 125 |
| 3 | 0..1 | AddressType |  | O |  |
|  |  |  | CountryCode | O | 40 |
|  |  |  | StateProvCode | O | 6362053 |
|  |  |  | CityCode | O | 2851556 |
|  |  |  | ZoneCode | O | 2554886 |
| 3 | 1 | StayDateRange |  | M |  |
|  |  |  | End | M | 2014-10-30 |
|  |  |  | Start | M | 2014-10-28 |
| 3 | 1 | RoomStayCandidatesType |  | M |  |
| 4 | 1..10 | RoomStayCandidate |  | M |  |
|  |  |  | Quantity | M | 1 |
|  |  |  | RPH | M | Must be a unique int |
|  |  |  | BookingCode | O | Res125/1 |
| 5 | 1 | GuestCountsType |  | M |  |
| 6 | 1…20 | GuestCount |  | M |  |
|  |  |  | Age | M/O | Mandatory for child/infant |
|  |  |  | Count | M | 2 |
|  |  |  | AgeQualifyingCode | M | AQC code |
| 3 | 0..1 | PositionType |  | O |  |
|  |  |  | Latitude | M | 38.71028 |
|  |  |  | Longitude | M | -9.14306 |
| 3 | 0..1 | Location |  | O |  |
|  |  |  | LocationCode | M | FAO |
|  |  |  | CodeContext | M | IATA |
| 3 | 0..1 | Radius |  | O |  |
|  |  |  | DistanceMax | M | 30 |
|  |  |  | UnitOfMeasureCodeType | M | UOC, Kilometers |
| 3 | 0..1 | Award |  | O |  |
|  |  |  | Rating | M | 4 |
| 3 | 0..1 | GetPricesPerGuest |  | O | True or False |
| 3 | 0..50 | RatePlanCandidatesType |  | O |  |
| 4 | 1 | RatePlanCandidate |  | M |  |
|  |  |  | GroupCode | O | 98763 |
|  |  |  | PromotionCode | O | 5g547 |
|  |  |  | RatePlanType | O | 1 |
| 3 | 0..1 | ProfilesType |  | O |  |
| 4 | 1 | ProfileInfo |  | M |  |
| 5 | 1 | Unique ID |  | M |  |
|  |  |  | ID | M | ABB |
|  |  |  | Type |  | UID Code |
|  |  |  |  |  |  |
| 3 | 0..1 | TPA\_Extensions |  |  |  |
|  |  |  | AmountIncludingMarkup | O | True |
|  |  |  | AmountNotIncludingMarkup | O | False |
|  |  |  | AmountIsPackageRates | O | True |
| 4 | 0..1 | MultimediaObjects |  |  |  |
|  |  |  | SendData |  | True |

Table 7 - OTA\_HotelAvailRQ Schema

#### General Rules

1. Echotoken – A Guid must be sent in every request. Must be unique per request;
2. When using paging must sent the current page and page size
   1. PageNumber – Indicate the current page requested
   2. Maxresponses – indicate the page size
3. PrimaryLangID – Indicates the language that responses should be sent;
4. Timestamp – Must send allways the now date in utc time;
5. Version – Version of the service;
6. IsModify – When true will make a search based on exitent reservation;
7. BestOnly – When true the response will have only the best available rate for each hotel;
8. AvailRatesOnly – When true the response will include only rates that are available for sell. When false the response will include all rates, including the rates that are not available for sell. Default value is false;
9. OnRequestInd – When true the response will include rates On Request, that can book with no allotment, but the hotel has to approved before reservation go to book state;
10. RequestedCurrency – Used to specify to return only rates that is on requested currency, ISO 4217, three alpha code;
11. POS
    1. Source – Will have the information about agency PCC. The GetAgencies Method can be used to get all agencies codes.
    2. RequestorID (only will be used if the request is from an agency post in a company)
       1. Type – Must be allways CorporationRepresentative
       2. ID – Must be the agency office pcc.
12. HotelRef
    1. HotelCode – Must send hotelcode if only need rates for one specific hotel;
    2. ChainCode – Used to search only hotels from a specified chain. Code must be supplied from Omnibees team.
13. AddressType
    1. CountryCode – The country uid can be obtained in GetCitiesList method
    2. StatesProvCode – The stateprovcode can be obtained in GetStateList method by passing countryCode
    3. CityCode – The citycode can be obtained in GetCities method by CountryCode and StateSearchCode.
    4. ZoneCode – The zonecode can be obtained in GetZonesList by CountryUid, StateSearchCode, ZoneSearchCode, CitiesDataUid
14. StaydateRage
    1. Start – Indicate the checkin date
    2. End – Indicate the checkout date
15. RoomStayCandidate
    1. RPH – This tag (reference place holder) must be sent as a unique integer identifier per room. Each RoomstayCandidate must have a unique RPH. This RPH will be returned on response as RoomStayCandidateRPH to identify the price for the room.
    2. Quantiy – Will be used to identify how many rooms with this roomstaycandidate are required. **The response always has the total price for the quantity passed**.
    3. GuestCount - Used to specify the count of guests by ages
       1. Age – Age is required for child and infant. For adults is optional
       2. Count – Number of guests for the same age.
       3. Type – Adult, Child Or infant
    4. BookingCode – Used to pass the reservation room number to change in modify reservation;
16. Position – Used to specifify the position of desired area. Must be sent in decimal format. Better used when radius is passed too.
17. RadiusType – Used to search in a radius area. The point of calculus will be the Position passed on the Position object or the cityCode passed on addresstype.
18. Location – Used to search hotels by an IATA code
19. Award
    1. Rating – Used to search the minimum starts of desired hotels.
20. GetPricesPerGuest
    1. True or False indicating if the response should include the prices for adults and childs in the RoomRate Total TPA\_Extension fields. (default is False)
21. RatePlanCandidate – Can be sent up to 50 nodes.
    1. GroupCode – Used to unlock rates that are exclusive for a specific groupcode. Note that if multiple RatePlanCandidate is sent, only the first Group code will be valid.
    2. PromotionCode – Used to get a discount for rates allowed to have an discount for passed code. Note that if multiple RatePlanCandidate is sent, only the first Group code will be valid.
    3. RatePlanType – Used to get prices by rate type. Up to 50 RatePlanCandidate with different RatePlanType
22. ProfileInfo
    1. UniqueID
       1. ID – Id of a company, or Reservation number. Reservation number is mandatory when IsForModify = true;
       2. Type – Send Companhy for Omnibees CompnhayUID or Reservation to tell that unique id is a reservation number;
23. TPA\_Extensions
    1. AmountIncludingMarkup – When true indicate that response should include rates with amount including markup (**Commisionable rates**), false otherwise. Default is True
    2. AmountNotIncludingMarkup – When true indicate that response should include rates with amount not including markup (**Net Rates**), false otherwise. Default is true.
    3. AmountIsPackageRates – when true indicate that response should include rates that are package. Currently packages rates are in net model, false otherwise. Default is true
    4. MultimediaObjects – Used to ask for multiple images in response.

#### XML Requests

##### Search with paging

Fill page number and page size.

<OTA\_HotelAvailRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<BestOnly>true</BestOnly>

<EchoToken>eacffe1d-e505-4b5e-b7f5-04b965525427</EchoToken>

<HotelSearchCriteria>

<Criterion>

<RoomStayCandidatesType>

<RoomStayCandidates>

<RoomStayCandidate>

<GuestCountsType>

<GuestCounts>

<GuestCount>

<AgeQualifyCode>Adult</AgeQualifyCode>

<Count>1</Count>

</GuestCount>

</GuestCounts>

</GuestCountsType>

<Quantity>1</Quantity>

</RoomStayCandidate>

</RoomStayCandidates>

</RoomStayCandidatesType>

<StayDateRange>

<End>2014-04-02T00:00:00</End>

<Start>2014-03-27T00:00:00</Start>

</StayDateRange>

</Criterion>

</HotelSearchCriteria>

<MaxResponses>50</MaxResponses>

<PageNumber>1</PageNumber>

<PrimaryLangID>en</PrimaryLangID>

<Target>Test</Target>

<TimeStamp>2014-03-26T11:12:09.9023809+00:00</TimeStamp>

<Version>1</Version>

</OTA\_HotelAvailRQ>

##### Search without paging

Send page number and page size null. This request can be very heavy for communication bettew two systems and should be avoid. We allways recommend to use pagging.

<OTA\_HotelAvailRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<BestOnly>false</BestOnly>

<EchoToken>e4fdb263-d8d5-45fc-8352-e14eabae191a</EchoToken>

<HotelSearchCriteria>

<Criterion>

<RoomStayCandidatesType>

<RoomStayCandidates>

<RoomStayCandidate>

<GuestCountsType>

<GuestCounts>

<GuestCount>

<AgeQualifyCode>Adult</AgeQualifyCode>

<Count>1</Count>

</GuestCount>

</GuestCounts>

</GuestCountsType>

<Quantity>1</Quantity>

</RoomStayCandidate>

</RoomStayCandidates>

</RoomStayCandidatesType>

<StayDateRange>

<End>2014-04-02T00:00:00</End>

<Start>2014-03-27T00:00:00</Start>

</StayDateRange>

</Criterion>

</HotelSearchCriteria>

<MaxResponses xsi:nil="true" />

<PageNumber xsi:nil="true" />

<PrimaryLangID>en</PrimaryLangID>

<Target>Test</Target>

<TimeStamp>2014-03-26T11:15:23.8454004+00:00</TimeStamp>

<Version>1</Version>

</OTA\_HotelAvailRQ>

##### Search by using Only best available rates for each hotel

Send property BestOnly = true

<OTA\_HotelAvailRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<BestOnly>true</BestOnly>

<EchoToken>0684642f-aaeb-4aad-ae59-9e80c4242639</EchoToken>

<HotelSearchCriteria>

<Criterion>

<RoomStayCandidatesType>

<RoomStayCandidates>

<RoomStayCandidate>

<GuestCountsType>

<GuestCounts>

<GuestCount>

<Age xsi:nil="true" />

<AgeQualifyCode>Adult</AgeQualifyCode>

<Count>1</Count>

</GuestCount>

</GuestCounts>

</GuestCountsType>

<Quantity>1</Quantity>

<RoomID xsi:nil="true" />

</RoomStayCandidate>

</RoomStayCandidates>

</RoomStayCandidatesType>

<StayDateRange>

<End>2014-04-02T00:00:00</End>

<Start>2014-03-27T00:00:00</Start>

</StayDateRange>

</Criterion>

</HotelSearchCriteria>

<MaxResponses>50</MaxResponses>

<PageNumber>1</PageNumber>

<PrimaryLangID>en</PrimaryLangID>

<Target>Test</Target>

<TimeStamp>2014-03-26T11:18:24.1082087+00:00</TimeStamp>

<Version>1</Version>

</OTA\_HotelAvailRQ>

##### Search for rooms with differ occupation

<OTA\_HotelAvailRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<BestOnly>true</BestOnly>

<EchoToken>2cb20b01-c69e-4f0d-8c66-d5d8cee173cc</EchoToken>

<HotelSearchCriteria>

<Criterion>

<RoomStayCandidatesType>

<RoomStayCandidates>

<RoomStayCandidate>

<GuestCountsType>

<GuestCounts>

<GuestCount>

<Age xsi:nil="true" />

<AgeQualifyCode>Adult</AgeQualifyCode>

<Count>1</Count>

</GuestCount>

<GuestCount>

<Age>4</Age>

<AgeQualifyCode>Child</AgeQualifyCode>

<Count>1</Count>

</GuestCount>

</GuestCounts>

</GuestCountsType>

The RPH Must be unique by each room. Thid id can be used at response to identify what price is for what room.

<Quantity>1</Quantity>

<RPH>0</RPH>

<RoomID xsi:nil="true" />

</RoomStayCandidate>

<RoomStayCandidate>

<GuestCountsType>

<GuestCounts>

<GuestCount>

<Age xsi:nil="true" />

<AgeQualifyCode>Adult</AgeQualifyCode>

<Count>3</Count>

</GuestCount>

</GuestCounts>

</GuestCountsType>

<Quantity>1</Quantity>

<RPH>1</RPH>

<RoomID xsi:nil="true" />

</RoomStayCandidate>

</RoomStayCandidates>

</RoomStayCandidatesType>

<StayDateRange>

<End>2014-05-27T00:00:00</End>

<Start>2014-05-21T00:00:00</Start>

</StayDateRange>

</Criterion>

</HotelSearchCriteria>

<MaxResponses>50</MaxResponses>

<PageNumber>1</PageNumber>

<PrimaryLangID>en</PrimaryLangID>

<Target>Test</Target>

<TimeStamp>2014-05-20T15:41:35.6735222+01:00</TimeStamp>

<Version>1</Version>

</OTA\_HotelAvailRQ>

PLEASE NOTE:

RPH will be used to make a relation between request and response.

For example on response RoomStayCandidateRPH indicates the rph used for search one specific occupation.

<RoomRates>

<RoomRate>

<EffectiveDate>2014-05-21T00:00:00</EffectiveDate>

<ExpireDate>2014-05-27T00:00:00</ExpireDate>

<RoomStayCandidateRPH>0</RoomStayCandidateRPH>

**For Booking proposes the partner need to split room rates object and send as room stays**.

##### Search by IATA With 2 Adults and no child’s

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

<OTA\_HotelAvailRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<EchoToken>a34bdc80-ce89-4083-bf19-8c4f8cf87302</EchoToken>

<HotelSearchCriteria>

<Criterion>

<Location>

<CodeContext>IATA</CodeContext>

<LocationCode>FAO</LocationCode>

</Location>

<RoomStayCandidatesType>

<RoomStayCandidates>

<RoomStayCandidate>

<GuestCountsType>

<GuestCounts>

<GuestCount>

<Age xsi:nil="true" />

<AgeQualifyCode>Adult</AgeQualifyCode>

<Count>2</Count>

</GuestCount>

</GuestCounts>

</GuestCountsType>

<RoomID xsi:nil="true" />

</RoomStayCandidate>

</RoomStayCandidates>

</RoomStayCandidatesType>

<StayDateRange>

<End>2013-08-03T00:00:00</End>

<Start>2013-08-01T00:00:00</Start>

</StayDateRange>

</Criterion>

</HotelSearchCriteria>

<MaxResponses>50</MaxResponses>

<PageNumber>1</PageNumber>

<PrimaryLangID>en</PrimaryLangID>

<Target>Test</Target>

<TimeStamp>2013-07-10T14:58:27.5930934+01:00</TimeStamp>

<Version>1</Version>

</OTA\_HotelAvailRQ>

##### Search by IATA with 2 Adults and 2 Childs

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

<OTA\_HotelAvailRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<EchoToken>3bdc9519-f4e2-4554-bf1b-7f81af1a3f07</EchoToken>

<HotelSearchCriteria>

<Criterion>

<Location>

<CodeContext>IATA</CodeContext>

<LocationCode>FAO</LocationCode>

</Location>

<RoomStayCandidatesType>

<RoomStayCandidates>

<RoomStayCandidate>

<GuestCountsType>

<GuestCounts>

<GuestCount>

<Age xsi:nil="true" />

<AgeQualifyCode>Adult</AgeQualifyCode>

<Count>2</Count>

</GuestCount>

<GuestCount>

<Age>1</Age>

<AgeQualifyCode>Child</AgeQualifyCode>

<Count>1</Count>

</GuestCount>

<GuestCount>

<Age>5</Age>

<AgeQualifyCode>Child</AgeQualifyCode>

<Count>1</Count>

</GuestCount>

</GuestCounts>

</GuestCountsType>

<RoomID xsi:nil="true" />

</RoomStayCandidate>

</RoomStayCandidates>

</RoomStayCandidatesType>

<StayDateRange>

<End>2013-08-18T00:00:00</End>

<Start>2013-08-11T00:00:00</Start>

</StayDateRange>

</Criterion>

</HotelSearchCriteria>

<MaxResponses>50</MaxResponses>

<PageNumber>1</PageNumber>

<PrimaryLangID>en</PrimaryLangID>

<Target>Test</Target>

<TimeStamp>2013-07-10T15:12:07.905835+01:00</TimeStamp>

<Version>1</Version>

</OTA\_HotelAvailRQ>

##### Search by country/state/city

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

<OTA\_HotelAvailRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<EchoToken>cc119c6a-9c28-4298-96cf-9b7c7017804d</EchoToken>

<HotelSearchCriteria>

<Criterion>

<Address>

<CityCode>3304186</CityCode>

<CountryCode>157</CountryCode>

<StateProvCode>7466773</StateProvCode>

</Address>

<RoomStayCandidatesType>

<RoomStayCandidates>

<RoomStayCandidate>

<GuestCountsType>

<GuestCounts>

<GuestCount>

<Age xsi:nil="true" />

<AgeQualifyCode>Adult</AgeQualifyCode>

<Count>2</Count>

</GuestCount>

</GuestCounts>

</GuestCountsType>

<RoomID xsi:nil="true" />

</RoomStayCandidate>

</RoomStayCandidates>

</RoomStayCandidatesType>

<StayDateRange>

<End>2013-08-18T00:00:00</End>

<Start>2013-08-11T00:00:00</Start>

</StayDateRange>

</Criterion>

</HotelSearchCriteria>

<MaxResponses>50</MaxResponses>

<PageNumber>1</PageNumber>

<PrimaryLangID>en</PrimaryLangID>

<Target>Test</Target>

<TimeStamp>2013-07-10T15:13:34.5974162+01:00</TimeStamp>

<Version>1</Version>

</OTA\_HotelAvailRQ>

##### Search by Agency PCC

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

<OTA\_HotelAvailRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<EchoToken>82de21b4-9f43-46e8-b370-810301b3758c</EchoToken>

<HotelSearchCriteria>

<Criterion>

<Location>

<CodeContext>IATA</CodeContext>

<LocationCode>FAO</LocationCode>

</Location>

<RoomStayCandidatesType>

<RoomStayCandidates>

<RoomStayCandidate>

<GuestCountsType>

<GuestCounts>

<GuestCount>

<Age xsi:nil="true" />

<AgeQualifyCode>Adult</AgeQualifyCode>

<Count>2</Count>

</GuestCount>

</GuestCounts>

</GuestCountsType>

<RoomID xsi:nil="true" />

</RoomStayCandidate>

</RoomStayCandidates>

</RoomStayCandidatesType>

<StayDateRange>

<End>2013-08-03T00:00:00</End>

<Start>2013-08-01T00:00:00</Start>

</StayDateRange>

</Criterion>

</HotelSearchCriteria>

<MaxResponses>50</MaxResponses>

<POS>

<Sources>

<Source>

<PseudoCityCode>2015</PseudoCityCode>

</Source>

</Sources>

</POS>

<PageNumber>1</PageNumber>

<PrimaryLangID>en</PrimaryLangID>

<Target>Test</Target>

<TimeStamp>2013-07-10T15:45:12.5652731+01:00</TimeStamp>

<Version>1</Version>

</OTA\_HotelAvailRQ>

##### Search by Corporate code

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

<OTA\_HotelAvailRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<EchoToken>ba63fa7b-476e-4ee4-a4a1-c1e65e2a4f1a</EchoToken>

<HotelSearchCriteria>

<Criterion>

<Location>

<CodeContext>IATA</CodeContext>

<LocationCode>FAO</LocationCode>

</Location>

<ProfilesType>

<ProfileInfos>

<ProfileInfo>

<UniqueID>

<ID>2151456</ID>

<Type>Company</Type>

</UniqueID>

</ProfileInfo>

</ProfileInfos>

</ProfilesType>

<RoomStayCandidatesType>

<RoomStayCandidates>

<RoomStayCandidate>

<GuestCountsType>

<GuestCounts>

<GuestCount>

<Age xsi:nil="true" />

<AgeQualifyCode>Adult</AgeQualifyCode>

<Count>2</Count>

</GuestCount>

</GuestCounts>

</GuestCountsType>

<RoomID xsi:nil="true" />

</RoomStayCandidate>

</RoomStayCandidates>

</RoomStayCandidatesType>

<StayDateRange>

<End>2013-08-03T00:00:00</End>

<Start>2013-08-01T00:00:00</Start>

</StayDateRange>

</Criterion>

</HotelSearchCriteria>

<MaxResponses>50</MaxResponses>

<PageNumber>1</PageNumber>

<PrimaryLangID>en</PrimaryLangID>

<Target>Test</Target>

<TimeStamp>2013-07-10T15:47:10.3071276+01:00</TimeStamp>

<Version>1</Version>

</OTA\_HotelAvailRQ>

##### Search by agency PCC and Corporate code

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

<OTA\_HotelAvailRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<EchoToken>4d0c94a2-e2b6-4f5d-bfdd-c5c12b7261c1</EchoToken>

<HotelSearchCriteria>

<Criterion>

<ProfilesType>

<ProfileInfos>

<ProfileInfo>

<UniqueID>

<ID>2151456</ID>

<Type>Company</Type>

</UniqueID>

</ProfileInfo>

</ProfileInfos>

</ProfilesType>

<RoomStayCandidatesType>

<RoomStayCandidates>

<RoomStayCandidate>

<GuestCountsType>

<GuestCounts>

<GuestCount>

<Age xsi:nil="true" />

<AgeQualifyCode>Adult</AgeQualifyCode>

<Count>1</Count>

</GuestCount>

</GuestCounts>

</GuestCountsType>

<RoomID xsi:nil="true" />

</RoomStayCandidate>

</RoomStayCandidates>

</RoomStayCandidatesType>

<StayDateRange>

<End>2013-08-18T00:00:00</End>

<Start>2013-08-11T00:00:00</Start>

</StayDateRange>

</Criterion>

</HotelSearchCriteria>

<MaxResponses>50</MaxResponses>

<POS>

<Sources>

<Source>

<PseudoCityCode>2015</PseudoCityCode>

</Source>

</Sources>

</POS>

<PageNumber>1</PageNumber>

<PrimaryLangID>en</PrimaryLangID>

<Target>Test</Target>

<TimeStamp>2013-07-10T15:47:54.5316173+01:00</TimeStamp>

<Version>1</Version>

</OTA\_HotelAvailRQ>

##### Search by a Agency Post. An Agency in a company

<OTA\_HotelAvailRQ xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<BestOnly>true</BestOnly>

<EchoToken>5caa3514-fd68-4874-a5b8-a907efa99284</EchoToken>

<HotelSearchCriteria>

<Criterion>

<RoomStayCandidatesType>

<RoomStayCandidates>

<RoomStayCandidate>

<GuestCountsType>

<GuestCounts>

<GuestCount>

<Age xsi:nil="true" />

<AgeQualifyCode>Adult</AgeQualifyCode>

<Count>1</Count>

</GuestCount>

</GuestCounts>

</GuestCountsType>

<Quantity>1</Quantity>

<RPH>0</RPH>

<RoomID xsi:nil="true" />

</RoomStayCandidate>

</RoomStayCandidates>

</RoomStayCandidatesType>

<StayDateRange>

<End>2014-07-01T00:00:00</End>

<Start>2014-06-25T00:00:00</Start>

</StayDateRange>

</Criterion>

</HotelSearchCriteria>

<MaxResponses>50</MaxResponses>

<POS>

<Sources>

<Source>

<RequestorID>

<ID>TESTP|ABB</ID>

<Type>CorporationRepresentative</Type>

</RequestorID>

</Source>

</Sources>

</POS>

<PageNumber>1</PageNumber>

<PrimaryLangID>en</PrimaryLangID>

<Target>Test</Target>

<TimeStamp>2014-06-24T12:54:42.9948777+01:00</TimeStamp>

<Version>1</Version>

</OTA\_HotelAvailRQ>

##### Search with all filters

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

<OTA\_HotelAvailRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<EchoToken>2350ffd2-4c3f-45c7-a49f-5e1be671568f</EchoToken>

<HotelSearchCriteria>

<Criterion>

<Address>

<CityCode>3304186</CityCode>

<CountryCode>157</CountryCode>

<StateProvCode>7466773</StateProvCode>

<ZoneCode>258622544</ZonCode>

</Address>

<Award>

<Rating>4</Rating>

</Award>

<HotelRef>

<HotelRef>

<ChainCode>1232</ChainCode>

<HotelCode>1814</HotelCode>

</HotelRef>

</HotelRef>

<Location>

<CodeContext>IATA</CodeContext>

<LocationCode>FAO</LocationCode>

</Location>

<ProfilesType>

<ProfileInfos>

<ProfileInfo>

<UniqueID>

<ID>12315</ID>

<Type>Company</Type>

</UniqueID>

</ProfileInfo>

</ProfileInfos>

</ProfilesType>

<RateRanges>

<RateRange>

<MaxRate>250</MaxRate>

<MinRate>10</MinRate>

</RateRange>

</RateRanges>

<RoomStayCandidatesType>

<RoomStayCandidates>

<RoomStayCandidate>

<GuestCountsType>

<GuestCounts>

<GuestCount>

<Age xsi:nil="true" />

<AgeQualifyCode>Adult</AgeQualifyCode>

<Count>2</Count>

</GuestCount>

<GuestCount>

<Age>0</Age>

<AgeQualifyCode>Child</AgeQualifyCode>

<Count>1</Count>

</GuestCount>

<GuestCount>

<Age>5</Age>

<AgeQualifyCode>Child</AgeQualifyCode>

<Count>1</Count>

</GuestCount>

</GuestCounts>

</GuestCountsType>

<RoomID>5820</RoomID>

</RoomStayCandidate>

</RoomStayCandidates>

</RoomStayCandidatesType>

<StayDateRange>

<End>2013-08-03T00:00:00</End>

<Start>2013-08-01T00:00:00</Start>

</StayDateRange>

</Criterion>

</HotelSearchCriteria>

<MaxResponses>50</MaxResponses>

<POS>

<Sources>

<Source>

<PseudoCityCode>2015</PseudoCityCode>

</Source>

</Sources>

</POS>

<PageNumber>1</PageNumber>

<PrimaryLangID>en</PrimaryLangID>

<Target>Test</Target>

<TimeStamp>2013-07-10T15:56:52.8265903+01:00</TimeStamp>

<Version>1</Version>

</OTA\_HotelAvailRQ>

### OTA\_HotelAvailRS

#### Schema

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Level | Occurrence | Element | Property | | Mandatory (M)/ Optional (O) | Sample Usage |
| 0 | 1 | OTA\_HotelAvailRS |  | |  |  |
|  |  |  | Echotoken | | M | Same as request |
|  |  |  | MaxResponses | | O | Same as request |
|  |  |  | PageNumber | | O | Same as request |
|  |  |  | PrimaryLanID | | O | Same as request |
|  |  |  | TimeStamp | | M | 2014-06-23T15:04:16.0294414Z  Responser time in utc |
|  |  |  | Target | | M | Same as request |
|  |  |  | Version | | M | Same as request |
|  |  |  |  | |  |  |
| 1 | 0..1 | POSType |  | | O | Same as request |
|  |  |  |  |  | |  |
| 1 | 1 | HotelSearchCriteria |  | | M | Same as request |
|  |  |  |  | |  |  |
| 1 | 0..1 | HotelStaysType |  | | M | When availability found |
| 2 | 1..n | HotelStay |  | | M |  |
|  |  |  | RoomStayRPH | | M | 1 - Reference to Roomstay |
| 3 | 0..1 | Availability |  | | O |  |
|  |  |  | Start | | M | Allways CheckInDate |
|  |  |  | End | | M | Allways CheckOutDate |
| 4 | 0..3 | Restriction |  | | M |  |
|  |  |  | Time | | M | 2 |
|  |  |  | TimeUnit | | M | Allways Day (TU Codes) |
|  |  |  | RestrictionType | | M | MinLOS (RT Codes) |
| 3 | 1 | Price |  | | M | Allways the best price found |
|  |  |  | AmountAfterTax | | M | 50.21 |
|  |  |  | AmountBeforeTax | | M | 35.00 |
|  |  |  | CurrencyCode | | M | EUR (Currencies Codes) |
|  |  |  | End | | M | Allways CheckOut |
|  |  |  | Start | | M | Allways CheckIN |
| 3 | 1 | BasicPropertyInfoType |  | | M |  |
|  |  |  | CurrencyCode | | M | Default Currency for hotel, can be different than rates price |
|  |  |  | ImageURL | | M | Image of hotel |
| 4 | 0..2 | Award |  | | O |  |
|  |  |  | Rating | | M | 4 |
|  |  |  | Name | | M | Stars |
| 4 | 0..1 | AddressType |  | | O |  |
|  |  |  | CountryCode | | O | 40 |
|  |  |  | StateProvCode | | O | 6362053 |
|  |  |  | CityCode | | O | 2851556 |
|  |  |  | ZoneCode | | O | 56531 |
|  |  |  | AddressLine | | O | Rua 25 de Abril Faro |
| 4 | 1 | HotelRef |  | | M |  |
|  |  |  | ChainCode | | O | 1452 |
|  |  |  | ChainName | | O | Omnibees |
|  |  |  | HotelCode | | M | 1635 |
|  |  |  | HotelName | | O | Hotel Demo Omnibees |
| 4 | 1 | PositionType |  | |  |  |
|  |  |  | Latitude | | O | -11,18639 |
|  |  |  | Longitude | | O | -22,37278 |
|  |  |  |  | |  |  |
| 1 | 0..1 | RoomStaysType |  | | M | When availability found |
|  |  |  | MoreIndicator | | M | True |
| 2 | 1..n | RoomStay |  | | M |  |
|  |  |  | RoomStayLanguage | | M | En |
|  |  |  | RPH | | M | 1 |
|  |  |  | WarningRPH | | O | 1 |
|  |  |  | AvailabilityStatus | | O | AvailableForSale |
| 3 | 1 | **BasicPropertyInfoType** |  | | M | Same as in hotelstay object |
|  |  |  |  | |  |  |
| 3 | 1..n | **RoomType** |  | | M |  |
|  |  |  | MaxOccupancy | | M | 4 |
|  |  |  | NumberOfUnits | | M | 10 |
|  |  |  | RoomID | | M | 2852 |
|  |  |  | RoomName | | M | Double Room |
| 4 | 0..1 | RoomDescrition |  | | O |  |
|  |  |  | Description | | O | Description of room |
|  |  |  | Language | | M | En |
| 4 | 0..2 | Occupancy |  | | O |  |
|  |  |  | AgeQualifyCode | | M | Adult |
|  |  |  | MaxOccupancy | | O | 4 |
|  |  |  | MinOccupancy | | O | 1 |
|  |  |  |  | |  |  |
| 3 | 1..n | **RatePlanType** |  | |  |  |
|  |  |  | EffectiveDate | | O | 2014-08-01 |
|  |  |  | ExpireDate | | O | 2014-10-30 |
|  |  |  | RatePlanID | | M | 23454 |
|  |  |  | RatePlanName | | M | BAR |
|  |  |  | RatePlanTypeCode | | O | Promotional |
|  |  | RatePlanDescription |  | | O |  |
|  |  |  | Description | | O | Rate bar description |
|  |  |  | Language | | M | en (Languages codes) |
| 4 | 0..1 | MealsIncluded |  | | O |  |
|  |  |  | Breakfast | | M | True |
|  |  |  | Dinner | | M | False |
|  |  |  | Lunch | | M | False |
|  |  |  | MealPlanIndicator | | M | Allways True |
|  |  |  | MealPlanCode | | M | BeadBreakfast (MPT codes) |
|  |  |  | ID | | M | 25468 |
|  |  |  | Name | | O | Meal plan name |
|  |  |  | Description | | O | Meal plan Description |
| 4 | 0..n | RatePlanInclusion |  | | O | Included Extras |
|  |  |  | ID | | M | 5874 |
|  |  |  | Start | | O | 2014-05-10 |
|  |  |  | End | | O | 2014-08-12 |
| 5 |  | RatePlanInclusionDescription |  | | M |  |
|  |  |  | Name | | M | Massage |
|  |  |  | Description | | O | Massage description |
|  |  |  | Language | | M | En (Languages codes) |
| 4 | 0..1 | Commission |  | | O | Only apply to total with AmountIncludingMarkup = true |
|  |  |  | Percent | | M | 20 |
| 4 | 0..n | Guarantee |  | |  |  |
|  |  |  | Start | | O | 2014-05-02 |
|  |  |  | End | | O | 2014-12-01 |
|  |  |  | GuaranteeCode | | M | 2548 |
| 5 | 0..1 | AmountPercent |  | | O | Indicates the values required for deposit |
|  |  |  | CurrencyCode | | M | EUR |
|  |  |  | Amount | | O | 0 |
|  |  |  | NmbrOfNights | | O | 2 |
|  |  |  | Percent | | O | 0 |
| 5 | 0..1 | Deadline |  | | O | Used to inform when policy will be applied. When null policy apply always. |
|  |  |  | OffsetDropTime | |  | Always BeforeArrival |
|  |  |  | OffsetUnitMultiplier | |  | 2 |
|  |  |  | TimeUnitType | |  | Always Day |
| 5 | 0..1 | GuaranteeDescription |  | | O | Used For Deposit policies |
|  |  |  | Name | | M | 25 % Advance |
|  |  |  | Description | | O | Policy description |
|  |  |  | Language | | M | En (Languages codes) |
| 5 | 0..1 | GuaranteesAcceptedType |  | | O | Used to send Guarantees Allowed for deposit |
| 6 | 1..n | GuaranteeAccepted |  | | M |  |
|  |  |  | GuaranteeTypeCode | | M | Credit Card (PMT Codes) |
| 4 | 1 | CancelPenalty |  | | M |  |
|  |  |  | Start | | O | Null |
|  |  |  | End | | O | Null |
|  |  |  | NonRefundable | | M | False |
| 5 | 1 | PenaltyDescription |  | | M |  |
|  |  |  | Name | | M | 2 Days |
|  |  |  | Description | | O | Cancelation policy |
|  |  |  | Language | | M | En |
| 5 | 0..1 | DeadLine |  | | O |  |
|  |  |  | OffsetUnitMultiplier | | M | 2 |
|  |  |  | TimeUnitType | | M | Allways Day |
|  |  |  | OffsetDropTime | | M | Allways BeforeArrival |
| 5 | 0..1 | AmountPercent |  | | O |  |
|  |  |  | Amount | | O | 0 |
|  |  |  | NmbrOfNights | | O | 0 |
|  |  |  | Percent | | O | 20 |
|  |  |  | CurrencyCode | | M | EUR (Currencies codes) |
| 4 | 0..N | TaxPolicy |  | | O |  |
|  |  |  | CurrencyCode | | M | EUR (Currencies codes) |
|  |  |  | Description | | O | IVA |
|  |  |  | ID | | M | 14525 |
|  |  |  | IsPerNight | | M | True |
|  |  |  | IsPerPerson | | M | False |
|  |  |  | IsPerRoom | | M | True |
|  |  |  | IsPerStay | | M | False |
|  |  |  | IsValuePercentage | | M | False |
|  |  |  | Name | | M | Tax Iva |
|  |  |  | Value | | M | 75.00 |
| 4 | 0..1 | AdditionalDetailsType |  | | O | Used as Other Policies (just information) |
| 5 | 0..N | AdditionalDetail |  | | O |  |
| 6 | 1 | AdditionalDetailDescription |  | | M |  |
|  |  |  | Description | | O | Dogs Policy description |
|  |  |  | Name | | M | Dogs Policy |
|  |  |  | Language | | M | En (Languages codes) |
| 4 | 0..N | Offer |  | | O |  |
|  |  |  | OfferCode | | M | 145#254 |
|  |  |  | RPH | | M | 145#254 |
| 5 | 0..1 | OfferRule |  | | O |  |
|  |  |  | MaxAdvancedBookingOffset | | O | 5 |
|  |  |  | MinAdvancedBookingOffset | | O | Null |
| 6 | 0..1 | DateRestriction |  | | O |  |
|  |  |  | Start | | M | 2014-05-12 |
|  |  |  | End | | M | 2014-08-12 |
| 5 | 0..1 | Discount |  | | M |  |
|  |  |  | ChargeUnitCode | | M | Allways PerRoomPerStay |
|  |  |  | DiscountPattern | | O | NULL |
|  |  |  | NightsDiscounted | | O | NULL |
|  |  |  | NightsRequired | | O | NULL |
|  |  |  | Percent | | O | 10 |
| 5 | 0..N | CompatibleOffer |  | | O | Used to indicate cumulative offers |
|  |  |  | IncompatibleOfferInidcator | | M | Allways false |
|  |  |  | OfferRPH | | M | 1458#566 |
| 5 | 0..1 | OfferDescription |  | | O |  |
|  |  |  | Description | | M | Special Discount |
|  |  |  | Language | | M | en ( Languages codes) |
| 4 | 0..1 | PaymentPolicies |  | |  |  |
| 5 | 0..1 | AcceptedPayments |  | |  |  |
| 6 | 1..N | AcceptedPayment |  | |  |  |
|  |  |  | GuaranteeTypeCode | | M | Credit Card (PMT Types) |
|  |  |  |  | |  |  |
| 3 | 1..n | **RoomRate** |  | | M |  |
|  |  |  | RoomStayCandidateRPH | | M | Allways matchinf rph sended in request |
|  |  |  | EffectiveDate | | M | Allways Checkin date |
|  |  |  | ExpireDate | | M | Allways Checkout date |
|  |  |  | RatePlanID | | M | 23454 |
|  |  |  | RoomID | | M | 2852 |
|  |  |  | PromotionCode | | O | 12CC |
|  |  |  | GroupCode | | O | 76542 |
|  |  |  |  | |  |  |
| 4 | 1 | RoomRateAvailability |  | |  |  |
|  |  |  | AvailabilityStatus | | M | AvailableForSale |
|  |  |  | WarningRPH | | M | 562 |
| 4 | 1 | RatesType |  | | M | Used for day prices without tax |
| 5 | 1..N | Rate |  | | M |  |
|  |  |  | EfectiveDate | | M | 2014-05-12 |
|  |  |  | MaxLOS | | O | 30 |
|  |  |  | MinLOS | | O | 1 |
|  |  |  | NumberOfUnits | | O | 5 |
|  |  |  | StayThrough | | O | 2 |
|  |  |  | MinAdvancedBookingOffset | | O | 5 |
|  |  |  | Status | | O | Close |
|  |  |  |  | |  |  |
| 6 | 1 | Total |  | | M | **Rate Level** |
|  |  |  | AmountBeforeTax | | M | 25.00 |
|  |  |  | AmountAfterTax | |  | Allways 0. Tax are not calculated at day level |
|  |  |  | AmountIncludingMarkup | | M | True |
|  |  |  | AmountIsPackage | | O | false |
|  |  |  | ChargeType | | M | Allways PerRoomPerNight at day price Level |
|  |  |  | CurrencyCode | | M | EUR (Currencies codes) |
| 5 | 0..1 | TPA\_Extensions |  | | M | Mandatory in version 1.8 |
| 6 | 0..1 | MultimediaDescriptionsType |  | |  |  |
| 7 | 1..N | MultimediaDescription |  | |  |  |
|  |  |  | AdditionalDetailCode | | M | PropertyDescription |
|  |  |  | ID | | M | 1814 |
|  |  |  | InfoCode | | M | Marketing |
| 8 | 0..1 | ImageItemsType |  | |  |  |
| 9 | 1..N | ImageItem |  | | M |  |
|  |  |  | Name | | M | Image |
| 10 | 1 | URL |  | | M |  |
|  |  |  | Address | | M | Media.omnibees.com |
|  |  |  | URLType | | M | Images |
| 10 | 1 | ImageFormat |  | | M |  |
| 11 | 1 | ImageItemFormat |  | | M |  |
|  |  |  | Sort | | M | 0 |
|  |  |  |  | |  |  |
|  |  |  |  | |  |  |
|  |  |  |  | |  |  |
|  |  |  |  | |  |  |
|  |  |  |  | |  |  |
| 6 | 1 | Rates |  | | M | Array with Rate type |
| 7 | 1..N | Rate |  | | M | Used to specify the per guest group by age |
|  |  |  | AgeQualifyingCode | | M | Adult |
|  |  |  | EffectiveDate | | M | 2015-06-19 |
|  |  |  | MaxGuestApplicable | | M | 2 |
|  |  |  | MinGuestApplicable | | M | 2 |
| 8 | 1 | Total |  | |  |  |
|  |  |  | AmountAfterTax | | - | Taxe values are not applied in prices by day |
|  |  |  | AmountBeforeTax | | M | 45.00 |
|  |  |  | AmountIncludingMarkup | | M | True |
|  |  |  | AmountIsPackage | | O | false |
|  |  |  | ChargeType | | M | Allways PerPerson at rate Level |
| 4 | 0..N | Discount |  | | O | This Discount is included in price |
|  |  |  | ChargeUnitCode | | M | PerRoomPerStay (CHG Codes) |
|  |  |  | DiscountCode | | M | 345125 |
|  |  |  | Percent | | M | 20 |
| 5 | 0..1 | DiscountReason |  | | O |  |
|  |  |  | Description | | M | Promotional Rate offer |
|  |  |  |  | |  |  |
| 4 | 0..N | ServiceRPH |  | | O | Used to identify allowed Optional extras for this roomrate |
|  |  | RPH |  | | M | 14552 |
| 4 | 1 | Total |  | |  | **RoomRate Level,** Allways the sum of rate plus taxes |
|  |  |  | AmountBeforeTax | | M | 20.00 |
|  |  |  | AmountAfterTax | |  | 95.00 |
|  |  |  | AmountIncludingMarkup | | M | True |
|  |  |  | AmountIsPackage | | O | false |
|  |  |  | ChargeType | | M | Allways PerRoomPerStay at roomrate price Level |
|  |  |  | CurrencyCode | | M | EUR (Currencies codes) |
| 5 | 0..1 | TPA\_Extensions |  | | M | Mandatory in version 1.8 |
| 6 | 1 | GuestsTotalRate |  | | M | Array with GuestTotalRateType |
| 7 | 1..N | GuestTotalRate |  | | M | Used to specify the total per guest age qualify code |
|  |  |  | AgeQualifyingCode | | M | Adult |
| 8 | 1 | Total |  | |  |  |
|  |  |  | AmountBeforeTax | | M | 20.00 |
|  |  |  | AmountAfterTax | | M | 95.00 |
|  |  |  | AmountIncludingMarkup | | M | True |
|  |  |  | AmountIsPackage | | O | false |
|  |  |  | ChargeType | | M | Allways PerRoomPerStay at roomrate price Level |
|  |  |  |  | |  |  |
| 3 | 1..n | **Total** |  | | M | **RoomStay Level**, Allways the best roomrate found |
|  |  |  | AmountBeforeTax | | M | 20.00 |
|  |  |  | AmountAfterTax | |  | 95.00 |
|  |  |  | AmountIncludingMarkup | | M | True |
|  |  |  | AmountIsPackage | | O | false |
|  |  |  | ChargeType | | M | Allways PerStay at roomStay price Level |
|  |  |  |  | |  |  |
| 3 | 1 | **TPA\_Extensions** |  | |  |  |
|  |  |  | ApprovalInvoiced | | O | 2 |
|  |  |  |  | |  |  |
| 1 | 0..N | Service |  | | O |  |
|  |  |  | ID | | M | 14552 |
|  |  |  | ServiceCategoryCode | | O | Null |
|  |  |  | ServicePricingType | | M | PerRoomPerNight (CHG Codes) |
|  |  |  | ServiceRPH | | M | 14552 |
| 2 | 1 | ServiceDescription |  | | M |  |
|  |  |  | Name | | M | Cleaning Service |
|  |  |  | Description | | O | Service Description |
|  |  |  | Language | | M | en (Languages Codes) |
|  |  |  | ImageURL | | O | Url of image |
|  |  |  |  | |  |  |
| 1 | 0..1 | Sucess |  | | O |  |
|  |  |  |  | |  |  |
| 1 | 0..1 | WarningsType |  | | O |  |
| 2 | 1..N | Warning |  | | M |  |
|  |  |  | Code | | M | 450 |
|  |  |  | Language | | M | En |
|  |  |  | ShortText | | M | Warning Text |
|  |  |  |  | |  |  |
| 1 | 0..1 | ErrorsType |  | | O |  |
| 2 | 1..N | Error |  | | M |  |
|  |  |  | Code | | M | 448 |
|  |  |  | Language | | M | en |
|  |  |  | ShortText | | M | System error: Internal system error. Please contact techincal support at integration.team@omnibees.com { echoToken = 57fde0db-32ed-40c5-bf88-e19a7601472a } |
|  |  |  |  | |  |  |

Table 8 - OTA\_HotelAvailRS schema

#### General Rules

1. TimeStamp – used to specific date at Omnibees server;
2. Echotoken – Same as request, to check if response match with request and to help on futher support;
3. Success- used to inform that communication with Omnibees server was ok. When this tag is null the service will fill ErrorsType;
4. Warning – Used to specify a warning related to business rules like no availability. This tag will be sent with success tag;
5. Error – Used to inform that an error occur at Omnibees server. The partner should contact Omnibees support with the echotoken passed.
6. HotelStay – this object will have a summary of the best price for each hotel found. Just one hotelstay per hotel will be found.
   1. RoomStayRPH – Used to make a relation with RoomStayObject. This value is the value found at RPH on roomstay object;
7. RoomStay- Used to specifify prices as room and rates info. Can be applied diferent concepts based on request. Just one roomstay per hotel will be found.
   1. When request is made by using **Quantiy > 1** on roomstaycandidate, the response roomstay has:
      1. The total price is the total price for quatity of rooms
      2. Further in reservation this object will be cloned and the prices should be divided by a number of rooms.
      3. On reservation Omibees has to receive one roomstay per room quantity made on request.
      4. For example:
         1. a request for quatity = 3 - the response roomstay has Priceaftertax=330 and the pricebeforetax =300
         2. The booking must be made with 3 Roomstays each with pricebeforetax=100 and priceaftertax=110
      5. The rule will be applied for every roomratefound.
   2. When request is made with more than on RoomstayCandidate with different RPH, the response roomstay has:
      1. The response will have has roomrate with different RoomStayCandidateRPH matching the rph passed on request;
      2. Each roomrate is just for one rph;
      3. On reservation desired roomrates must be splited into roomstays
      4. Each roomstays on reservation must have the same hotelID
   3. RoomstayLanguage – Each object on roomstays will have the request language. If hotel do not have configured the request language the service will respond this tag the same of request, but the description can be on hotel default language.
   4. RPH – Unique id for roomstay
   5. Total – **ROOM STAY LEVEL TOTAL**: Used to specify the total for the best RoomRate found.
      1. AmountAfterTax - Use to specify the amount after tax, the tax info can be found in RatePlan object;
      2. AmountBeforeTax – Use to specify the amount before tax for this day;
      3. AmountIncludingMarkup – When true means that price is a pvp price. When false means the channel must apply the markup before sell.
      4. AmountIsPackage – Indicates that price is a package price
   6. TPA\_Extensions – this object is used to send objects in response that are specific to Omnibees.
      1. ApprovalInvoiced – Informs the type of approval that partner can use to get the relation with payment type. Allowed values are:

1 – Approved only daily;

2 – Approved daily and extras;

3 – Unapproved;

When Unapproved is return, Voucher cannot be used for payment type. Voucher can be returned in allowed guarantees for payment, because it’s configured for rate, but if Unapproved is return for Approval Invoiced the Voucher cannot be used due a restriction configured by hotel in credit module.

1. BasicPropertyInfo – Used to specify basic propertyinfo, like City code, country code, state code, Number of stars, hotel base currency code, image of hotel, chaincode, chain name, hotel code, hotel name, latitude and longitude and address.
2. Roomtype – Used to specify basic information about the room, for more detailed information the GetDescritiveInfo message should be used
   1. MaxOccupancy – Used to specify the total number of persons (Adults, Child, Infant) that are allowed in room;
   2. Numbers of units – used to specify available rooms for sell;
   3. RoomID – used to identify the room at Omnibees;
   4. Occupancy – Used to specify Adult min and max occupancy for adults and child.
3. RatePlanType – Used to specify information about the rateplan for a period of time based on checkin and checkout;
   1. MealsIncluded – Used to specify the board type associated with rate. When this object is null, then it means that rate has no meal included and the roomrate should be seen as room only.
   2. RatePlanInclusion – Used to specify the included extras
      1. Start – To specify that extra is valid from
      2. End – To specify that extra is valid to
      3. When start and end is null, then the extra is always valid.
   3. Commission – Used to inform the commission percentage to roomrate that has markup included
      1. Percent – This value is a decimal with 2 decimal places. The information about commission only applies to roomrate that has amount including markup = true.
   4. Guarantee – Used to specify deposit policies and allowed payment types.
      1. Start – Start date from this policy can be applied;
      2. End – End date for this policy can be applied;
      3. GuaranteeCode – Unique id for deposit policy at Omnibees;
      4. AmountPercent – Value of deposit required for reservation
         1. CurrencyCode – Currency code
         2. Amount – Amount in value
         3. NumberOfNights – The value of first n nights
         4. Percent – Percentagem of total

Note: Only Amount, NumberOfNights, Percent will be greater than zero.

* + 1. GuaranteeDescription –this information is a deposit policy and should be used as information proposes for the final client.
    2. GuaranteesAcceptedType – Omnibees will allowed only certain types of guarantee for reservation found on PMT codes. The reservation service will validate this object in realtime.
    3. When no GuaranteesAcceptedType is passed on HotelAvailRS then Omnibees will allow all PMT Codes.
  1. Payment Policy – this object is used to inform the possible payment types for reservation. The object GuaranteeTypeCode can be any type of PMT table: DirectBill, CreditCard, Deposit, Voucher, PrePay.
  2. Cancel Penalty – Used to inform the cancelation policy associated to rate for requested checkin and checkout dates. Just one and allways one cancelation policy is sent in HotelAvailRS
     1. Deadline – used to specify the number of days that cost will apply.
        1. OffsetunitMultipler – Used to specify number of days;
        2. TimeUnitType – Allways Day;
        3. OffsetDropTime – Allways BeforeArrival
     2. AmountPercentType – Used to specify cost of cancelation
        1. Amount – Used to specify value of cancelation;
        2. CurrencyCode – Used to specify currency code of amount of cancelation;
        3. NumberOfNights – Used to specify the number of nights that will be charged if cancelation will occur;
        4. Percent – Used to spsecify the percentage of amount that will be charged in case of cancelation;
        5. Just one of Amount, NumbrOfNight or Percent will be greater than 0. Use section 6.2 to get more information.
  3. TaxPolicy – Used to inform the tax policy applied. The value of tax are in amountaftertax tag.
     1. Value – Used to indicate da value of tax
     2. IsValuePercentage –When true the value should be seen as a percentage of amountbeforetax. When false the value should seen as a monetary value.
     3. Combination for Charging the value:
        1. Per room per night
           1. IsPerRoom = true
           2. IsPerPerson = false
           3. IsPerNight = true
           4. IsPerStay = false
        2. Per room per stay
           1. IsPerRoom = true
           2. IsPerPerson = false
           3. IsPerNight = false
           4. IsPerStay = true
        3. Per person per night
           1. IsPerRoom = false
           2. IsPerPerson = true
           3. IsPerNight = true
           4. IsPerStay = false
        4. Per person per stay
           1. IsPerRoom = false
           2. IsPerPerson = true
           3. IsPerNight = false
           4. IsPerStay = true
  4. AdditionalDetail – Used to specify other policies. This object will be used just for information proposues and have no impact on validation system at booking.
  5. Offers – Used to show the special offers included in rateplan. The totalbeforetax already has the price with offers on it. So no calculations are needed at client side. For the reservation system, Omnibees need to receive back the offers objects. Please use section 6.5 to check how offers are sent.
     1. **LastMinute Offer**
        1. OfferRule – Used to specify rule for the offer
           1. MaxAdvancedBookingOffset – Used to specify the number of maximum days before arrival that last minute offer can be applied;
           2. Daterestriction – Used to specify the date that the offer is valid. When this object is null, the offer is allways valid;
        2. Discount
           1. ChargeUnitCode – Allways Per room Per Stay
           2. Percent – Percentage of price to make a discount.
     2. **EarlyBooking Offer**
        1. OfferRule – Used to specify rule for the offer
           1. MinAdvancedBookingOffset – Used to specify the number of minimum of days in advanced before arrival that earlybooking offer can be applied;
           2. Daterestriction – Used to specify the date that the offer is valid. When this object is null, the offer is allways valid;
        2. Discount
           1. ChargeUnitCode – Allways Per room Per Stay
           2. Percent – Percentage of price to make a discount.
     3. **FreeNights**
        1. OfferRule – Used only to specify dates the offer is valid. If object is null, then the offer is allways valid.
        2. Discount
           1. ChargeUnitCode – Allways Per room Per Stay
           2. NightsDiscounted – Used to specify the number of night that will be offered;
           3. NightsRequired – Used to specify the number of nights needed to be booked for getting this offer;
           4. DiscountPattern – Used to specify the nights that will be offered. The string has the leght of the number of nights booked. 0’s means normal price and 1’s means nights offered. Example: 0001, booking for 4 nights with last night free.
     4. **Discount**
        1. OfferRule – Used to specify rule for the offer
           1. Daterestriction – Used to specify the date that the offer is valid. When this object is null, the offer is always valid;
        2. Discount
           1. ChargeUnitCode – Allways Per room Per Stay
           2. Percent – Percentage of price to make a discount.
     5. **Stay Discount**
        1. OfferRule – Used to specify rule for the offer
           1. Daterestriction – Used to specify the date that the offer is valid. When this object is null, the offer is always valid;
           2. LengthsOfStay – Used to specify the minimum and maximum length of stay that is required to apply the offer.

ArrivalDateBase – Allways is true, indicate that check in date is used to set da min/ma LOS;

MinMaxMessageType – Can be SetMinLOS for minimum length of stay, or SetMaxLOS for maxmim length of stay;

Time – indicate the number of nights that is requires for MLos/mLos

TimeUnit – Allways Day

* + - 1. Discount
         1. ChargeUnitCode – Allways Per room Per Stay
         2. Percent – Percentage of price to make a discount.

1. RoomRate – Used to specify room rate conditions
   1. RoomStayCandidateRPH- Used to specify the requested room. This RPH will match the RPH passed on request in RoomStayCandidate, meaning that this roomrate will satisfy the request for that room. Diferent RoomRates can have the same RatePlanID and RoomID, but the RoomStayCandidateRPH is different. This will happen because in Omnibees the price is per occupation so in a example with two different roomrate with equal roomID, equal RatePlanID and different RoomStayCandidateRPH can satisfy a request for two diferent occupation.
   2. EfetiveDate – Allways the checkin date;
   3. ExpireDate – Allways the checkout date;
   4. RatePlanID – used to specify rateplanID that can be cross referenced with RatePlan objects;
   5. RoomID – used to specify RoomID that can be cross referenced with RoomType objects;
   6. Discount – Used to specify a discount applied to price. This discount will be returned when a valid PromotionalCode is passed;
   7. RoomRateAvailability – Used to specify availability of room rate. Please check section Business Rule -> Rates that are not available for sell, to know the possible values;
      * 1. AvailabilityStatus – Used to indicate the availability status;
        2. WarningRPH – Used to cross reference with Warnings object;
   8. RatesType – Used to specify the day price with no tax
      1. Rate – Used to specify a single day
         1. EffectiveDate – Used to specify the day
         2. MaxLOS – Used to specify the maximum length of stay configured for this day;
         3. MinLOS – Used to specify the minimum length of stay configured for this day;
         4. NumberOfUnits – Used to specified available allotment at day level;
         5. StayThrough - Stay is not the same number of days that hotel expects;
         6. MinAdvancedBookingOffset – Release days make check in too close for book;
         7. Status – Used to indicate the status of day rate. Can be Close, ClosedOnArrival, ClosedToDeparture.
         8. Total – **RATE LEVEL TOTAL**: Used to specify the total for this day
            1. AmountAfterTax - Allways be 0 at this level;
            2. AmountBeforeTax – Use to specify the amount before tax for this day;
            3. AmountIncludingMarkup – When true means that price is a pvp price.
            4. AmountIsPackage – Indicates that price is a package price
            5. ChargeType – Allways per room per night
            6. TPA\_Extensions

Rate – Used to specify price per guest grouped by age

AgeQualifyingCode – Can be Adult, Child or Infant, according to request

EffectiveDate – date of this price

MaxGuestApplicable – Max Guest for total

MinGuestApplicable – Min Guest for total

MinAge – Min Age for this group

MaxAge – Max age for this group

Total

AmountAfterTax - Allways be 0 at this level;

AmountBeforeTax – Use to specify the amount before tax for this day;

AmountIncludingMarkup – When true means that price is a pvp price.

AmountIsPackage – Indicates that price is a package price

ChargeType – Allways Per person

* 1. ServiceRPH – used to specify allowed services for this roomrate. This RPH represents a unique id that can be cross referenced with Services Object.
  2. Total – **ROOM RATE LEVEL TOTAL**: Used to specify the total for the Rate sumatory.
     1. AmountAfterTax - Use to specify the amount after tax, the tax info can be found in RatePlan object;
     2. AmountBeforeTax – Use to specify the amount before tax for this day;
     3. AmountIncludingMarkup – When true means that price is a pvp price.
     4. AmountIsPackage – Indicates that price is a package price
     5. ChargeType – Allways per Per Room Per Stay
     6. TPA\_Extensions
        1. GuestTotalRate – Used to specify price total per guest age qualifying code
           1. AgeQualifyingCode – Can be Adult, Child or Infant, according to request
           2. Total

AmountAfterTax - Allways be 0 at this level;

AmountBeforeTax – Use to specify the amount before tax for this day;

AmountIncludingMarkup – When true means that price is a pvp price.

AmountIsPackage – Indicates that price is a package price

ChargeType – Allways Per Room Per Stay

1. Services – Used to specify optional extras tha can be choosed by the client. This services are not included in price. On reservation just need to send information about what extras are to apply. For the reservation the price cannot include Services price. The Omnibees service will make all total.
   1. ID – used to specify ID of the service at Omnibees system;
   2. ServiceRPH – Used to specify the unique id to be cross referenced with other objects that need information about the detail of service;
   3. ServicePriceType:
      * 1. PerRoomPerNight
        2. PerRoomPerStay
        3. PerPersonPerNight
        4. PerPersonPerStay
   4. ServiceCategoryCode: used to specify the category of service
      1. Null – Specify a general service;
      2. 1 – Specify a Breakfast service;
      3. 2 – Specify a HalfBoard service;
      4. 3 – Specify a FullBoard service;
   5. Price – used to specify the price of service
      1. Amount before tax used to specify the total price beforetax;
      2. Start – Used to specify the date from the price is valid, if it is null the price is allways valid;
      3. End - Used to specify the date to the price is valid, if it is null the price is allways valid;
2. Success – Used to specify that communication from service was hadled without errors. The success tag can be return with Warnings.
3. Warnings – Used to specify the request cannot be satisfied because of business rules. The descritption and code should be used to detect the problem.
4. Errors – Used to specify that the request cannot be satisfied because of an internal error. Echotoken should be send to Omnibees support to check what the problem was.

#### XML Responses

##### Single Hotel response with hotel ID with best only = true

The best rate for the hotel is returned

Files on directory HotelAvail: **Ex1RQ.xml, Ex1RS.xml**

##### Single Hotel response with hotel ID with best only = false

All the rates for that hotel are returned

Files on directory HotelAvail: **Ex2RQ.xml, Ex2RS.xml**

##### Multiple Hotel response with best only = true

Only best rate for hotel are returned.

Files on directory HotelAvail: **Ex3RQ.xml, Ex3RS.xml**

##### Multiple hotel response with best only = false

All rates that match criteria are on response.

Files on directory HotelAvail: **Ex4RQ.xml, Ex4RS.xml**

##### Single Hotel with one room stay candidate with quantity 2

The final price will be the total for quatity = 2. For the reservation the roomstay must be cloned in 2 (quantity) and the total must be splited by 2 (quantity). The roomstay on reservation represents one room in omnibees.

Files on directory HotelAvail: **Ex5RQ.xml, Ex5RS.xml**

##### Single Hotel with two room stay candidate

Request for two rooms with diferent occupation. The requestd room must be identified by a unique RPH so they can be cross referenced on response.

Files on directory HotelAvail: **Ex6RQ.xml, Ex6RS.xml**

##### Single hotel request and returned result package rate

Request for best rate in hotel. The response was a package rate.

Files on directory HotelAvail: **Ex7RQ.xml, Ex7RS.xml**

#### Other possible responses

Please use pulldev.omnibees.com to try different types of requests.

## Booking Service

### SendHotelRes

#### Description

This message allow partner to send a reservation after use message HotelAvail. To prevent errors on message the use of HotelAvail before SendHotelRes is recommended.

The basic flow to send a reservations is

1. Step One – Search with HotelAvail to get the last prices.
2. Step two – SendHotelRes
   1. The request must have TransactionActionType Book Or Hold (For on request).
   2. The request must have all data sent previously in HotelAvail is the response was for best rate only. (By not passing hotel ID)
   3. The request must have RatePlan object, RoomTypeObejct and RoomRateObject for the RoomRate chosen if more than one rate was sent in HotelAvail ( by passing the hotelID)
3. Step three – Confirm if reservation number is sent on response and if no warnings was sent.

**Please Note:** Each roomstay in reservation represents one room in Omnibees. For booking more than one room, Omnibees need to receive one roomstay per room.

The total of roomstays must be sent in ResGlobalInfo. This total will be the total of reservation in Omnibees.

#### OTA\_HotelResRQ

##### Schema

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Level | Occurrence | Element | Property | Mandatory (M)/ Optional (O) | Sample Usage |
| 0 | 1 | OTA\_HotelResRQ |  |  |  |
|  |  |  | Echotoken | M | c2ab0184-e9ac-4eb7-a9e1-7f1e8df2b66f |
|  |  |  | PrimaryLanID | O | En |
|  |  |  | TimeStamp | M | 2014-06-23T15:04:16.0294414Z |
|  |  |  | Target | M | Test |
|  |  |  | Version | M | 1 |
|  |  |  |  |  |  |
| 1 | 1 | HotelReservationsType |  | M |  |
| 2 | 1..100 | HotelReservationType |  | M |  |
|  |  |  |  |  |  |
| 3 | 1 | ResStatus |  | M |  |
|  |  |  | TransactionActionType | M | Must be allways book |
|  |  |  |  |  |  |
| 3 | 0..1 | POSType |  | O |  |
| 4 | 1 | Source |  | M |  |
|  |  |  | PseudoCityCode | O | TESTP |
|  |  |  | ISOCountry | O | PTR |
| 5 | 0..1 | RequestorID |  | O |  |
|  |  |  | ID | M | TESTP|ABB |
|  |  |  | Type | M | UID Code (Only 3 – CorporationRepresentative is valid) |
|  |  |  |  |  |  |
| 3 | 1 | RoomStaysType |  | M |  |
| 4 | 1..100 | RoomStay |  | M |  |
| 5 | 1 | **Guarantees** |  | O | Used to save the guarantee type and number for deposit |
| 6 | 1 | Guarantee |  | M |  |
| 6 | 1 | GuaranteesAcceptedType |  | M |  |
| 7 | 1 | GuaranteesAccepted |  | M |  |
| 8 | 1 | GuaranteeAccepted |  | M |  |
|  |  |  | GuaranteeID | M | AgencyName |
|  |  |  | GuaranteeTypeCode | M | TravelAgencyNameOrAddress |
|  |  |  |  |  |  |
| 5 | 1 | **RoomType** |  | M | Must be equal to object return on hotelavail response |
|  |  |  |  |  |  |
| 5 | 1 | **RatePlanType** |  | M | Must be equal to object return on hotelavail response |
|  |  |  |  |  |  |
| 5 | 1 | **GuestCountsType** |  | M |  |
| 6 | 1..20 | GuestCount |  | M |  |
|  |  |  | Age | O/M | Optional for Adult / Mandatory for child |
|  |  |  | AgeQualifyingCode | M | Adult (AQC Codes) |
|  |  |  | Count | M | 2 |
| 7 | 1..20 | ResGuestRPH | Int[] | M | 1 |
|  |  |  |  |  |  |
| 5 | 1 | **BasicPropertyInfo** |  | M |  |
| 6 | 1 | HotelRef |  | M |  |
|  |  |  | HotelCode | M | 1053 |
|  |  |  |  |  |  |
| 5 | 0..1 | **CommentsType** |  | O |  |
| 6 | 1..100 | Comment |  | M |  |
|  |  |  | Description | M | This is my room comment |
|  |  |  |  |  |  |
| 5 | 0..1 | **SpecialRequestType** |  | O |  |
| 6 | 1..100 | SpecialRequest |  | M |  |
|  |  |  | Description | M | This is my special request |
|  |  |  |  |  |  |
| 5 | 1 | **RoomRate** |  | M | Representes on Room in omnibees |
|  |  |  | EffectiveDate | M | Allways Checkin date |
|  |  |  | ExpireDate | M | Allways Checkout date |
|  |  |  | RatePlanID | M | 23454 |
|  |  |  | RoomID | M | 2852 |
|  |  |  | PromotionCode | O | 12CC |
|  |  |  | GroupCode | O | 76542 |
|  |  |  | Efective | M |  |
| 6 | 1 | RatesType |  | M | Used for day prices without tax |
| 7 | 1..N | Rate |  | M |  |
|  |  |  | EfectiveDate | M | 2014-05-12 |
|  |  |  | MaxLOS | O | 30 |
|  |  |  | MinLOS | O | 1 |
| 7 | 1 | Total |  | M | **Rate Level** |
|  |  |  | AmountBeforeTax | M | 25.00 |
|  |  |  | AmountAfterTax |  | Allways 0. Tax are not calculated at day level |
|  |  |  | AmountIncludingMarkup | M | True |
|  |  |  | AmountIsPackage | O | False |
|  |  |  | ChargeType | M | Allways PerRoomPerNight at day price Level |
|  |  |  | CurrencyCode | M | EUR (Currencies codes) |
|  |  |  |  |  |  |
| 6 | 0..N | Discount |  | O | Must be equal to object return on hotelavail response |
|  |  |  |  |  |  |
| 6 | 0..N | ServiceRPH |  | O | Used to identify Optional extras for this roomrate |
|  |  | RPH |  | M | 14552 |
|  |  |  |  |  |  |
| 6 | 1 | Total |  |  | **RoomRate Level,** Allways the sum of rate plus taxes |
|  |  |  | AmountBeforeTax | M | 20.00 |
|  |  |  | AmountAfterTax |  | 95.00 |
|  |  |  | AmountIncludingMarkup | M | True |
|  |  |  | AmountIsPackage | O | False |
|  |  |  | ChargeType | M | Allways PerRoomPerStay at roomrate price Level |
|  |  |  | CurrencyCode | M | EUR (Currencies codes) |
|  |  |  |  |  |  |
| 5 | 1..n | **Total** |  | M | **RoomStay Level** |
|  |  |  | AmountBeforeTax | M | 20.00 |
|  |  |  | AmountAfterTax | M | 95.00 |
|  |  |  | AmountIncludingMarkup | M | True |
|  |  |  | AmountIsPackage | O | False |
|  |  |  | ChargeType | M | Allways PerStay at roomStay price Level |
|  |  |  |  |  |  |
| 3 | 1 | ResGlobalInfo |  | M | Used to send information abou reservation |
| 4 | 0..1 | CommentsType |  | O |  |
| 5 | 1..100 | Comment |  | M |  |
|  |  |  | Description | M | This is my reservation comment |
|  |  |  |  |  |  |
| 4 | 0..1 | HotelReservationsIDsType |  | O | Used to indicate company on reservation |
| 5 | 1 | HotelReservationID |  | M |  |
|  |  |  | ResID\_Value | M | 12545 |
|  |  |  | ResID\_Type | M | Company or AssociatedReservation |
|  |  |  |  |  |  |
| 4 | 1 | Guarantee |  | M | Used to indicate Payment Type |
| 5 | 1 | GuaranteeAcceptedType |  | M |  |
| 6 | 1 | GuaranteeAccepted |  |  |  |
|  |  |  | PaymentType | M | CreditCard (PMT Codes) |
| 7 | 0..1 | PaymentCardType |  | O/M | Mandatory when PaymentType is CreditCard |
|  |  |  | CardHolderName | M | Jon |
|  |  |  | CardNumber | M | 1222122222222 |
|  |  |  | Expiredate | M | 2015-05-31 |
|  |  |  | SeriesCode | M | 123 |
| 4 | 1 | Total |  | M | **ResGlobalInfo Level** |
|  |  |  | AmountBeforeTax | M | 30.00 |
|  |  |  | AmountAfterTax | M | 75.00 |
|  |  |  | ChargeType | M | Allways PerStay at ResGlobalInfo price Level |
|  |  |  |  |  |  |
| 3 | 1 | ResGuestType |  | M |  |
| 4 | 1..40 | ResGuest |  | M |  |
|  |  |  | Age | O/M | Optional for Adult / Mandatory for child |
|  |  |  | AgeQualifyingCode | M | Adult (AQC Codes) |
|  |  |  | PrimaryGuest | M | False |
|  |  |  | ResGuestRPH | M | 1 |
| 5 | 1 | Profiles |  | M |  |
| 6 | 1 | ProfileInfo |  | M |  |
| 7 | 1 | Profile |  | M |  |
| 8 | 1 | Customer |  | M |  |
|  |  |  | BirthDate | O | 1988-05-01 |
|  |  |  | Email | O | jon@gmail.com |
| 9 | 1 | PersonNameType |  | M | Mandantory only for adults |
|  |  |  | NamePrefix | O | Mr (NPT Codes) |
|  |  |  | GivenName | M | Jon Bee |
|  |  |  | MiddleName | O | Muller |
|  |  |  | Surname | O | Bee |
| 9 | 0..1 | AddressType |  | O |  |
|  |  |  | AddressLine | O | Rua 25 |
|  |  |  | CityCode | O |  |
|  |  |  | CountryCode | O |  |
|  |  |  | PostalCode | O | 8600 |
|  |  |  | StateProvCode | O |  |
| 9 | 0..N | TelephoneInfo |  |  |  |
|  |  |  | Extension | O | 333 |
|  |  |  | PhoneNumber | O | 65245214 |
|  |  |  | PhoneTechType | O | Mobile (PTT Codes) |
| 9 | 0..1 | Document |  |  |  |
|  |  |  | DocID | O | 1284547562 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| 1 | 0..N | Service |  | O |  |
|  |  |  | ID | M | 14552 |
|  |  |  | ServiceCategoryCode | O | Null |
|  |  |  | ServicePricingType | M | PerRoomPerNight (CHG Codes) |
|  |  |  | ServiceRPH | M | 14552 |
|  |  |  | Quantity | O | null |
|  |  |  | RatePlanCode | M | 12343 |
|  |  |  | RequestIndicator | M | False |
| 2 | 1 | ServiceDescription |  | M |  |
|  |  |  | Name | M | Cleaning Service |
|  |  |  | Description | O | Service Description |
|  |  |  | Language | M | en (Languages Codes) |
|  |  |  | ImageURL | O | Url of image |
| 2 | 0..1 | TPA\_Extensions |  | O |  |
| 3 | 0..N | Schedules |  | M |  |
|  |  |  | Date | M | 2015-05-06 9:00 |

Table 9 - OTA\_HotelResRQ schema

##### General Rules

1. Echotoken – A Guid must be sent in every request. Must be unique per request.
2. PrimaryLangID – Indicates the language that responses should be sent
3. Timestamp – Must send allways the date now in utc time.
4. Version – Version of the service.
5. HotelReservation – The service can receive multiple reservation in one time
6. POS
   1. Source – Will have the information about agency PCC. The GetAgencies Method can be used to get all agencies codes.
   2. ISOCountry - The country code of the requesting party in ISO 3166-3 encoding.
   3. RequestorID (only will be used if the request is from a agency post in a company)
      1. Type – Must be allways CorporationRepresentative
      2. ID – must be agency office pcc.
7. ResStatus
   1. TransactionActionType – Must be allways book, this will inform the service the intention of booking. Must be allways Hold, for On Request reservations.
8. RoomStay
   1. Total – **ROOM STAY LEVEL TOTAL**: Used to specify the total for the best RoomRate found.
      1. AmountAfterTax - Use to specify the amount after tax, the tax info can be found in RatePlan object;
      2. AmountBeforeTax – Use to specify the amount before tax for this day;
      3. AmountIncludingMarkup – When true means that price is a pvp price;
      4. AmountIsPackage – Indicates that price is a package price
   2. Deposit Payments – Used to specify the deposit guarantee for this roomstay
      1. GuaranteeID – the value for guarantee code, example is iata number, agency name, etc. This field must be sent null when GuaranteeTypeCode is credit card. The payment for reservation must be credit card.
      2. GuaranteeTypeCode – Code for Guarantee. Must be one sent in HotelAvailRS.
   3. RoomType – Must be an equal object from the roomtype object returned in hotelavail
   4. RatePlanType – Must be an equal object from the rateplantype object returned in hotelavail
   5. GuestCount – Used to specify the guest counting by room. The service will check this object and cross reference with Guest objects to get information about the guest.
      1. ResGuestRPH – Used to specify Guests ID (RPH) to cross reference with Guests Object.
   6. BasicPropertyInfo
      1. HotelRef
         1. HotelCode – Used to specify the hotelcode for the reservation. For each roomstay in same reservation, the HotelCode must be the same;
   7. Comment – Used to specify a comment to this room stay
   8. Special request- used to specify a special request;
   9. RoomRate – Used to specify the roomrate conditions.
      1. Only one roomrate is allowed per roomstay.
      2. The service will allow higher prices than prices returned in hotel avail, but will not allow lower prices.
      3. EfetiveDate – Allways the checkin date;
      4. ExpireDate – Allways the checkout date;
      5. RatePlanID – used to specify rateplanID that can be cross referenced with RatePlan objects;
      6. RoomID – used to specify RoomID that can be cross referenced with RoomType objects;
      7. Discount – Should be the equal to discount on response on hotelavail request;
      8. RatesType – Used to specify the day price with no tax
         1. Rate – Used to specify a single day
            1. EffectiveDate – Used to specify the day
            2. Total – **RATE LEVEL TOTAL**: Used to specify the total for this day

AmountAfterTax - Allways be 0 at this level;

AmountBeforeTax – Use to specify the amount before tax for this day;

AmountIncludingMarkup – When true means that price is a pvp price;

AmountIsPackage – Indicates that price is a package price;

* + 1. ServiceRPH – used to specify chosen services for this roomrate. This RPH represents a unique id that can be cross referenced with Services Object.
    2. Total – **ROOM RATE LEVEL TOTAL**: Used to specify the total for the Rate sumatory.
       1. AmountAfterTax - Use to specify the amount after tax, the tax info can be found in RatePlan object;
       2. AmountBeforeTax – Use to specify the amount before tax for this day;
       3. AmountIncludingMarkup – When true means that price is a pvp price;
       4. AmountIsPackage – Indicates that price is a package price

1. ResGlobalInfo – Used to specify information about the reservation
   1. HotelReservationIDsType – Used to specify company of reservation
      1. HotelReservationID
         1. ResID\_Value – Id of a company. This codes can be obtained in GetCorporates.
         2. ResID\_Type – Can be:
            1. Companhy unique identifier (this id is Omnibees company id, to identify what company is associated with reservation);
            2. AssociatedReservation for partner reservation number;
2. Guarantee
   1. GuaranteeAccepted – Used to specify the type of payment
      1. GuaranteeTypeCode – This is information will inform the hotel what type of payment should consider for reservation. When CreditCard is used the PaymentCardType object is mandatory;
      2. PaymentCardType – Used to specify the credit card values
3. ResGuest – Used to specify the information about the guest
   1. Omnibees only saves the information about the primary guest;
   2. Other Guests that are not primary only need to send the names;
   3. The children must have defined the ages;
   4. The children names are optional
4. Services – Used to specify optional extras tha can be choosed by the client. This services are not included in price. On reservation just need to send information about what extras are to apply. For the reservation the price cannot include Services price. The Omnibees service will make all total.
   1. ID – used to specify ID of the service at Omnibees system;
   2. ServiceRPH – Used to specify the unique id to be cross referenced with other objects that need information about the detail of service;
   3. ServicePriceType:
      * 1. PerRoomPerNight
        2. PerRoomPerStay
        3. PerPersonPerNight
        4. PerPersonPerStay
   4. ServiceCategoryCode: used to specify the category of service
      1. Null – Specify a general service;
      2. 1 – Specify a Breakfast service;
      3. 2 – Specify a HalfBoard service;
      4. 3 – Specify a FullBoard service;
   5. Price – used to specify the price of service
      1. Amount before tax used to specify the total price beforetax;
      2. Start – Used to specify the date from the price is valid, if it is null the price is allways valid;
      3. End - Used to specify the date to the price is valid, if it is null the price is allways valid;
   6. Quantity – The number of service to apply. Only spplies on per stay prices;
   7. RatePlanCode – The representation of the rate plan under which this service was booked;
   8. RequestIndicator – When true, the service has been requested by the guest;
   9. Schedule – Used to specify the time the service should be delivered
      1. Date – The starting value of the time span.

##### Supported Codes

|  |  |  |  |
| --- | --- | --- | --- |
| OTA Code List Name | Code Value | Code Name | Where Used |
|  | 0 | Test | Target, to specify wich enviroment request is destinated |
|  | 1 | Production | Target, to specify wich enviroment request is destinated |
|  |  |  |  |
|  | 0 | IATA | CodeContext, To spefify that location is an IATA code |
|  |  |  |  |
| UOM | 1 | Miles | RadiusType |
| UOM | 2 | Kilometers | RadiusType |
| UOM | 3 | Meters | RadiusType |
| UOM | 6 | Yards | RadiusType |
| UOM | 7 | Feet | RadiusType |
|  |  |  |  |
| UID | 3 | CorporationRepresentative | UniqueID, to specify that request is from an agency post |
| UID | 4 | Company | UniqueID, to specifiy that request is from a Company |
|  |  |  |  |
| AQC | 7 | Infant | GuestCount |
| AQC | 8 | Child | GuestCount |
| AQC | 10 | Adult | GuestCount |
|  |  |  |  |
| RT | 0 | MinLOS | Restriction |
| RT | 1 | MaxLOS | Restriction |
| RT | 2 | FixedLOS | Restriction |
| RT | 3 | MinAdvancedBook | Restriction |
|  |  |  |  |
| TU | 0 | Day | Restriction, DeadLine |
|  |  |  |  |
| CHG | 18 | PerRoomPerStay | ChargeType, ChargeUnitCode |
| CHG | 19 | PerRoomPerNight | ChargeType, ChargeUnitCode |
| CHG | 12 | PerStay | ChargeType, ChargeUnitCode |
| CHG | 7 | PerPerson | ChargeType, ChargeUnitCode |
| CHG | 99 | PerNight | ChargeType, ChargeUnitCode |
| CHG | 25 | PerRoom | ChargeType, ChargeUnitCode |
| CHG | 20 | PerPersonPerStay | ChargeType, ChargeUnitCode |
| CHG | 21 | PerPersonPerNight | ChargeType |
|  |  |  |  |
| MPT | 1 | AllInclusive | MealIncluded |
| MPT | 3 | BedBreakfast | MealIncluded |
| MPT | 10 | FullBoard | MealIncluded |
| MPT | 12 | HalfBoard | MealIncluded |
|  |  |  |  |
| PMT | 2 | DirectBill | GuaranteeTypeCode |
| PMT | 5 | CreditCard | GuaranteeTypeCode |
| PMT | 8 | Deposit | GuaranteeTypeCode |
| PMT | 3 | Voucher | GuaranteeTypeCode |
| PMT | 4 | PrePay | GuaranteeTypeCode |
|  |  |  |  |
| TST | 1 | Book | TransactionType |
|  |  |  |  |
| CCT | 1 | Visa | PaymentCardType |
| CCT | 2 | MasterCard | PaymentCardType |
| CCT | 3 | AmericanExpress | PaymentCardType |
| CCT | 4 | Discover | PaymentCardType |
| CCT | 5 | HiperCard | PaymentCardType |
| CCT | 6 | DinnersClub | PaymentCardType |
| CCT | 7 | JCB | PaymentCardType |
| CCT | 8 | Electron | PaymentCardType |
| CCT | 9 | EuroCard | PaymentCardType |
| CCT | 10 | EnRouteCard | PaymentCardType |
| CCT | 11 | Laser | PaymentCardType |
| CCT | 12 | Maestro | PaymentCardType |
| CCT | 13 | Solo | PaymentCardType |
| CCT | 14 | PayPal | PaymentCardType |
| CCT | 15 | Aura | PaymentCardType |
| CCT | 16 | Liberate | PaymentCardType |
|  |  |  |  |
| GT | 0 | Male | Customer |
| GT | 1 | Female | Customer |
|  |  |  |  |

Table 10 - Supported codes for OTA\_HotelResRQ

##### Class Diagram Zip Image



#### OTA\_HotelResRS

This object should basicly the same object that is sent in request. The only changes are at ResStatus a UniqueID objects.

##### Schema

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Level | Occurrence | Element | Property | Mandatory (M)/ Optional (O) | Sample Usage |
| 0 | 1 | OTA\_HotelResRS |  |  |  |
| 1 | 1 | HotelReservationsType |  | M |  |
| 2 | 1..100 | HotelReservationType |  | M |  |
| 3 | 1 | ResStatus |  | M |  |
|  |  |  | PMS\_ResStatusType | M | Reserved |
| 3 | 0..1 | UniqueID |  | M |  |
|  |  |  | ID | M | Res000001254-1635 |
|  |  |  | Type | M | Allways Reservation |
|  |  |  |  |  |  |
| 1 | 0..1 | Sucess |  | O |  |
|  |  |  |  |  |  |
| 1 | 0..1 | WarningsType |  | O |  |
| 2 | 1..N | Warning |  | M |  |
|  |  |  | Code | M | 450 |
|  |  |  | Language | M | En |
|  |  |  | ShortText | M | Warning Text |
|  |  |  |  |  |  |
| 1 | 0..1 | ErrorsType |  | O |  |
| 2 | 1..N | Error |  | M |  |
|  |  |  | Code | M | 448 |
|  |  |  | Language | M | en |
|  |  |  | ShortText | M | System error: Internal system error. Please contact techincal support at integration.team@omnibees.com { echoToken = 57fde0db-32ed-40c5-bf88-e19a7601472a } |
|  |  |  |  |  |  |

Table 11 - OTA\_HotelResRS schema

##### General Rules

1. HotelReservationType – Used to specify the request object and to inform the reservation status at Omnibees and the reservation number;
   1. UniqueID – Used to specify the reservation number
      1. ID – used to specify reservation number at Omnibees;
      2. Type – The type for Reservation number is reservation;
   2. ResStatus – Used to specify status of reservation at Omnibees
      1. PMS\_ResStatus – Current status of reservation
2. Success – Used to specify that communication from service was hadled without errors. The success tag can be return with Warnings.
3. Warnings – Used to specify the warnings that can be information or that cannot be satisfied. The descritption and code should be used to detect the problem. **PLEASE NOTE:** **The partner should not use exitent warnings to assume that reservation can not be done**. For check if reservation was successful please use ResStatus on response.
4. Errors – Used to specify that the request cannot be satisfied because of an internal error. Echotoken should be send to Omnibees support to check what the problem was.

##### Supported Codes

|  |  |  |  |
| --- | --- | --- | --- |
| OTA Code List Name | Code Value | Code Name | Where Used |
|  | 0 | Test | Target, to specify wich enviroment request is destinated |
|  | 1 | Production | Target, to specify wich enviroment request is destinated |
|  |  |  |  |
| UID | 3 | Reservation | UniqueID, to specify Reservation number |
|  |  |  |  |
|  | 1 | Reserved | PMS\_ResStatusType |

Table 12 - Supported codes for OTA\_HotelResRS

##### Class Diagram Zip Image



#### XML Examples

(Just one hotel was used for this example)

##### Booking for one Adult with one room (only best rates)

###### Step One – Check availability without passing hotel id

Files on directory HotelRes: Ex1StepOneRQ.xml, Ex1StepOneRS.xml

###### Step two – SendHotelRes

Files on directory HotelRes: Ex1StepTwoRQ.xml, Ex1StepTwoRS.xml

##### Booking two adult one child

###### Step One – Check availability without passing hotel id

Files on directory HotelRes: Ex2StepOneRQ.xml, Ex2StepOneRS.xml

###### Step two – SendHotelRes

Files on directory HotelRes: Ex2StepTwoRQ.xml, Ex2StepTwoRS.xml

##### Booking one adult (passing hotel ID)

###### Step One – Check availability passing hotel id

Files on directory HotelRes: Ex3StepOneRQ.xml, Ex3StepOneRS.xml

###### Step two – SendHotelRes

Files on directory HotelRes: Ex3StepTwoRQ.xml, Ex3StepTwoRS.xml

##### Booking by agency (passing Agency PCC)

###### Step One – Check availability

Files on directory HotelRes: Ex4StepOneRQ.xml, Ex4StepOneRS.xml

###### Step two – SendHotelRes

Files on directory HotelRes: Ex4StepTwoRQ.xml, Ex4StepTwoRS.xml

##### Booking to corporate (passing Corporate Code)

###### Step One – Check availability

Files on directory HotelRes: Ex5StepOneRQ.xml, Ex5StepOneRS.xml

###### Step two – SendHotelRes

Files on directory HotelRes: Ex5StepTwoRQ.xml, Ex5StepTwoRS.xml

##### Booking using Agency to corporate (passing Agency PCC and Corporate Code)

###### Step One – Check availability

Files on directory HotelRes: Ex6StepOneRQ.xml, Ex6StepOneRS.xml

###### Step two – SendHotelRes

Files on directory HotelRes: Ex6StepTwoRQ.xml, Ex6StepTwoRS.xml

##### Booking two rooms with same rateID and roomID by searching in hotelavail with 2 on quatity of rooms

###### Step One – Check availability

Files on directory HotelRes: Ex7StepOneRQ.xml, Ex7StepOneRS.xml

###### Step two – SendHotelRes

Files on directory HotelRes: Ex7StepTwoRQ.xml, Ex7StepTwoRS.xml

##### Booking two rooms with same rateID and roomID by searching in hotelavail with 2 roomstaycandidate

###### Step One – Check availability

Files on directory HotelRes: Ex8StepOneRQ.xml, Ex8StepOneRS.xml

###### Step two – SendHotelRes

Files on directory HotelRes: Ex8StepTwoRQ.xml, Ex8StepTwoRS.xml

##### Booking two rooms with diferent rateID and roomID by searching in hotelavail with 2 roomstaycandidate

###### Step One – Check availability

Files on directory HotelRes: Ex9StepOneRQ.xml, Ex9StepOneRS.xml

###### Step two – SendHotelRes

Files on directory HotelRes: Ex9StepTwoRQ.xml, Ex9StepTwoRS.xml

##### Booking a package rate

###### Step One – Check availability

Files on directory HotelRes: Ex10StepOneRQ.xml, Ex10StepOneRS.xml

###### Step two – SendHotelRes

Files on directory HotelRes: Ex10StepTwoRQ.xml, Ex10StepTwoRS.xml

### SendHotelResModify

#### Description

The SendHotelResModify message is used to modify a reservation that was previously confirmed. In the request, partner will send the relevant reservation number generated by Omnibees at previous confirmed reservation, plus all other information pertinent to the reservation even though only a subset of it may be “modified”.

Modify is a two-step process: Initiate and then commit. The ResStatus attribute is used to indicate the type of request. Additionally for consecutive modify transactions of the confirmed reservation partner will use ResStatus=’Modify’.

This two-step process allow partner to retry messages in case of time out, or other problem in communication.

**NOTE: Omnibees only allow 3 retries messages at time, a retry message has to send RetransmissionIndicator = true. After 3 retries the transaction will be blocked**. If partner retry a message with RetransmissionIndicator = false, transaction will fail and ResResponseType will be Unsucefull.

For certain types of Modify transactions, partner may use alternative backup functionality to deliver transactions that are not successfully delivered Omnibees.

The response for initiate and modify will be the reservation. For commit or ignore the response only has the status for transaction. In case on fail the response will have the warning or error related to the problem occurred.

To get more information about the request and response object please check objects at <http://pullcert.omnibees.com/webapi/Help/Api/POST-api-Pull-SendHotelResModify>.

#### How to

For modify a reservation there is a few steps need:

1. Use hotel avail to search prices and availability for modify a reservation. Please view section regarding request for hotel availability. Searching for modify a reservation will require the following fields to be filled in OTA\_HotelAvailRQ:
   * 1. IsModify – This field must be sent true;
     2. Profile Unique ID – Unique id type must be sent Reservation and ID must be filled with Omnibees reservation number;
     3. BookingCode – When change is for a room BookingCode inside roomstaycandidate must be filled;
2. Initiate the Reservation request
3. Check for response. The response will have all data regarding the reservation;
4. Commit the modification.

#### Allowed Reservation Modifications

Omnibees recommend that partner send a Hotel Avail request to check if new change to reservation can be done.

Allowed modification for reservation are:

1. Check in/Check out date – Partner can modify stay period;
2. Room Type – The change of a room is allowed, but the rateplan must be the same;
3. Room occupation – The partner can change the occupation of a room;
4. Add new room – Adding a new room with a new rateplan is allowed;
5. Primary Guest Name – Partner can change the primary guest name for reservation;
6. Room Guest Name – Changing the name of the guest in each room is allowed;

#### Not Allowed Reservation Modification

Some modification are not allowed:

1. Rate – Partner cannot change the rate for one room. The work around is cancel the room and add a new one. This can only be done if Rate has no cancelation costs or deposit associated at modification date;
2. Services (Extras) – Partner cannot change the services associated with room;
3. Meal Plan – Meal plan is automatic reconfigured according to allowed change;
4. Change a room for a check in date before the current data;
5. Change a room when cancelation cost apply;
6. Chance a room when deposit value is required;
7. If stay date do not fill the required restrictions;

#### Request Message Status

|  |  |  |
| --- | --- | --- |
| Action | Property | Property Value |
| Request a Modify to a confirmed Booking | ResStatus | Initiate |
| Request a consecutive Modify to a confirmed Booking | ResStatus | Modify |
| Ignore a Modify to a Committed Booking | ResStatus | Ignore |
| Commit a Modify to a Committed Booking | ResStatus | Commit |

Table 13 - OTA\_HotelResModifyRQ transaction ResStatus

#### Response Message Status

|  |  |  |
| --- | --- | --- |
| Action | Property | Property Value |
| Request a Modify to a confirmed Booking | ResResponseType | Pending |
| Request a consecutive Modify to a confirmed Booking | ResResponseType | Pending |
| Ignore a Modify to a Committed Booking | ResResponseType | Ignored |
| Commit a Modify to a Committed Booking | ResResponseType | Committed |

Table 14 - OTA\_HotelResModifyRS transaction ResStatus

#### Message Flows diagrams

##### Initiate-> Modify (Optional) -> End with Commit or Ignore



Figure 13 - Transaction - Initiate-> Modify (Optional) -> End with Commit or Ignore

#### Type B Messages

##### Initiate -> Time out on initiate



Figure 14 - Initiate -> Time out on initiate

##### Initiate -> Modify -> Time out on Modify



Figure 15 - Initiate -> Modify -> Time out on Modify

##### Initiate -> Commit -> Time out on commit



Figure 16 - Initiate -> Commit -> Time out on commit

##### Initiate -> Ignore -> Time out on ignore



Figure 17 - Initiate -> Ignore -> Time out on ignore

#### Unsuccessful ResResponseType

Unsuccessful responses are sent when something is wrong with request or transaction. The partner has to begin all transactions with initiate and end commit or ignore. The following examples show messages flows that do not follow the excepted order of transactions requests.

##### Initiate -> Initiate (RetransmissionIndicator = false)



Figure 18 - Initiate -> Initiate (RetransmissionIndicator = false)

##### Initiate -> Commit -> Commit (RetransmissionIndicator = false)



Figure 19 - Initiate -> Commit -> Commit (RetransmissionIndicator = false)

##### Initiate -> Ignore -> Ignore (RetransmissionIndicator = false)



Figure 20 - Initiate -> Ignore -> Ignore (RetransmissionIndicator = false)

##### Commit (Without previous initiate)



Figure 21 - Commit (Without previous initiate)

#### Schema

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Level | Occurrence | Element | Property | Mandatory (M)/ Optional (O) | Sample Usage |
| 0 | 1 | OTA\_HotelResModifyRQ |  |  |  |
|  |  |  | Echotoken | M | c2ab0184-e9ac-4eb7-a9e1-7f1e8df2b66f |
|  |  |  | PrimaryLanID | O | En |
|  |  |  | TimeStamp | M | 2014-06-23T15:04:16.0294414Z |
|  |  |  | Target | M | Test |
|  |  |  | Version | M | 1 |
|  |  |  | ResStatus | M | Initiate |
|  |  |  | RetransmissionIndicator | O | True |
|  |  |  | TransactionIdentifier | M | c2ab0184-e9ac-4eb7-a9e1-7f1e8df2b66f |
|  |  |  |  |  |  |
| 1 | 1 | HotelResModifiesType |  | M |  |
| 2 | 1..100 | HotelReservation |  | M |  |
|  |  |  |  |  |  |
| 3 | 1 | RoomStaysType |  | M |  |
| 4 | 1..100 | RoomStay |  | M |  |
|  |  |  | IndexNumber | O | Res000125/1 |
|  |  |  |  |  |  |
| 5 | 1 | **GuestCountsType** |  | M |  |
| 6 | 1..20 | GuestCount |  | M |  |
|  |  |  | Age | O/M | Optional for Adult / Mandatory for child |
|  |  |  | AgeQualifyingCode | M | Adult (AQC Codes) |
|  |  |  | Count | M | 2 |
| 7 | 1..20 | ResGuestRPH | Int[] | M | 1 |
|  |  |  |  |  |  |
| 5 | 1 | **RoomRate** |  | M | Representes on Room in omnibees |
|  |  |  | EffectiveDate | M | Allways Checkin date |
|  |  |  | ExpireDate | M | Allways Checkout date |
|  |  |  | RatePlanID | M | 23454 |
|  |  |  | RoomID | M | 2852 |
|  |  |  |  |  |  |
| 4 | 0..1 | UniqueID |  | O | Used to indicate company on reservation |
|  |  |  | ResID\_Value | M | Res000125 |
|  |  |  | ResID\_Type | M | Reservation |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| 3 | 1 | ResGuestType |  | M |  |
| 4 | 1..40 | ResGuest |  | M |  |
|  |  |  | Age | O/M | Optional for Adult / Mandatory for child |
|  |  |  | AgeQualifyingCode | M | Adult (AQC Codes) |
|  |  |  | PrimaryIndicator | M | False |
|  |  |  | ResGuestRPH | M | 1 |
| 5 | 1 | Profiles |  | M |  |
| 6 | 1 | ProfileInfo |  | M |  |
| 7 | 1 | Profile |  | M |  |
| 8 | 1 | Customer |  | M |  |
| 9 | 1 | PersonNameType |  | M | Mandantory only for adults |
|  |  |  | GivenName | M | Jon Bee |
|  |  |  | MiddleName | O | Muller |
|  |  |  | Surname | O | Bee |

Table 15 - OTA\_HotelResModifyRQ schema

#### General Rules

1. Echotoken – A Guid must be sent in every request. Must be unique per request.
2. PrimaryLangID – Indicates the language that responses should be sent
3. Timestamp – Must send always the date now in utc time.
4. Version – Version of the service.
5. ResStaus – Indicate the action request for transaction. Can be one of the following:
   1. Initiate – Used to initiate the transaction. The response will allow partner to confirm all field that will be saved on Omnibees.
   2. Modify – used to modify a reservation inside the current transaction. Only can be used after initiate.
   3. Commit – Used to end the transaction and save modifications to reservation.
   4. Ignore – Used to end the transaction and ignore modifications to reservation.
6. RestrasmissionIndicator – When true indicate that the message is a retry message. This field can only be true after one request for each ResStaus. This field can only be used for retries a message after something fail, for example a timeout.
7. TransactionIdentifier – This field is used to identify the transaction, the transaction id must unique for each reservation, but as to be the same for all res status inside the reservation number. For example:
   1. Valid Example
      1. ResNumber(55) => Initiate (transactionidentifier A) -> Commit(A) -> Initiate(A) -> Modify(A) -> Commit(A)
      2. ResNumber(66) => Initiate (transactionidentifier B) -> Commit(B) -> Initiate(B) -> Modify(B) -> Commit(B)
   2. Invalid Example
      1. ResNumber(55) => Initiate (transactionidentifier A) -> Commit(A) -> Initiate(B)
      2. ResNumber(66) => Initiate (transactionidentifier A)
8. RoomStay
   1. IndexNumber – Used to specify the Ominbees room number when this roomstay is for modify an existent room.
   2. GuestCount – Used to specify the guest counting by room. The service will check this object and cross reference with Guest objects to get information about the guest.
      1. ResGuestRPH – Used to specify Guests ID (RPH) to cross reference with Guests Object.
   3. RoomRate – Used to specify the roomrate conditions.
      1. Only one roomrate is allowed per roomstay.
      2. EfetiveDate – Allways the checkin date;
      3. ExpireDate – Allways the checkout date;
      4. RatePlanID – used to specify rateplanID;
      5. RoomID – used to specify RoomID;
9. ResGuest – Used to specify the information about the guest
   1. Omnibees only saves the information about the primary guest;
   2. Other Guests that are not primary only need to send the names;
   3. The children must have defined the ages;
   4. The children names are optional
10. UniqueID – Used to send the reservation Number to modify
    1. ID – used to indicate the Omnibees reservation number;
    2. Type – Can only be Reservation

#### Class Diagram



#### Examples

Please use Pull web app at <http://pulldev.omnibees.com> to get xml examples.

### SendHotelResCancel

#### Description

This message allow partner to cancel one booking for reservation number. To check authenticity of message a Verification field is required. Message detail to get this information can be found at messages section. Then verification fields are available in response at get bookings method.

On response the object is almost the same that was received on Reservation request. The service will have the information about costs of cancelation, and the status of the reservation.

It is possible to check on cancelation costs without cancel the booking. For check only canceltaion cost the canceltype = VerifyPrices is required to send on request.

To cancel a reservation, the following fields are required:

1. Res number – indicates the reservation number;
2. Min Check In – indicates the minimum date found on all rooms in reservation;
3. Max check out – indicates the maximum date found on all rooms in reservation;
4. Hotel code – indicates hotel id;
5. Email – indicates the email on reservation guest with primary indicator;
6. Agency PCC – (optional) if reservation has agency PCC this field is required;
7. Company Code – (optional) if reservation has company code this field is required;
8. ComporationRepresentative – (optional) if reservation has agecy office code this field is required;

Example of a request:

<POS>

<Sources>

<Source>

<PseudoCityCode>TAFAO</PseudoCityCode> **Optional field: Agency PCC**

</Source>

</Sources>

</POS>

<PrimaryLangID>en</PrimaryLangID>

<Target>Test</Target>

<TimeStamp>2013-07-11T00:00:00Z</TimeStamp>

<UniqueID>

<UniqueID>

<ID>CRP</ID> **Optional field: Company code**

<Type>Company</Type>

</UniqueID>

<UniqueID>

<ID>RES000094-1814</ID> **Res number field**

<Type>Reservation</Type>

</UniqueID>

</UniqueID>

<Verification>

<Email>[bee@omnibees.com</Email](mailto:bee@omnibees.com%3c/Email)> **Guest email**

<HotelRef>

<ChainCode xsi:nil="true" />

<HotelCode>1814</HotelCode> **Hotel Code**

</HotelRef>

<ReservationTimeSpan>

<End>2013-07-20T00:00:00</End> **Reservation Max checkout date**

<Start>2013-07-18T00:00:00</Start> **Reservation min checkin date**

</ReservationTimeSpan>

</Verification>

<Version>1</Version>

#### OTA\_CancelRQ

##### Schema

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Level | Occurrence | Element | Property | Mandatory (M)/ Optional (O) | Sample Usage |
| 0 | 1 | OTA\_CancelRQ |  |  |  |
|  |  |  | Echotoken | M | c2ab0184-e9ac-4eb7-a9e1-7f1e8df2b66f |
|  |  |  | PrimaryLanID | O | En |
|  |  |  | TimeStamp | M | 2014-06-23T15:04:16.0294414Z |
|  |  |  | Target | M | Test |
|  |  |  | Version | M | 1 |
|  |  |  | CancelType | O | VerifyPrice or CancelRoom |
|  |  |  |  |  |  |
| 1 | 1 | Segment |  | O |  |
| 1 | 0..1 | Segment |  | O |  |
|  |  |  | ItinsegNbr | M | Res000001254-1635/2 |
|  |  |  | Status | M | Allways Cancelled |
|  |  |  |  |  |  |
| 1 | 0..1 | POSType |  | O |  |
| 2 | 1 | Source |  | M |  |
|  |  |  | PseudoCityCode | O | TESTP |
| 3 | 0..1 | RequestorID |  | O |  |
|  |  |  | ID | M | TESTP|ABB |
|  |  |  | Type | M | UID Code (Only 3 – CorporationRepresentative is valid) |
|  |  |  |  |  |  |
| 1 | 1 | UniqueID |  | M |  |
|  |  |  | ID | M | Res000001254-1635 |
|  |  |  | Type | M | Allways Reservation |
| 1 | 0..1 | UniqueID |  | O |  |
|  |  |  | ID | M | ABB |
|  |  |  | Type | M | Allways Company |
|  |  |  |  |  |  |
| 1 | 1 | Verification |  | M |  |
|  |  |  | Email | O | [support@visualforma.pt](mailto:support@visualforma.pt) |
| 2 | 1 | HotelRef |  | M |  |
|  |  |  | HotelCode | M | 1053 |
| 2 | 1 | ReservationTimeSpan |  | M |  |
|  |  |  | Start | M | 2014-05-01 |
|  |  |  | End | M | 2014-05-15 |
|  |  |  |  |  |  |

Table 16 - OTA\_CancelRQ schema

##### GeneralRules

1. Echotoken – A Guid must be sent in every request. Must be unique per request.
2. PrimaryLangID – Indicates the language that responses should be sent
3. Timestamp – Must sent allways the date now in utc time.
4. Version – Version of the service.
5. CancelType – When VerifyPrice is sent the cancel request take no action on reservation. This is used only to know the cost for the cancelation.

When CancelRoom is sent the cancel request will cancel a specific room from a reservation.

1. Segment
   1. Optional (only used if CancelType is CancelRoom)
   2. ItinSegNbr – Must have the reservation number / room.
   3. Status – will change status of the room to Cancelled.
2. POS
   1. Optional (only used is sent on reservation)
   2. Source – Will have the information about agency PCC. The GetAgencies Method can be used to get all agencies codes.
   3. RequestorID (only will be used if the request is from a agency post in a company)
      1. Type – Must be allways CorporationRepresentative
      2. ID – must be agency office code
3. UniqueID (For reservation number)
   1. The unique id for reservation number is mandatory;
   2. ID – used to specify reservation number
   3. Type – allways Reservation
4. UniqueID (For company code)
   1. Optional (only used is sent on reservation)
   2. ID – used to specify company code
   3. Type – allways Company
5. Verification – Used to validate data on reservation
   1. Email – indicates the email on reservation guest with primary indicator;
   2. ReservationTimeSpan
      1. Start – indicates the minimum date found on all rooms in reservation;
      2. End – indicates the maximum date found on all room in reservation;
   3. HotelRef
      1. HotelCode – indicates hotel id;

##### Supported codes

|  |  |  |  |
| --- | --- | --- | --- |
| OTA Code List Name | Code Value | Code Name | Where Used |
|  | 0 | Test | Target, to specify wich enviroment request is destinated |
|  | 1 | Production | Target, to specify wich enviroment request is destinated |
|  |  |  |  |
| UID | 3 | CorporationRepresentative | UniqueID, to specify that request is from an agency office |
| UID | 4 | Company | UniqueID, to specifiy that request is from a Company |
| UID | 3 | Reservation | UniqueID, to specify Reservation number |
|  |  |  |  |
|  | 10 | VerifyPrice | CancelType |
|  | 98 | CancelRoom | CancelType |

Table 17 - OTA\_CancelRQ supported codes

##### ClassDiagram

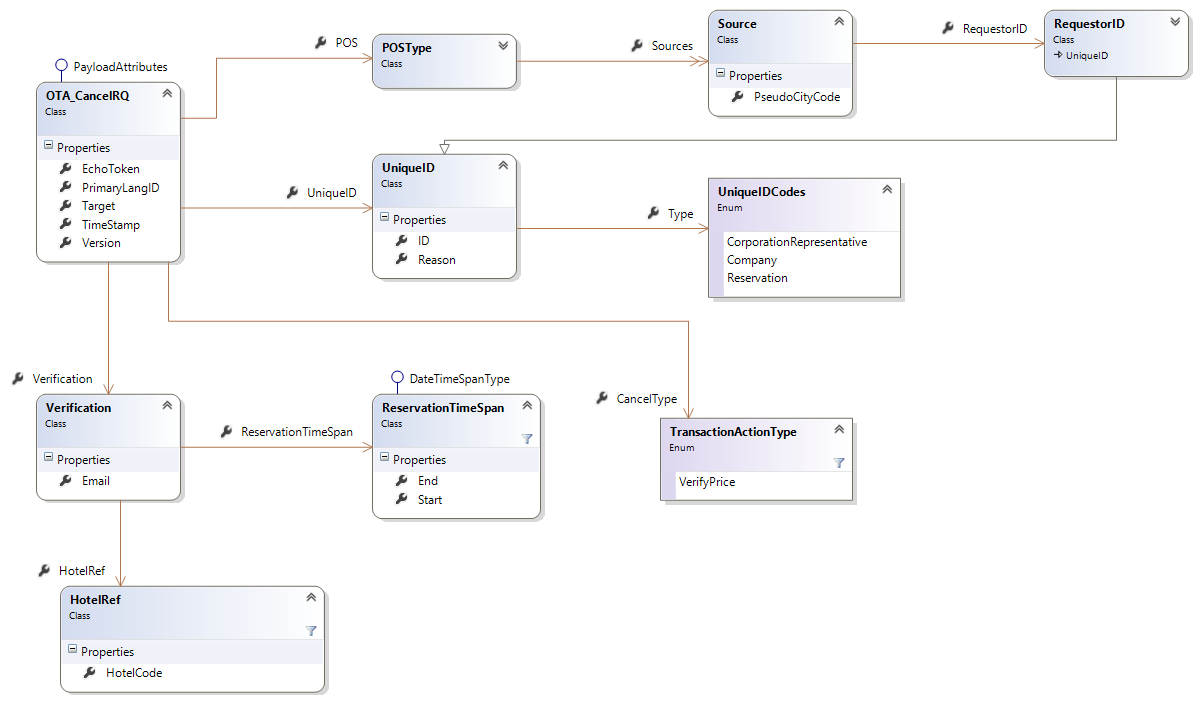


Figure 22 - OTA\_CancelRQ class diagram

#### OTA\_CancelRS

##### Schema

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Level | Occurrence | Element | Property | Mandatory (M)/ Optional (O) | Sample Usage |
| 0 | 1 | OTA\_CancelRS |  |  |  |
|  |  |  | Echotoken | M | Same as request |
|  |  |  | PrimaryLanID | O | Same as request |
|  |  |  | TimeStamp | M | 2014-06-23T15:04:16.0294414Z |
|  |  |  | Target | M | Same as request |
|  |  |  | Version | M | Same as request |
|  |  |  | StatusType | M | Cancelled |
|  |  |  |  |  |  |
| 1 | 1 | Segment |  | O |  |
| 1 | 0..1 | Segment |  | O |  |
|  |  |  | ItinsegNbr | M | Res000001254-1635/2 |
|  |  |  | Status | M | Allways Cancelled |
|  |  |  |  |  |  |
| 1 | 0..1 | Comment |  | O |  |
|  |  |  | Description | M | Cancellation penalties apply: Reservation is canceled with 520,00 BRL of total cost. |
|  |  |  | Language | O | En |
|  |  |  |  |  |  |
| 1 | 0..1 | CancelInfoRS |  | O |  |
| 2 | 1 | CancelRule |  | M |  |
|  |  |  | Amount | M | 520 |
|  |  |  | CurrencyCode | M | BRL |
|  |  |  |  |  |  |
| 1 | 0..1 | Sucess |  | O |  |
|  |  |  |  |  |  |
| 1 | 0..1 | WarningsType |  | O |  |
| 2 | 1..N | Warning |  | M |  |
|  |  |  | Code | M | 450 |
|  |  |  | Language | M | En |
|  |  |  | ShortText | M | Warning Text |
|  |  |  |  |  |  |
| 1 | 0..1 | ErrorsType |  | O |  |
| 2 | 1..N | Error |  | M |  |
|  |  |  | Code | M | 448 |
|  |  |  | Language | M | en |
|  |  |  | ShortText | M | System error: Internal system error. Please contact techincal support at integration.team@omnibees.com { echoToken = 57fde0db-32ed-40c5-bf88-e19a7601472a } |
|  |  |  |  |  |  |

Table 18 - OTA\_CancelRS schema

##### GeneralRules

1. TimeStamp – used to specific date at Omnibees server;
2. Echotoken – Same as request, to check if response match with request and to help on futher support;
3. Segment
   1. ItinSegNbr – Used to specify the number of reservation/room cancelled
   2. Status – Used to specify that reservation/room is cancelled.
4. Status – used to specify th current status of reservation after the response;
   1. Cancelled – Used to specify that reservation is canceled;
   2. Unchange – Used to specfy that no action was taken on reservation;
5. Comment – Used to specify information the total cost of canceling a reservation;
6. CancelRule – Used to specify the cost of cancelation;
   1. Amount – used to specify the amount of cancelation cost;
   2. CurrencyCode – Used to specify the currency code of cancelation cost
7. Success- used to inform that communication with Omnibees server was ok. When this tag is null the service will fill ErrorsType;
8. Warning – Used to specify a warning related to business rules like no cancelation cost to apply if request is only to check costs.This tag will be sent with success tag;
9. Error – Used to inform that an error occur at Omnibees server. The partner should contact Omnibees support with the echotoken passed.

##### Supported codes

|  |  |  |  |
| --- | --- | --- | --- |
| OTA Code List Name | Code Value | Code Name | Where Used |
|  | 0 | Test | Target, to specify wich enviroment request is destinated |
|  | 1 | Production | Target, to specify wich enviroment request is destinated |
|  |  |  |  |
| STS | 0 | Cancelled | StatusType |
| STS | 1 | UnChanged | StatusType |

Table 19 - OTA\_CancelRS supported codes

##### ClassDiagram

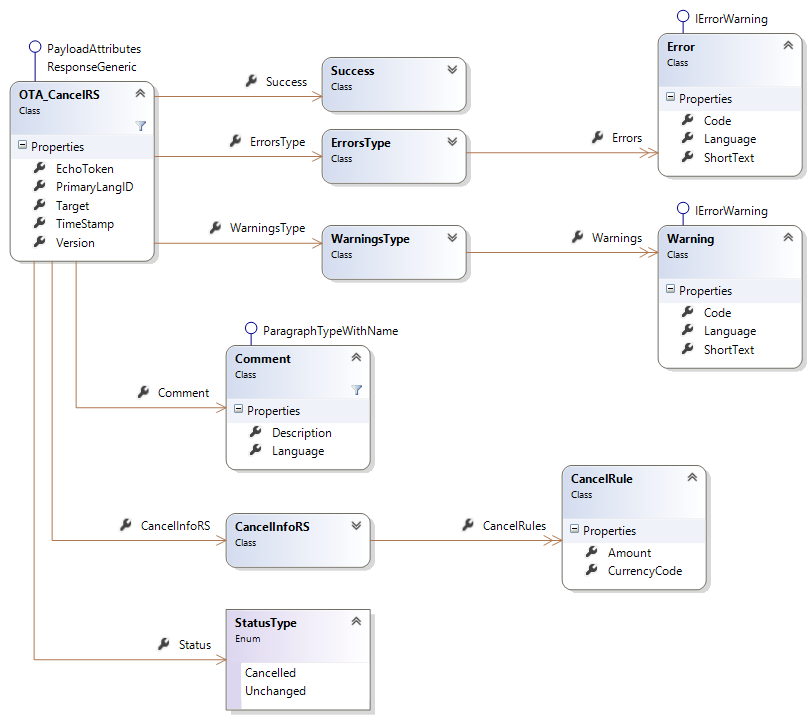


Figure 23 - OTA\_CancelRS class diagram

#### XML Examples

##### Cancel one reservation without Agency code and without Corporate Code

File on directory HotelResCancel: Ex1RQ.xml, Ex1RS.xml

##### Cancel one reservation with Agency

File on directory HotelResCancel: Ex2RQ.xml, Ex2RS.xml

##### Cancel one reservation with Corporate Code

File on directory HotelResCancel: Ex3RQ.xml, Ex3RS.xml

##### Cancel one reservation with Agency in Corporate Code

File on directory HotelResCancel: Ex4RQ.xml, Ex4RS.xml

##### Cancel one reservation with two room with same checkin and checkout dates

File on directory HotelResCancel: Ex5RQ.xml, Ex5RS.xml

##### Cancel one reservation with two room with different checkin and checkout dates

File on directory HotelResCancel: Ex6RQ.xml, Ex6RS.xml

##### Check Cancel Costs of one reservation but take no action on it

File on directory HotelResCancel: Ex7RQ.xml, Ex7RS.xml

##### Cancel Fail because min check is past dates

File on directory HotelResCancel: Ex8RQ.xml, Ex8RS.xml

##### Cancel Fail because verification field (one or more) fail

File on directory HotelResCancel: Ex9RQ.xml, Ex9RS.xml

##### Cancel one Room from a Reservation with two rooms

File on directory HotelResCancel: Ex10RQ.xml, Ex10RS.xml

### Get Bookings

#### Description

This messages provide information about single or multiple bookings. Filters to get bookings are allowed such as hotel id, last modify date from – to, Omnibees reservation number, partner reservation number.

Only one hotel can be passed in the request.

#### Get reservation by reservation id

<OTA\_ReadRQ xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<EchoToken>be571056-a407-4cc2-b454-537e2186c362</EchoToken>

<PrimaryLangID>en</PrimaryLangID>

<Target>Test</Target>

<TimeStamp>2015-06-22T14:40:58.5943902Z</TimeStamp>

<UniqueID>

<ID>RES002377-1635</ID>

<Type>Reservation</Type>

</UniqueID>

<Version>1.8</Version>

</OTA\_ReadRQ>

#### Get reservation by reservation number

<OTA\_ReadRQ xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<EchoToken>a77d3afa-824a-4d5f-9254-2c16eea9908b</EchoToken>

<PrimaryLangID>en</PrimaryLangID>

<Target>Test</Target>

<TimeStamp>2015-06-22T15:57:12.1031406Z</TimeStamp>

<UniqueID>

<ID>215685REN</ID>

<Type>AssociatedReservation</Type>

</UniqueID>

<Version>1.8</Version>

</OTA\_ReadRQ>

#### Get all bookings by last modified and hotel id

File on directory GetBookings: Ex1RQ.xml, Ex1RS.xml

#### Get booking by reservation number

File on directory GetBookings: Ex2RQ.xml, Ex2RS.xml

### HotelResModifyNotif

#### Description

This message allow partner to receive notifications about changes in one reservations. Each notification will inform partner that something is changed in reservation. The Omnibees reservation number and reservation status are the fields that goes in notifications. After receive the notification partner can use GetBookings message, with Omnibees reservation number, to get the reservation and see the current state of reservation.

The partner should implement a Rest webservice to receive the Omnibees OTA\_HotelResModifyNotifRQ object, and must return OTA\_HotelResModifyNotifRS.

There are some important objects that must be considered in response:

* + - 1. Echotoken – This object has to be returned always exactly the same object as the request object in OTA\_HotelResModifyNotifRQ. This will be used for further support;
      2. Success – This object should be sent in response in case of success in communication. This will be allowed when warnings is on response, but cannot be sent when Errors is filled on response;
      3. Warning – This object must be filled when some business rule is not correct, if for example the reservation do not exist in partner system, because some error happen when partner called SendHotelRes. This object must be sent with success object;
      4. Errors – This object must be return when some unexpected error happened in partner system. The partner should not return unthread exceptions. Omnibees advice for partner to have a generic handler that can return a friendly message when something unexpected happen at partner system. This object, when has some error, the success object must be null;

#### Login

#### OTA\_HotelResModifyNotifRQ

##### Schema

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Level | Occurrence | Element | Property | Mandatory (M)/ Optional (O) | Sample Usage |
| 0 | 1 | OTA\_HotelResModifyNotifRQ |  |  |  |
|  |  |  | Echotoken | M | c2ab0184-e9ac-4eb7-a9e1-7f1e8df2b66f |
|  |  |  | PrimaryLanID | M | Allways En |
|  |  |  | TimeStamp | M | 2014-06-23T15:04:16.0294414Z |
|  |  |  | Target | M | Test |
|  |  |  | Version | M | 2 |
|  |  |  |  |  |  |
| 1 | 1 | HotelResModifiesType |  | M |  |
| 2 | 1 | HotelResModifies |  | M |  |
| 3 | 1 | ResStatus |  | M |  |
|  |  |  | PMS\_ResStatusType | M | Reserved |
| 3 | 1 | UniqueID |  | M |  |
|  |  |  | ID | M | Res000001254-1635 |
|  |  |  | Type | M | Allways Reservation |

Table 20 - OTA\_HotelResModifyNotifRQ schema

##### General Rules

1. Echotoken – A Guid must be sent in every request. Must be unique per request;
2. PrimaryLangID – Indicates the language that responses should be sent;
3. Timestamp – Must send allways the date now in utc time;
4. Version – Version of the service;
5. Target – Must be Test for test proposes and Production for real scenario communications;
6. HotelReservationType – Used to specify the request object and to inform the reservation status at Omnibees and the reservation number;
   1. UniqueID – Used to specify the reservation number
      1. ID – used to specify reservation number at Omnibees;
      2. Type – The type for Reservation number is reservation;
   2. ResStatus – Used to specify status of reservation at Omnibees
      1. PMS\_ResStatus – Current status of reservation.

##### Supported Codes

|  |  |  |  |
| --- | --- | --- | --- |
| OTA Code List Name | Code Value | Code Name | Where Used |
|  | 0 | Test | Target, to specify which environment request is designated |
|  | 1 | Production | Target, to specify which environment request is designated |
|  |  |  |  |
| UID | 3 | Reservation | UniqueID, to specify Reservation number |
|  |  |  |  |
|  | 1 | Reserved | PMS\_ResStatusType |
|  | 2 | Cancelled | PMS\_ResStatusType |
|  | 3 | Waitlisted | PMS\_ResStatusType |
|  | 4 | Modified | PMS\_ResStatusType |
|  | 5 | Requested | PMS\_ResStatusType |
|  | 6 | RequestDenied | PMS\_ResStatusType |
|  | 7 | Cancelled | PMS\_ResStatusType |
|  | 8 | RequestAccepted | PMS\_ResStatusType |

Table 21 - OTA\_HotelResModifyNotifRQ supported codes

##### Class Diagram

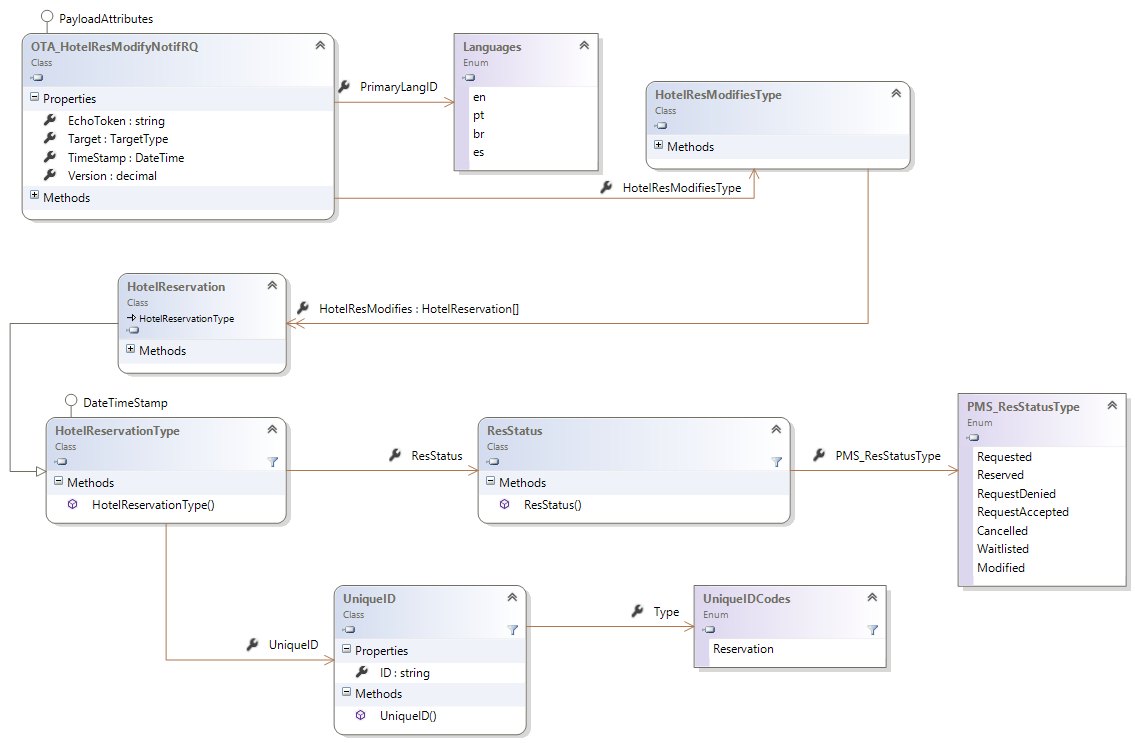


Figure 24 - OTA\_HotelResModifyNotifRQ class diagram

##### Example

{

"EchoToken": "3fd29adc-fc01-49f6-aea3-5f7bea0a9016",

"TimeStamp": "2016-02-11T14:29:13.4112672Z",

"Target": “Production”,

"Version": 2.0,

"PrimaryLangID": “en”,

"HotelResModifiesType": {

"HotelResModifies": [{

"ResStatus": {

"PMS\_ResStatusType": “RequestAccepted”

},

"UniqueID": {

"ID": "Res00001256-1881",

"Type": “Reservation”

}

}]

}

}

#### OTA\_HotelResModifyNotifRS

##### Schema

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Level | Occurrence | Element | Property | Mandatory (M)/ Optional (O) | Sample Usage |
| 0 | 1 | OTA\_HotelResModifyNotifRS |  |  |  |
|  |  |  | Echotoken | M | c2ab0184-e9ac-4eb7-a9e1-7f1e8df2b66f |
|  |  |  | PrimaryLanID | M | Allways En |
|  |  |  | TimeStamp | M | 2014-06-23T15:04:16.0294414Z |
|  |  |  | Target | M | Test |
|  |  |  | Version | M | 2 |
|  |  |  |  |  |  |
| 1 | 0..1 | Sucess |  | O |  |
|  |  |  |  |  |  |
| 1 | 0..1 | WarningsType |  | O |  |
| 2 | 1..N | Warning |  | M |  |
|  |  |  | ShortText | M | Warning Text |
|  |  |  |  |  |  |
| 1 | 0..1 | ErrorsType |  | O |  |
| 2 | 1..N | Error |  | M |  |
|  |  |  | ShortText | M | System error: Internal system error. Please contact techincal support at integration.team@omnibees.com { echoToken = 57fde0db-32ed-40c5-bf88-e19a7601472a } |

Table 22 - OTA\_HotelResModifyNotifRS schema

##### General Rules

1. Echotoken – A Guid must be sent in every response. Must be equal to request;
2. PrimaryLangID – Indicates the language that responses should be sent;
3. Timestamp – Must send always the date now in utc time;
4. Version – Version of the service;
5. Target – Must be Test for test proposes and Production for real scenario communications;
6. Success – This object should be sent in response in case of success in communication. This will be allowed when warnings is on response, but cannot be sent when Errors is filled on response;
7. Warning – This object must be filled when some business rule is not correct, if for example the reservation do not exist in partner system, because some error happen when partner called SendHotelRes. This object must be sent with success object;
8. Errors – This object must be return when some unexpected error happened in partner system. The partner should not return unthread exceptions. Omnibees advice for partner to have a generic handler that can return a friendly message when something unexpected happen at partner system. This object, when has some error, the success object must be null;

##### Class Diagram

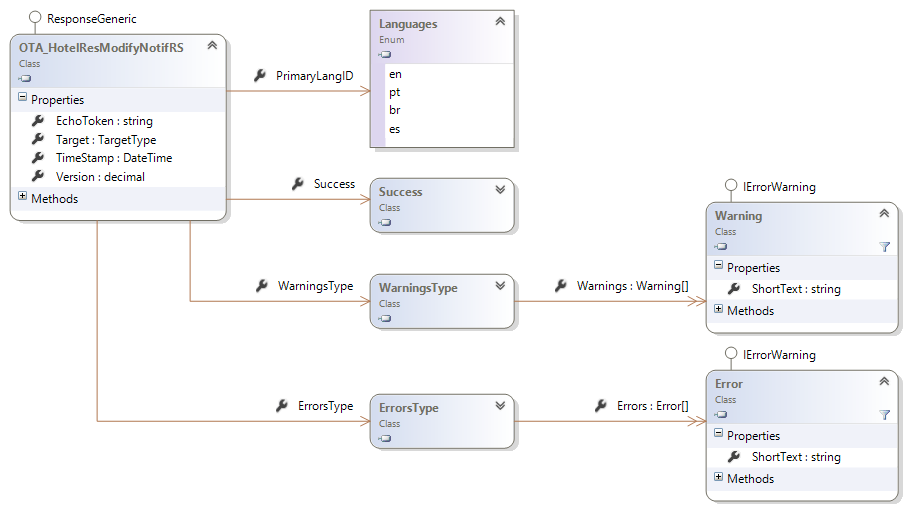


Figure 25 - OTA\_HotelResModifyNotifRS class diagram

##### Examples

###### Success

{

"EchoToken" : "3fd29adc-fc01-49f6-aea3-5f7bea0a9016",

"TimeStamp" : "2016-02-11T14:30:47.21276Z",

"Target" : “Production”,

"Version" : 2.0,

"PrimaryLangID" : “en”,

"Success" : {},

"WarningsType" : null,

"ErrorsType" : null

}

###### Warnings

{

"EchoToken" : "3fd29adc-fc01-49f6-aea3-5f7bea0a9016",

"TimeStamp" : "2016-02-11T14:30:47.21276Z",

"Target" : “Production”,

"Version" : 2.0,

"PrimaryLangID" : “en”,

"Success" : {},

"WarningsType" : {

"Warnings" : [{

"ShortText" : "Reservation does not exist in our system",

}

]

},

"ErrorsType" : null

}

###### Errors

{

"EchoToken" : "3fd29adc-fc01-49f6-aea3-5f7bea0a9016",

"TimeStamp" : "2016-02-11T14:30:47.21276Z",

"Target" : “Production”,

"Version" : 2.0,

"PrimaryLangID" : “en”,

"Success" : null,

"WarningsType" : null,

"ErrorsType" : {

"Errors" : [{

"ShortText" : "Some unexpected error occurred.",

}

]

}

}

## Hotel Descriptive Info

The hotel descriptive info allow partner to get information about hotels, rooms and multimedia descriptions.

For the partner the images are receive in a Url, but the partner should download the image and cache it for a better performance.

### OTA\_HotelDescriptiveInfoRQ

This message is what partner should fill to get information about hotel.

The partner should always ask what is need only. For example if the partner do not need the near attraction then should send <SendAttractions>false</SendAttractions>.

#### Options

##### HotelRef

The hotel ref is the most important tag in this request. The hotel Code should always be filled so the service can know which hotel to search. The service will answer just one hotel per request.

<HotelRef>

<ChainCode xsi:nil="true" />

<HotelCode>1814</HotelCode>

</HotelRef>

##### Awards

The awards is the information about the quality of the hotel.

<AffiliationInfo>

<SendAwards>true</SendAwards>

</AffiliationInfo>

##### AreaInfo

The area info indicates the geo location (SendRefPoints) and the near attraction’s (SendAttractions).

<AreaInfo>

<SendAttractions>true</SendAttractions>

<SendRefPoints>true</SendRefPoints>

</AreaInfo>

##### ContactInfo

The contact info allow to request all the available contacts for the hotel.

<ContactInfo>

<SendData>true</SendData>

</ContactInfo>

##### FacilityInfo

The Facility info allow to request information about all the rooms. This tag can be used with other tags like Multimedia objects. When Facility Info e set to true and multimedia objects is set to true, the response will send all rooms with images.

<FacilityInfo>

<SendGuestRooms>true</SendGuestRooms>

</FacilityInfo>

##### HotelInfo

The hotel info allow to get all hotel info like description’s, directions, check in hours, check ours, opening close, etc.

<HotelInfo>

<SendData>true</SendData>

</HotelInfo>

##### MultimediaObjects

The multimedia objects should be send when the partner need photos available in the hotel.

The photos are return in a url. The partner should download the photo and cache it for better performance. The image server will answer only for authorized ips.

<MultimediaObjects>

<SendData>true</SendData>

</MultimediaObjects>

##### Policies

The Polices tag allow to get all the policies for the hotel. For example cancelation policies, tax policies, deposit policies, child terms. This information should be used with the service that will get information about the rates because only rates has or not this policies.

<Policies>

<SendPolicies>true</SendPolicies>

</Policies>

#### Example

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

<OTA\_HotelDescriptiveInfoRQ xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<EchoToken>083549df-7514-4fad-bd6e-abda888ba0a9</EchoToken>

<HotelDescriptiveInfosType>

<HotelDescriptiveInfos>

<HotelDescriptiveInfo>

<AffiliationInfo>

<SendAwards>true</SendAwards>

</AffiliationInfo>

<AreaInfo>

<SendAttractions>true</SendAttractions>

<SendRefPoints>true</SendRefPoints>

</AreaInfo>

<ContactInfo>

<SendData>true</SendData>

</ContactInfo>

<FacilityInfo>

<SendGuestRooms>true</SendGuestRooms>

</FacilityInfo>

<HotelInfo>

<SendData>true</SendData>

</HotelInfo>

<HotelRef>

<ChainCode xsi:nil="true" />

<HotelCode>1814</HotelCode>

</HotelRef>

<MultimediaObjects>

<SendData>true</SendData>

</MultimediaObjects>

<Policies>

<SendPolicies>true</SendPolicies>

</Policies>

</HotelDescriptiveInfo>

</HotelDescriptiveInfos>

<LangRequested>en</LangRequested>

</HotelDescriptiveInfosType>

<PrimaryLangID>en</PrimaryLangID>

<Target>Test</Target>

<TimeStamp>2014-01-10T15:51:20.2976019Z</TimeStamp>

<Version>1</Version>

</OTA\_HotelDescriptiveInfoRQ>

### OTA\_HotelDescriptiveInfoRS

This message is the response for OTA\_HotelDescriptiveInfoRQ.

#### Class Diagram



#### Example

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

<OTA\_HotelDescriptiveInfoRS xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<ExtensionData />

<EchoToken>083549df-7514-4fad-bd6e-abda888ba0a9</EchoToken>

<HotelDescriptiveContentsType>

<HotelDescriptiveContents>

<HotelDescriptiveContent>

<ExtensionData />

<AffiliationInfo>

<ExtensionData />

<AwardsType>

<ExtensionData />

</AwardsType>

</AffiliationInfo>

<ContactInfosType>

<ExtensionData />

<ContactInfos>

<ContactInfo>

<ExtensionData />

<AddressesType>

<ExtensionData />

<Addresses>

<AddressType>

<ExtensionData />

<AddressLine>Av. Senador Dinarte de Mariz, 5525 - VIA COSTEIRA Parque das Dunas</AddressLine>

<CityCode>1109342</CityCode>

<CityName>Natal</CityName>

<CountryCode>40</CountryCode>

<CountryName>Brasil</CountryName>

<PostalCode>59090-001</PostalCode>

<StateProv>Rio Grande do Norte</StateProv>

<StateProvCode>2247291</StateProvCode>

</AddressType>

</Addresses>

</AddressesType>

<Emails>

<string>reservas.br@pestana.com</string>

</Emails>

<Location>HotelDirect</Location>

<Phones>

<ExtensionData />

<ContactNumbers>

<TelephoneInfo>

<ExtensionData />

<PhoneLocationType>PropertyDirect</PhoneLocationType>

<PhoneNumber>(84) 3220-8900</PhoneNumber>

<PhoneTechType>Voice</PhoneTechType>

<ShareMarketInd>Yes</ShareMarketInd>

<ShareSyncInd>Yes</ShareSyncInd>

</TelephoneInfo>

<TelephoneInfo>

<ExtensionData />

<PhoneLocationType>PropertyDirect</PhoneLocationType>

<PhoneTechType>Mobile</PhoneTechType>

<ShareMarketInd>Yes</ShareMarketInd>

<ShareSyncInd>Yes</ShareSyncInd>

</TelephoneInfo>

<TelephoneInfo>

<ExtensionData />

<PhoneLocationType>PropertyDirect</PhoneLocationType>

<PhoneTechType>Fax</PhoneTechType>

<ShareMarketInd>Yes</ShareMarketInd>

<ShareSyncInd>Yes</ShareSyncInd>

</TelephoneInfo>

</ContactNumbers>

</Phones>

<RelationshipTypeCode xsi:nil="true" />

<URLsType>

<ExtensionData />

<URLs>

<URL>

<ExtensionData />

<Address>www.pestana.com.br</Address>

<URLType>Website</URLType>

</URL>

</URLs>

</URLsType>

</ContactInfo>

<ContactInfo>

<ExtensionData />

<Emails>

<string xsi:nil="true" />

</Emails>

<Location>CustomerServiceOffice</Location>

<Names>

<ExtensionData />

<Names>

<PersonNameType>

<ExtensionData />

<NamePrefix>None</NamePrefix>

</PersonNameType>

</Names>

</Names>

<Phones>

<ExtensionData />

<ContactNumbers>

<TelephoneInfo>

<ExtensionData />

<PhoneLocationType>PropertyDirect</PhoneLocationType>

<PhoneTechType>Voice</PhoneTechType>

<ShareMarketInd>Yes</ShareMarketInd>

<ShareSyncInd>Yes</ShareSyncInd>

</TelephoneInfo>

</ContactNumbers>

</Phones>

<RelationshipTypeCode xsi:nil="true" />

</ContactInfo>

<ContactInfo>

<ExtensionData />

<Emails>

<string>reservas@pestananatal.com.br</string>

</Emails>

<Location>LocalReservationOffice</Location>

<Names>

<ExtensionData />

<Names>

<PersonNameType>

<ExtensionData />

<NamePrefix>None</NamePrefix>

</PersonNameType>

</Names>

</Names>

<Phones>

<ExtensionData />

<ContactNumbers>

<TelephoneInfo>

<ExtensionData />

<PhoneLocationType>PropertyReservationOffice</PhoneLocationType>

<PhoneNumber>(11) 4062-0609</PhoneNumber>

<PhoneTechType>Voice</PhoneTechType>

<ShareMarketInd>Yes</ShareMarketInd>

<ShareSyncInd>Yes</ShareSyncInd>

</TelephoneInfo>

<TelephoneInfo>

<ExtensionData />

<PhoneLocationType>PropertyReservationOffice</PhoneLocationType>

<PhoneNumber>(84) 3220-8920</PhoneNumber>

<PhoneTechType>Fax</PhoneTechType>

<ShareMarketInd>Yes</ShareMarketInd>

<ShareSyncInd>Yes</ShareSyncInd>

</TelephoneInfo>

</ContactNumbers>

</Phones>

<RelationshipTypeCode xsi:nil="true" />

</ContactInfo>

<ContactInfo>

<ExtensionData />

<AddressesType>

<ExtensionData />

<Addresses>

<AddressType>

<ExtensionData />

<AddressLine>Av. Senador Dinarte de Mariz, 5525 - VIA COSTEIRA Parque das Dunas</AddressLine>

<CityCode xsi:nil="true" />

<CityName>Natal</CityName>

<CountryCode>40</CountryCode>

<CountryName>Brasil</CountryName>

<PostalCode>59090-001</PostalCode>

<StateProvCode xsi:nil="true" />

</AddressType>

</Addresses>

</AddressesType>

<Company>

<ExtensionData />

<Name>BRASTURINVEST INVESTIMENTOS TURISTICOS S/A</Name>

<NumberHotels xsi:nil="true" />

<VAT>03.422.594/0004-60</VAT>

</Company>

<Emails>

<string>reservas1@pestananatal.com.br</string>

</Emails>

<Location>SalesOffice</Location>

<Names>

<ExtensionData />

<Names>

<PersonNameType>

<ExtensionData />

<GivenName>Pestana Natal Beach Resort</GivenName>

<NamePrefix>None</NamePrefix>

</PersonNameType>

</Names>

</Names>

<Phones>

<ExtensionData />

<ContactNumbers>

<TelephoneInfo>

<ExtensionData />

<PhoneLocationType>PropertyDirect</PhoneLocationType>

<PhoneNumber>(84) 3220-8900</PhoneNumber>

<PhoneTechType>Voice</PhoneTechType>

<ShareMarketInd>Yes</ShareMarketInd>

<ShareSyncInd>Yes</ShareSyncInd>

</TelephoneInfo>

</ContactNumbers>

</Phones>

<RelationshipTypeCode xsi:nil="true" />

</ContactInfo>

<ContactInfo>

<ExtensionData />

<AddressesType>

<ExtensionData />

<Addresses>

<AddressType>

<ExtensionData />

<AddressLine>RUA TUTÓIA, 77 </AddressLine>

<CityCode xsi:nil="true" />

<CityName>Sao Paulo</CityName>

<CountryCode>40</CountryCode>

<PostalCode>04007-000</PostalCode>

<StateProv>Sao Paulo</StateProv>

<StateProvCode xsi:nil="true" />

</AddressType>

</Addresses>

</AddressesType>

<Company>

<ExtensionData />

<NumberHotels>9</NumberHotels>

<VAT>02.002.263/0001-65</VAT>

</Company>

<Emails>

<string>reservas.br@pestana.com</string>

</Emails>

<Location>OwnershipCompany</Location>

<Names>

<ExtensionData />

<Names>

<PersonNameType>

<ExtensionData />

<NamePrefix>None</NamePrefix>

</PersonNameType>

</Names>

</Names>

<Phones>

<ExtensionData />

<ContactNumbers>

<TelephoneInfo>

<ExtensionData />

<PhoneLocationType>ManagingCompany</PhoneLocationType>

<PhoneNumber>(11) 2187 5000</PhoneNumber>

<PhoneTechType>Voice</PhoneTechType>

<ShareMarketInd>Yes</ShareMarketInd>

<ShareSyncInd>Yes</ShareSyncInd>

</TelephoneInfo>

<TelephoneInfo>

<ExtensionData />

<PhoneLocationType>ManagingCompany</PhoneLocationType>

<PhoneTechType>Mobile</PhoneTechType>

<ShareMarketInd>Yes</ShareMarketInd>

<ShareSyncInd>Yes</ShareSyncInd>

</TelephoneInfo>

<TelephoneInfo>

<ExtensionData />

<PhoneLocationType>ManagingCompany</PhoneLocationType>

<PhoneTechType>Voice</PhoneTechType>

<ShareMarketInd>Yes</ShareMarketInd>

<ShareSyncInd>Yes</ShareSyncInd>

</TelephoneInfo>

</ContactNumbers>

</Phones>

<RelationshipTypeCode xsi:nil="true" />

<URLsType>

<ExtensionData />

<URLs>

<URL>

<ExtensionData />

<Address>www.pestana.com</Address>

<URLType>Website</URLType>

</URL>

</URLs>

</URLsType>

</ContactInfo>

</ContactInfos>

</ContactInfosType>

<CurrencyCode>BRL</CurrencyCode>

<FacilityInfo>

<ExtensionData />

<GuestRoomsType>

<ExtensionData />

<GuestRooms>

<GuestRoom>

<ExtensionData />

<DescriptiveText>SUPERIOR DUNE VIEW</DescriptiveText>

<ID>5842</ID>

<MaxAdultOccupancy>3</MaxAdultOccupancy>

<MaxChildOccupancy>1</MaxChildOccupancy>

<MaxFreeChildOccupancy>1</MaxFreeChildOccupancy>

<MaxOccupancy>3</MaxOccupancy>

<MinAdultOccupancy>1</MinAdultOccupancy>

<MultimediaDescriptionsType>

<ExtensionData />

<MultimediaDescriptions>

<MultimediaDescription>

<ExtensionData />

<AdditionalDetailCode>RoomInformation</AdditionalDetailCode>

<InfoCode>Description</InfoCode>

<TextItemsType>

<ExtensionData />

<TextItems>

<TextItem>

<ExtensionData />

<Language xsi:nil="true" />

</TextItem>

</TextItems>

</TextItemsType>

</MultimediaDescription>

</MultimediaDescriptions>

</MultimediaDescriptionsType>

<Quantity>5</Quantity>

</GuestRoom>

<GuestRoom>

<ExtensionData />

<DescriptiveText>DELUXE SEA VIEW</DescriptiveText>

<ID>5843</ID>

<MaxAdultOccupancy>3</MaxAdultOccupancy>

<MaxChildOccupancy>1</MaxChildOccupancy>

<MaxFreeChildOccupancy>1</MaxFreeChildOccupancy>

<MaxOccupancy>3</MaxOccupancy>

<MinAdultOccupancy>1</MinAdultOccupancy>

<MultimediaDescriptionsType>

<ExtensionData />

<MultimediaDescriptions>

<MultimediaDescription>

<ExtensionData />

<AdditionalDetailCode>RoomInformation</AdditionalDetailCode>

<InfoCode>Description</InfoCode>

<TextItemsType>

<ExtensionData />

<TextItems>

<TextItem>

<ExtensionData />

<Language xsi:nil="true" />

</TextItem>

</TextItems>

</TextItemsType>

</MultimediaDescription>

</MultimediaDescriptions>

</MultimediaDescriptionsType>

<Quantity>40</Quantity>

</GuestRoom>

<GuestRoom>

<ExtensionData />

<DescriptiveText>SUPERIOR FAMILY SEA VIEW</DescriptiveText>

<ID>5844</ID>

<MaxAdultOccupancy>2</MaxAdultOccupancy>

<MaxChildOccupancy>2</MaxChildOccupancy>

<MaxFreeChildOccupancy>2</MaxFreeChildOccupancy>

<MaxOccupancy>4</MaxOccupancy>

<MinAdultOccupancy>1</MinAdultOccupancy>

<MultimediaDescriptionsType>

<ExtensionData />

<MultimediaDescriptions>

<MultimediaDescription>

<ExtensionData />

<AdditionalDetailCode>RoomInformation</AdditionalDetailCode>

<InfoCode>Description</InfoCode>

<TextItemsType>

<ExtensionData />

<TextItems>

<TextItem>

<ExtensionData />

<Language xsi:nil="true" />

</TextItem>

</TextItems>

</TextItemsType>

</MultimediaDescription>

</MultimediaDescriptions>

</MultimediaDescriptionsType>

<Quantity>2</Quantity>

</GuestRoom>

<GuestRoom>

<ExtensionData />

<DescriptiveText>DELUXE FAMILY SEA VIEW</DescriptiveText>

<ID>5845</ID>

<MaxAdultOccupancy>2</MaxAdultOccupancy>

<MaxChildOccupancy>2</MaxChildOccupancy>

<MaxFreeChildOccupancy>2</MaxFreeChildOccupancy>

<MaxOccupancy>4</MaxOccupancy>

<MinAdultOccupancy>1</MinAdultOccupancy>

<MultimediaDescriptionsType>

<ExtensionData />

<MultimediaDescriptions>

<MultimediaDescription>

<ExtensionData />

<AdditionalDetailCode>RoomInformation</AdditionalDetailCode>

<InfoCode>Description</InfoCode>

<TextItemsType>

<ExtensionData />

<TextItems>

<TextItem>

<ExtensionData />

<Language xsi:nil="true" />

</TextItem>

</TextItems>

</TextItemsType>

</MultimediaDescription>

</MultimediaDescriptions>

</MultimediaDescriptionsType>

<Quantity>2</Quantity>

</GuestRoom>

<GuestRoom>

<ExtensionData />

<DescriptiveText>Executive Suite Sea View</DescriptiveText>

<ID>5846</ID>

<MaxAdultOccupancy>3</MaxAdultOccupancy>

<MaxChildOccupancy>0</MaxChildOccupancy>

<MaxFreeChildOccupancy xsi:nil="true" />

<MaxOccupancy>3</MaxOccupancy>

<MinAdultOccupancy>1</MinAdultOccupancy>

<MultimediaDescriptionsType>

<ExtensionData />

<MultimediaDescriptions>

<MultimediaDescription>

<ExtensionData />

<AdditionalDetailCode>RoomInformation</AdditionalDetailCode>

<InfoCode>Description</InfoCode>

<TextItemsType>

<ExtensionData />

<TextItems>

<TextItem>

<ExtensionData />

<Language xsi:nil="true" />

</TextItem>

</TextItems>

</TextItemsType>

</MultimediaDescription>

</MultimediaDescriptions>

</MultimediaDescriptionsType>

<Quantity>2</Quantity>

</GuestRoom>

<GuestRoom>

<ExtensionData />

<DescriptiveText>Executive Suite Family Sea View</DescriptiveText>

<ID>5847</ID>

<MaxAdultOccupancy>3</MaxAdultOccupancy>

<MaxChildOccupancy>0</MaxChildOccupancy>

<MaxFreeChildOccupancy xsi:nil="true" />

<MaxOccupancy>3</MaxOccupancy>

<MinAdultOccupancy>1</MinAdultOccupancy>

<MultimediaDescriptionsType>

<ExtensionData />

<MultimediaDescriptions>

<MultimediaDescription>

<ExtensionData />

<AdditionalDetailCode>RoomInformation</AdditionalDetailCode>

<InfoCode>Description</InfoCode>

<TextItemsType>

<ExtensionData />

<TextItems>

<TextItem>

<ExtensionData />

<Language xsi:nil="true" />

</TextItem>

</TextItems>

</TextItemsType>

</MultimediaDescription>

</MultimediaDescriptions>

</MultimediaDescriptionsType>

<Quantity>5</Quantity>

</GuestRoom>

</GuestRooms>

</GuestRoomsType>

</FacilityInfo>

<HotelInfo>

<ExtensionData />

<CategoryCodes>

<ExtensionData />

<HotelCategory>

<ExtensionData />

<PropertyClassType>Hotel</PropertyClassType>

</HotelCategory>

</CategoryCodes>

<CheckInHours>

<ExtensionData />

<End>2013-05-17T12:00:00</End>

<Start>2013-05-17T14:00:00</Start>

</CheckInHours>

<Descriptions>

<ExtensionData />

<DescriptiveText>Na Via Costeira, sobre a praia, o Pestana é o ponto de partida ideal para se descobrir os principais pontos turísticos, além dos bons negócios. São 188 apartamentos bem equipados e confortaveis. Além disso, cotamos com o SPA Pestana Natal by L'occitane com tratamentos especiais. Venha conferir! </DescriptiveText>

<MultimediaDescriptionsType>

<ExtensionData />

<MultimediaDescriptions>

<MultimediaDescription>

<ExtensionData />

<AdditionalDetailCode>PropertyLocation</AdditionalDetailCode>

<InfoCode>DrivingDirections</InfoCode>

<TextItemsType>

<ExtensionData />

<TextItems>

<TextItem>

<ExtensionData />

<Language xsi:nil="true" />

<Name>Travel Direction</Name>

</TextItem>

</TextItems>

</TextItemsType>

</MultimediaDescription>

<MultimediaDescription>

<ExtensionData />

<AdditionalDetailCode>PropertyDescription</AdditionalDetailCode>

<InfoCode>Other</InfoCode>

<TextItemsType>

<ExtensionData />

<TextItems>

<TextItem>

<ExtensionData />

<Language xsi:nil="true" />

<Name>Additional Working Hours</Name>

</TextItem>

</TextItems>

</TextItemsType>

</MultimediaDescription>

<MultimediaDescription>

<ExtensionData />

<AdditionalDetailCode>PropertyDescription</AdditionalDetailCode>

<InfoCode>Marketing</InfoCode>

<TextItemsType>

<ExtensionData />

<TextItems>

<TextItem>

<ExtensionData />

<Language xsi:nil="true" />

<Name>TripadvisorURL</Name>

</TextItem>

</TextItems>

</TextItemsType>

</MultimediaDescription>

</MultimediaDescriptions>

</MultimediaDescriptionsType>

</Descriptions>

<HandicapRooms xsi:nil="true" />

<NFloors xsi:nil="true" />

<NonSmokingHotel>false</NonSmokingHotel>

<OwnershipManagementInfosType>

<ExtensionData />

<OwnershipManagementInfo>

<OwnershipManagementInfo>

<ExtensionData />

<Emails>

<string xsi:nil="true" />

</Emails>

<Location>HotelDirect</Location>

<Names>

<ExtensionData />

<Names>

<PersonNameType>

<ExtensionData />

<NamePrefix>None</NamePrefix>

</PersonNameType>

</Names>

</Names>

<Phones>

<ExtensionData />

<ContactNumbers>

<TelephoneInfo>

<ExtensionData />

<PhoneLocationType>Office</PhoneLocationType>

<PhoneNumber>(84) 3220-8900</PhoneNumber>

<PhoneTechType>Voice</PhoneTechType>

<ShareMarketInd>Yes</ShareMarketInd>

<ShareSyncInd>Yes</ShareSyncInd>

</TelephoneInfo>

</ContactNumbers>

</Phones>

<RelationshipTypeCode>GeneralManager</RelationshipTypeCode>

</OwnershipManagementInfo>

</OwnershipManagementInfo>

</OwnershipManagementInfosType>

<Position>

<ExtensionData />

</Position>

<SmokingRooms xsi:nil="true" />

<TotalRooms>60</TotalRooms>

<WhenBuilt />

</HotelInfo>

<HotelRef>

<ExtensionData />

<ChainCode xsi:nil="true" />

<HotelCode>1814</HotelCode>

<HotelName>Pestana Natal Beach Resort</HotelName>

</HotelRef>

<Policies>

<ExtensionData />

<ChildTerms>

<ChildTerm>

<ExtensionData />

<ChargeChildrenGroup>Free</ChargeChildrenGroup>

<CountForOccupancy>true</CountForOccupancy>

<Description>

De 0 a 5 anos - Free // De 6 a 12 anos = 50% da tarifa // Acima de 12 anos = 100% da tarifa.

De acordo com a legislação brasileira (art. 250, Estatuto da Criança e do Adolescente), para hospedagem de menor faz-se necessária a apresentação de declaração autorizando a hospedagem assinada por ambos os pais, com firma reconhecida, acompanhada de cópias autenticadas dos documentos de ambos os pais e da certidão de nascimento do menor, ou autorização da autoridade judiciária.

</Description>

<MaxAge>5</MaxAge>

<MinAge>0</MinAge>

<Name>Gratuidade de Criança</Name>

</ChildTerm>

<ChildTerm>

<ExtensionData />

<ChargeChildrenGroup>Child</ChargeChildrenGroup>

<CountForOccupancy>true</CountForOccupancy>

<Description>Cobrado.</Description>

<MaxAge>12</MaxAge>

<MinAge>6</MinAge>

<Name>Criança de 06 a 12 anos</Name>

</ChildTerm>

<ChildTerm>

<ExtensionData />

<ChargeChildrenGroup>Adult</ChargeChildrenGroup>

<CountForOccupancy>true</CountForOccupancy>

<Description>Cobrado como Adulto</Description>

<MaxAge>18</MaxAge>

<MinAge>13</MinAge>

<Name>Criança Pagante</Name>

</ChildTerm>

</ChildTerms>

</Policies>

<PrimaryLangID>br</PrimaryLangID>

<TimeZone>GMT-03:00, S. America Eastern (Brasilia)</TimeZone>

</HotelDescriptiveContent>

</HotelDescriptiveContents>

</HotelDescriptiveContentsType>

<PrimaryLangID>en</PrimaryLangID>

<Success>

<ExtensionData />

</Success>

<Target>Test</Target>

<TimeStamp>2014-01-10T15:51:20.3266035Z</TimeStamp>

<Version>1</Version>

</OTA\_HotelDescriptiveInfoRS>

## Rate Plan Request

This message allow partner to complete information in hotel descriptive info.

This request can be made by hotel. This message can be large because it covers all static information about rateplan for a request hotel. This message will not give price. For prices you should use hotelavail.

### OTA\_HotelRatePlanRQ

#### Options

##### Hotel Ref

The hotel ref is the most important tag in this request. The hotel Code should always be filled so the service can know which hotel to search. The service will answer just one hotel per request.

<HotelRef>

<HotelCode>1814</HotelCode>

</HotelRef>

<OTA\_HotelRatePlanRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<EchoToken>863164de-33c4-487f-9721-5e76d7facbcf</EchoToken>

<PrimaryLangID>en</PrimaryLangID>

<RatePlansType>

<RatePlans>

<RatePlanRQ>

<HotelRef>

<ChainCode xsi:nil="true" />

<HotelCode>1814</HotelCode>

</HotelRef>

</RatePlanRQ>

</RatePlans>

</RatePlansType>

<Target>Test</Target>

<TimeStamp>2014-04-24T11:12:01.9238846Z</TimeStamp>

<Version>1</Version>

</OTA\_HotelRatePlanRQ>

##### Search by One RatePlanType

Partner can search by a specific rate plan type.

<ota\_HotelRatePlanRQ xmlns:d4p1="http://schemas.datacontract.org/2004/07/Pull.BLL.Models.OTA" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

<d4p1:EchoToken>09316fd2-5ef9-4990-ac43-7af50c91bd97</d4p1:EchoToken>

<d4p1:PrimaryLangID>en</d4p1:PrimaryLangID>

<d4p1:RatePlansType>

<d4p1:RatePlans>

<d4p1:RatePlanRQ>

<d4p1:HotelRef>

<d4p1:HotelCode>1814</d4p1:HotelCode>

</d4p1:HotelRef>

<d4p1:RatePlanCandidatesType>

<d4p1:RatePlanCandidates>

<d4p1:RatePlanCandidate>

<d4p1:RatePlanType>10</d4p1:RatePlanType>

</d4p1:RatePlanCandidate>

</d4p1:RatePlanCandidates>

</d4p1:RatePlanCandidatesType>

</d4p1:RatePlanRQ>

</d4p1:RatePlans>

</d4p1:RatePlansType>

<d4p1:Target>Test</d4p1:Target>

<d4p1:TimeStamp>2017-09-21T10:41:00.8039707Z</d4p1:TimeStamp>

<d4p1:Version>2.5</d4p1:Version>

</ota\_HotelRatePlanRQ>

##### Search by Multiple RatePlanType

Partner can search by multiple specific rate plan type.

<ota\_HotelRatePlanRQ xmlns:d4p1="http://schemas.datacontract.org/2004/07/Pull.BLL.Models.OTA" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

<d4p1:EchoToken>09316fd2-5ef9-4990-ac43-7af50c91bd97</d4p1:EchoToken>

<d4p1:PrimaryLangID>en</d4p1:PrimaryLangID>

<d4p1:RatePlansType>

<d4p1:RatePlans>

<d4p1:RatePlanRQ>

<d4p1:HotelRef>

<d4p1:HotelCode>1814</d4p1:HotelCode>

</d4p1:HotelRef>

<d4p1:RatePlanCandidatesType>

<d4p1:RatePlanCandidates>

<d4p1:RatePlanCandidate>

<d4p1:RatePlanType>10</d4p1:RatePlanType>

</d4p1:RatePlanCandidate>

<d4p1:RatePlanCandidate>

<d4p1:RatePlanType>12</d4p1:RatePlanType>

</d4p1:RatePlanCandidate>

</d4p1:RatePlanCandidates>

</d4p1:RatePlanCandidatesType>

</d4p1:RatePlanRQ>

</d4p1:RatePlans>

</d4p1:RatePlansType>

<d4p1:Target>Test</d4p1:Target>

<d4p1:TimeStamp>2017-09-21T10:41:00.8039707Z</d4p1:TimeStamp>

<d4p1:Version>2.5</d4p1:Version>

</ota\_HotelRatePlanRQ>

### OTA\_HotelRatePlanRS

#### Example



#### Example for version 2.6 – One RatePlanType Response



#### Example for version 2.6 – Multiple RatePlanType Response



## GetAgencies

### Description

This message provides information about one agency or multiple agencies. This message allow to check agencies PCC codes.

On get agencies, service will inform all Agencies Offices for the agency.

An Agency office can have, optional attached one company.

The code for agency is PCC and the code for agencies offices is PCC inside AgenciesOffices object. The code for Company is in Code inside object Corporate.

When needed to send requests for availability for agencies the pcc sould be used.

When needed to send requests for availability for agencies offices the pcc of agency office sould be used.

### Deprecated objects

CorporatesType found in Agency object is deprecated.

AgencyOfficeCode is deprecated.

### Class Diagram

Please check class diagram attached.



### Get all agencies

File on directory GetAgencies: Ex1RQ.xml, Ex1RS.xml

### Get agency by PCC code

File on directory GetAgencies: Ex2RQ.xml, Ex2RS.xml

## GetCorporates

### Description

This message provides information about one Corporate or multiple corporates. This message allow to check corporates codes.

### Get all Corporates

File on directory GetCorporates: Ex1RQ.xml, Ex1RS.xml

### Get Corporate by corporate code

File on directory GetCorporates: Ex2RQ.xml, Ex2RS.xml

## GetCountryList

### Description

This message provides information about system country codes;

#### Request Example



#### Response Example



## GetStateList

### Description

This message provides information about states codes by country code;

#### Request



#### Response



## GetCityList

### Description

This message provides information about cities codes by country code and state code.

#### Request



#### Response



## GetZoneList

### Description

This message provide information about zone codes by Country code, StateSearchCode, ZoneSearchCode, CitiesDataUid.

#### Request



#### Response



# Attachments

## Examples

For most recent examples please use Pulldev environment website to generate examples.



## Suported Codes



# References

Microsoft. (2012, 08 02). *basicHttpBiding*. Retrieved 07 12, 2013, from MSDN: http://msdn.microsoft.com/en-us/library/ms731361.aspx

Opentravel. (2013, 02 15). *Online Schema - OpenTravel 2012B*. Retrieved 07 12, 2013, from Opentravel: http://www.opentravel.org/Specifications/SchemaIndex.aspx?FolderName=2012B