



Hotel Service XML Interface Specification **developer's guide**

v1.2.0
August 10

Hotelbeds Accommodation & Destination Services
Camí de Son Fangós, 100 - Torre A, 5ª Planta
07007 - Palma de Mallorca
Spain

I.	Document Control	3
II.	Operations.....	4
II.1	HotelValuedAvail.....	4
II.1.1	Request business rules	4
II.1.2	Request Observations	4
II.1.3	Request Acceptance Test	6
II.1.4	Response business rules	6
II.1.5	Response observations	6
III.	Particularities of common operations	8
III.1	ServiceAdd.....	8
III.1.1	Request Business Rules	8
III.1.2	Request Observations	8
III.1.3	Request Acceptance Test	8
III.1.4	Response Business Rules	9
III.1.5	Response Observations	9
III.2	ServiceRemove.....	9
III.2.1	Response Business Rules	10
III.3	PurchaseConfirm.....	10
III.3.1	Request Business Rules	10
III.3.2	Response Business Rules	10
III.3.3	Response Observations	11
III.3.4	Request Acceptance Test	11
III.4	PurchaseDetail.....	11
III.4.1	Response Business Rules	11
III.4.2	Response Observations	11
III.5	PurchaseList.....	11
III.5.1	Response Business Rules	12
III.6	PurchaseCancel	13
III.6.1	Response Business Rules	13
IV.	FAQ	14

I. Document Control

1.2.0	02/08/10	<ul style="list-style-type: none"> - New elements (HotelValuedAvailRQ.xsd): <ol style="list-style-type: none"> 1. HotelValuedAvailRQ/ShowDirectPayment 2. HotelValuedAvailRQ/ShowNetPrice - Added new section Opaque on Request Observations from HotelValuedAvail operation. - Added new section Opaque on Response Observations from HotelValuedAvail operation. - Added FAQ "Direct payment and net prices". Added commissionable model: - Updated section Response Business Rules from operation HotelValuedAvail. - Added section Response Observations from operation HotelValuedAvail. - Updated section Request Business Rules from operation ServiceAdd. - Updated section Response Business Rules from operation ServiceAdd. - Added section Request Observations from operation ServiceAdd. - Added section Response Observations from operation ServiceAdd. - Added section Request Business Rules from operation PurchaseConfirm. - Added section Response Observations from operation PurchaseConfirm. - Updated section Response Business Rules from operation PurchaseConfirm. - Updated section Response Business Rules from operation PurchaseDetail. - Added section Response Observation from operation PurchaseDetail. - Updated section Response Business Rules from operation PurchaseList. 	1
1.0.3	02/08/10	<ul style="list-style-type: none"> - Changed FAQs order. - HotelDetail operation deleted. - Deleted HotelDetailRQ.xsd, HotelDetailRQ.xml, HotelDetailRS.xsd and HotelDetailRS.xml from /xml. - Deleted HotelValuedAvailRQ.html and HotelValuedAvailRS.html from /xsddocs. - Modified FAQ No.8 "How can I modify the hotel board of a confirmed hotel service?" and FAQ No.9 "How can I modify the number of nights of a confirmed hotel service?" - Sections CancelProtectionAdd and CancelProtectionRemove deleted. 	1
1.0.2	22/08/08	<ul style="list-style-type: none"> - Modified section Request Observations from operation HotelValuedAvail - Modified FAQ No.2 	2
1.0.1	26/06/08	Fixed section Request Acceptance Test from operation HotelValuedAvail	2
1.0.0	02/06/08	New Hotel documentation file	2

II. Operations

II.1 HotelValuedAvail

Get a list of hotel services with available rooms according to requested parameters. The returned amounts are final prices.

HotelValuedAvail

Details:	xsddocs/HotelValuedAvailRQ.html	xsddocs/HotelValuedAvailRS.html
Schema:	xml/HotelValuedAvailRQ.xsd	xml/HotelValuedAvailRS.xsd
Example:	xml/HotelValuedAvailRQ.xml	xml/HotelValuedAvailRS.xml

II.1.1 REQUEST BUSINESS RULES

Language	Mandatory
CheckInDate/@date	At least today
CheckOutDate/@date	Later than CheckInDate/@date
CategoryList/HotelCategory/@type	SIMPLE value
BoardList/HotelBoard/@type	SIMPLE value
Destination/ZoneList/Zone	Can't be provided more than once.

II.1.2 REQUEST OBSERVATIONS

If `OccupancyList/HotelOccupancy/Occupancy/GuestList/Customer/@type` attribute value is CH, element `OccupancyList/HotelOccupancy/Occupancy/GuestList/Customer/Age` will be mandatory.

Use `HotelCodeList` element to search availability only for the selected hotel codes, or to exclude from the response the selected hotel codes.

Use Y value in the `HotelCodeList/@withinResults` attribute to indicate the system if you want to search only the provided hotel code list, or use N value to indicate the system if you want to exclude the hotel code list from the response.

Note that a maximum of 10 hotel codes will be taken in consideration.

To get the availability results ordered by price add an `ExtendedData` element to the `ExtraParamList` with the following values:

- `ExtraParamList/ExtendedData/@type`: EXT_ORDER value.
- `ExtraParamList/ExtendedData/Name`: ORDER_CONTRACT_PRICE value.
- `ExtraParamList/ExtendedData/Value`: ASC for ascending order, DESC for descending order.

Element `HotelValuedAvailRQ/Destination/ZoneList/Zone` can be SIMPLE or GROUP. There are some destinations divided in grouped zones (Zone type GROUP). If you only need availability results from one of these zones, you can get it sending the element `HotelValuedAvailRQ/Destination/ZoneList/Zone` in your request. Example:

PMI	-	-	189
PMI	MALLOPMI13	Majorca - East Coast	54
PMI	MALLOPMI21	Majorca - South Coast	67
PMI	MALLOPMI14	Majorca - North Coast	38
PMI	MALLOPMI15	Majorca - Centre and West Coast	13
PMI	PALMAPMI10	Palma City and Area	35

For destinations like **Paris** ('PAR'), **Barcelona** ('BCN'), **Berlin** ('BER'), **London** ('LON') and **Rome** ('ROE') we recommend you to send a grouped zone. The reason is that these destination codes include another zones and nearby towns. Example:

BCN	-	-	196
BCN	BARCEBCN9	Barcelona City Center	120
BCN	BARCEBCN10	Barcelona Greater	52
BCN	RESTOBCN16	REST OF PROVINCE	52

In addition, you must consider some limitations for HotelValuedAvail request:

A maximum of **31 nights** per hotel service can be requested.

A maximum of **5 rooms** per hotel service can be requested.

To get direct payment product and net prices please read FAQ "*Direct payment and net prices*".

II.1.2.1 Opaque

To get the Opaque product add an ExtendedData element to the ExtraParamList with the following values:

- ExtraParamList/ExtendedData/@type: EXT_ADDITIONAL_PARAM value.
- ExtraParamList/ExtendedData/Name: PARAM_SHOW_OPAQUE_CONTRACT value.
- ExtraParamList/ExtendedData/Value: Y to receive Opaque product, N to not receive.

II.1.3 REQUEST ACCEPTANCE TEST

The following elements and attributes are useless:

- Contract/IncomingOffice/Description
- Contract/IncomingOffice/ContactInfo
- Contract/IncomingOffice/FiscalNumber
- Contract/Sequence
- Contract/Classification
- Contract/CommentList
- CheckInDate/@time
- CheckOutDate/@time
- Destination/Name
- OccupancyList/HotelOccupancy/Occupancy/GuestList/Customer/CustomerId
- OccupancyList/HotelOccupancy/Occupancy/GuestList/Customer/Name
- OccupancyList/HotelOccupancy/Occupancy/GuestList/Customer/LastName
- OccupancyList/HotelOccupancy/Occupancy/GuestList/Customer/AdditionalInfo
- OccupancyList/HotelOccupancy/Occupancy/GuestList/Customer/Document
- OccupancyList/HotelOccupancy/Occupancy/GuestList/Customer/BirthDate
- OccupancyList/HotelOccupancy/Occupancy/GuestList/Customer/CountryCode
- CategoryList/HotelCategory content
- CategoryList/HotelCategory/@shortName
- BoardList/HotelBoard content
- BoardList/HotelBoard/@shortname

Furthermore:

- OccupancyList/HotelOccupancy/Occupancy/GuestList/Customer/Age is useless for adult customers.

II.1.4 RESPONSE BUSINESS RULES

One of ServiceHotel or ErrorList element must be present in the response unless the @totalItems attribute is 0.

If ServiceHotel element is present the following elements and attributes must be always present:

- PaginationData
- @timeToExpiration
- ServiceHotel/@availToken
- ServiceHotel/ContractList/Contract/Classification
- ServiceHotel/DateTo
- ServiceHotel/Currency
- ServiceHotel/TotalAmount
- ServiceHotel/HotelInfo/Destination/Name
- ServiceHotel/HotelInfo/Destination/ZoneList/Zone
- ServiceHotel/DirectPayment

II.1.5 RESPONSE OBSERVATIONS

If ServiceHotel element is present and ServiceHotel/DirectPayment = Y the following element could be provided:

- ServiceHotel/AvailableRoom/HotelRoom/Price/ExtraOverride

To get direct payment product and net prices please read FAQ "*Direct payment and net prices*".

II.1.5.1 Opaque

If you send the opaque extended data on availability request you will obtain the element `ServiceHotel/PackageRate` in the response.

If `PackageRate` value is `Y` means the contract is exclusive for packaging. If `PackageRate` value is `N` means the contract is not for packaging and can be sold as hotel only.

III. Particularities of common operations

III.1 ServiceAdd

In this section you will see the particularities of this operation for Hotel service. If you want to see the general specification for ServiceAdd, please see [HotelBeds XML Interface Specification-v2.4.4](#).

III.1.1 REQUEST BUSINESS RULES

Service/DateTo	Mandatory
Service/HotelInfo/Code	Mandatory if it is provided in the HotelValuedAvailRS/ServiceHotel/HotelInfo/Code element
Service/HotelInfo/Destination	Mandatory
Service/AvailableRoom/HotelOccupancy	Must be the same as HotelValuedAvailRS/ServiceHotel/AvailableRoom/HotelOccupancy element
Service/AvailableRoom/HotelRoom/RoomType/@characteristic	Mandatory
Service/AvailableRoom/HotelOccupancy/Occupancy/GuestList/Customer/Age	Mandatory if Service/AvailableRoom/HotelOccupancy/Occupancy/GuestList/Customer/@type attribute is CH.

The use of commissionable model implies a restriction: **one service per booking**. If more than one service is tried to be added in the shopping cart, the system will return an error.

III.1.2 REQUEST OBSERVATIONS

To get net prices please read FAQ "*Direct payment and net prices*".

III.1.3 REQUEST ACCEPTANCE TEST

The following elements and attributes are useless:

- ServiceAddRQ/Service/DateTo/@time
- ServiceAddRQ/Service/HotelInfo/Name
- ServiceAddRQ/Service/HotelInfo/DescriptionList
- ServiceAddRQ/Service/HotelInfo/ImageList
- ServiceAddRQ/Service/HotelInfo/Contact
- ServiceAddRQ/Service/HotelInfo/Category
- ServiceAddRQ/Service/HotelInfo/Destination/Name
- Only first ServiceAddRQ/Service/HotelInfo/Destination/ZoneList/Zone is taken in consideration
- ServiceAddRQ/Service/HotelInfo/ChildAge
- ServiceAddRQ/Service/HotelInfo/FacilityList
- ServiceAddRQ/Service/HotelInfo/Chain
- ServiceAddRQ/Service/HotelInfo/LicenseNumber
- ServiceAddRQ/Service/HotelInfo/Position
- ServiceAddRQ/Service/HotelInfo/IssuesList
- ServiceAddRQ/Service/AvailableRoom/HotelOccupancy/Occupancy/GuestList/Customer/CustomerId

- ServiceAddRQ/Service/AvailableRoom/HotelOccupancy/Ocuppancy/GuestList/ Customer/Name
- ServiceAddRQ/Service/AvailableRoom/HotelOccupancy/Ocuppancy/GuestList/ Customer/LastName
- ServiceAddRQ/Service/AvailableRoom/HotelOccupancy/Ocuppancy/GuestList/ Customer/AdditionalInfo
- ServiceAddRQ/Service/AvailableRoom/HotelOccupancy/Ocuppancy/GuestList/ Customer/Document
- ServiceAddRQ/Service/AvailableRoom/HotelOccupancy/Ocuppancy/GuestList/ Customer/BirthDate
- ServiceAddRQ/Service/AvailableRoom/HotelOccupancy/Ocuppancy/GuestList/ Customer/CountryCode
- ServiceAddRQ/Service/AvailableRoom/HotelOccupancy/Ocuppancy/GuestList/ Customer/Age is useless for adult customers
- ServiceAddRQ/Service/AvailableRoom/HotelRoom/@availCount
- ServiceAddRQ/Service/AvailableRoom/HotelRoom/@onRequest
- ServiceAddRQ/Service/AvailableRoom/HotelRoom/@modifiesSHRUI is useless if service status is NEW or CONFIRMED.
- ServiceAddRQ/Service/AvailableRoom/HotelRoom/Board/@shortName
- ServiceAddRQ/Service/AvailableRoom/HotelRoom/Price
- ServiceAddRQ/Service/AvailableRoom/HotelRoom/CancellationPolicy
- ServiceAddRQ/Service/AvailableRoom/HotelRoom/HotelRoomExtraInfo
- ServiceAddRQ/Service/AvailableRoom/HotelRoom/@SHRUI (Use only in case of modification)

III.1.4 RESPONSE BUSINESS RULES

If ServiceAddRS/Purchase element is present and ServiceAddRS/Purchase/ServiceList/Service/@type is ServiceHotel the following elements and attributes must be provided:

- ServiceAddRS/Purchase/ServiceList/Service/@SPUI
- ServiceAddRS/Purchase/ServiceList/Service/Status
- ServiceAddRS/Purchase/ServiceList/Service/Supplier
- ServiceAddRS/Purchase/ServiceList/Service/DateTo
- ServiceAddRS/Purchase/ServiceList/Service/HotelInfo/Name
- ServiceAddRS/Purchase/ServiceList/Service/HotelInfo/Destination/Name
- ServiceAddRS/Purchase/ServiceList/Service/DirectPayment
- ServiceAddRS/Purchase/ServiceList/Service/AdditionalCostList/Currency
- ServiceAddRS/Purchase/ServiceList/Service/AdditionalCostList/PvpEquivalent

In case of commissionable hotel service the following element must be provided:

- ServiceAddRS/Purchase/ServiceList/Service/AcceptedCardTypes

III.1.5 RESPONSE OBSERVATIONS

In case of commissionable hotel service the following element could be provided:

- ServiceAddRS/Purchase/ServiceList/Service/AvailableRoom/HotelRoom/Price/ExtraOverride

To get net prices please read FAQ "*Direct payment and net prices*".

III.2 ServiceRemove

In this section you will see the particularities of this operation for Hotel service. If you want to see the general specification for ServiceRemove, please see [HotelBeds XML Interface Specification-v2.4.4](#).

III.2.1 RESPONSE BUSINESS RULES

If `ServiceRemoveRS/Purchase/ServiceList/Service/@type` is `ServiceHotel` the following elements and attributes must be provided:

- `ServiceRemoveRS/Purchase/ServiceList/Service/@SPUI`
- `ServiceRemoveRS/Purchase/ServiceList/Service/Status`
- `ServiceRemoveRS/Purchase/ServiceList/Service/Supplier`
- `ServiceRemoveRS/Purchase/ServiceList/Service/DateTo`
- `ServiceRemoveRS/Purchase/ServiceList/Service/HotelInfo/Name`
- `ServiceRemoveRS/Purchase/ServiceList/Service/HotelInfo/Destination/Name`

III.3 PurchaseConfirm

In this section you will see the particularities of this operation for Hotel service. If you want to see the general specification for `PurchaseConfirm`, please see [HotelBeds XML Interface Specification-v2.4.4](#).

III.3.1 REQUEST BUSINESS RULES

To confirm a commissionable hotel service you should invoke confirmation operation through HTTPS (Hypertext Transfer Protocol **Secure**). If you don't do it you will obtain a security error.

In addition, there are some new elements that should be send to confirm a commissionable hotel service:

- `PurchaseConfirmRQ/ConfirmationData/ConfirmationServiceDataList/ServiceData/PaymentData/PaymentCard`
- `PurchaseConfirmRQ/ConfirmationData/ConfirmationServiceDataList/ServiceData/PaymentData/ContactData`

III.3.2 RESPONSE BUSINESS RULES

If `PurchaseConfirmRS/Purchase` element is present and `PurchaseConfirmRS/Purchase/ServiceList/Service/@type` is `ServiceHotel` the following elements and attributes must be provided:

- `PurchaseConfirmRS/Purchase/ServiceList/Service/@SPUI`
- `PurchaseConfirmRS/Purchase/ServiceList/Service/Status`
- `PurchaseConfirmRS/Purchase/ServiceList/Service/Supplier`
- `PurchaseConfirmRS/Purchase/ServiceList/Service/DateTo`
- `PurchaseConfirmRS/Purchase/ServiceList/Service/HotelInfo/Name`
- `PurchaseConfirmRS/Purchase/ServiceList/Service/HotelInfo/Destination/Name`
- `PurchaseConfirmRS/Purchase/ServiceList/Service/DirectPayment`
- `PurchaseConfirmRS/Purchase/ServiceList/Service/AdditionalCostList/Currency`
- `PurchaseConfirmRS/Purchase/ServiceList/Service/AdditionalCostList/PvpEquivalent`

In case of commissionable hotel service the following element must be provided:

- `PurchaseConfirmRS/Purchase/ServiceList/Service/AcceptedCardTypes`

In addition, in case of commissionable hotel service the element `supplier` will always be the same constant value:

- `PurchaseConfirmRS/Purchase/ServiceList/Service/Supplier/@name` = `"The Customer / El Cliente"` .
- `PurchaseConfirmRS/Purchase/ServiceList/Service/Supplier/@vatNumber` = `"00000000"` .

III.3.3 RESPONSE OBSERVATIONS

In case of commissionable hotel service the following element could be provided:

- PurchaseConfirmRS/Purchase/ServiceList/Service/AvailableRoom/HotelRoom/Price/ExtraOverride

III.3.4 REQUEST ACCEPTANCE TEST

III.4 PurchaseDetail

In this section you will see the particularities of this operation for Hotel service. If you want to see the general specification for PurchaseDetail, please see [HotelBeds XML Interface Specification-v2.4.4](#).

III.4.1 RESPONSE BUSINESS RULES

If PurchaseDetailRS/Purchase element is present and PurchaseDetailRS/Purchase/ServiceList/Service/@type is ServiceHotel the following elements and attributes must be provided:

- PurchaseDetailRS/Purchase/ServiceList/Service/@SPUI
- PurchaseDetailRS/Purchase/ServiceList/Service/Status
- PurchaseDetailRS/Purchase/ServiceList/Service/Supplier
- PurchaseDetailRS/Purchase/ServiceList/Service/DateTo
- PurchaseDetailRS/Purchase/ServiceList/Service/HotelInfo/Name
- PurchaseDetailRS/Purchase/ServiceList/Service/HotelInfo/Destination/Name
- PurchaseDetailRS/Purchase/ServiceList/Service/DirectPayment
- PurchaseDetailRS/Purchase/ServiceList/Service/AdditionalCostList/Currency
- PurchaseDetailRS/Purchase/ServiceList/Service/AdditionalCostList/PvpEquivalent

In case of commissionable hotel service the following element must be provided:

- PurchaseDetailRS/Purchase/ServiceList/Service/AcceptedCardTypes

In addition, in case of commissionable hotel service the element supplier will always be the same constant value:

- PurchaseDetailRS/Purchase/ServiceList/Service/Supplier/@name = "The Customer / El Cliente" .
- PurchaseDetailRS/Purchase/ServiceList/Service/Supplier/@vatNumber = "00000000" .

III.4.2 RESPONSE OBSERVATIONS

In case of commissionable hotel service the following element could be provided:

- PurchaseDetailRS/Purchase/ServiceList/Service/AvailableRoom/HotelRoom/Price/ExtraOverride

III.5 PurchaseList

In this section you will see the particularities of this operation for Hotel service. If you want to see the general specification for PurchaseList, please see [HotelBeds XML Interface Specification-v2.4.4](#).

III.5.1 RESPONSE BUSINESS RULES

If `PurchaseListRS/Purchase` element is present and `PurchaseListRS/Purchase/ServiceList/Service/@type` is `ServiceHotel` the following elements and attributes must be provided:

- `PurchaseListRS/Purchase/ServiceList/Service/Status`
- `PurchaseListRS/Purchase/ServiceList/Service/DateTo`
- `PurchaseListRS/Purchase/ServiceList/Service/HotelInfo/Name`
- `PurchaseListRS/Purchase/ServiceList/Service/HotelInfo/Destination/Name`
- `PurchaseListRS/Purchase/ServiceList/Service/DirectPayment`

III.6 PurchaseCancel

In this section you will see the particularities of this operation for Hotel service. If you want to see the general specification for PurchaseCancel, please see [HotelBeds XML Interface Specification-v2.4.4](#).

III.6.1 RESPONSE BUSINESS RULES

If `PurchaseCancelRS/Purchase` element is present and `PurchaseCancelRS/Purchase/ServiceList/Service/@type` is `ServiceHotel` the following elements and attributes must be provided:

- `PurchaseCancelRS/Purchase/ServiceList/Service/@SPUI`
- `PurchaseCancelRS/Purchase/ServiceList/Service/Status`
- `PurchaseCancelRS/Purchase/ServiceList/Service/Supplier`
- `PurchaseCancelRS/Purchase/ServiceList/Service/DateTo`
- `PurchaseCancelRS/Purchase/ServiceList/Service/HotelInfo/Name`
- `PurchaseCancelRS/Purchase/ServiceList/Service/HotelInfo/Destination/Name`

IV. FAQ

1. Can a hotel valued availability return hotel services without product code?

Yes, a hotel valued availability can return hotel services without product code. These hotel services have a fixed price depending on category and board. No hotel is associated with this service so far. Prior to check-in the client is informed of the hotel name.

2. How can I search multiple occupancy for a hotel service?

You can search multiple occupancy for a hotel service specifying as many `HotelOccupancy` tags in the `HotelValuedAvail` request as you want. At least one adult **MUST** be provided. `GuestList` is mandatory for non adult guests. Guest age is mandatory for children.

If you want to search more than one room with the same occupancy you need to use the element `HotelOccupancy/RoomCount` to indicate the number of rooms. Example:

Invalid	Valid
<pre> <OccupancyList> <HotelOccupancy> <RoomCount>1</RoomCount> <Occupancy> <AdultCount>2</AdultCount> <ChildCount>0</ChildCount> </Occupancy> </HotelOccupancy> <HotelOccupancy> <RoomCount>1</RoomCount> <Occupancy> <AdultCount>2</AdultCount> <ChildCount>0</ChildCount> </Occupancy> </HotelOccupancy> </OccupancyList> </pre>	<pre> <OccupancyList> <HotelOccupancy> <RoomCount>2</RoomCount> <Occupancy> <AdultCount>2</AdultCount> <ChildCount>0</ChildCount> </Occupancy> </HotelOccupancy> </OccupancyList> </pre>

3. How can I modify the room data of a confirmed hotel service?

To modify the room data of a confirmed hotel service follow the steps below:

- Call the `HotelValuedAvail` operation providing the desired parameters.
- Choose a `HotelValuedAvailRS/ServiceHotel/AvailableRoom` element containing the new desired room data.
- Call the `ServiceAdd` operation with the following considerations:
 - Set the `ServiceAddRQ/Service/@SPUI` attribute with the `SPUI` of the service you want to modify.
 - Set the selected `HotelValuedAvailRS/ServiceHotel/AvailableRoom` element to the `ServiceAddRQ/Service/AvailableRoom`.
 - Set the `ServiceAddRQ/Service/AvailableRoom/HotelRoom/@SHRUI` with the value from `HotelValuedAvailRS/ServiceHotel/AvailableRoom/HotelRoom/@SHRUI` attribute.
 - Set the `ServiceAddRQ/Service/AvailableRoom/HotelRoom/@modifiesSHRUI` attribute with the old `SHRUI`.
- Call `PurchaseConfirm` operation to commit the changes to the system.

Note that using this method you can modify in one go all the room data (room type, hotel board,...).

Note that you need the purchase detailed information to get the correct purchase token of the purchase. (see `PurchaseDetail` operation).

4. How can I modify the room type of a confirmed hotel service?

See How can I modify the room data of a confirmed hotel service?

5. How can I add a new room to a previously confirmed hotel service?

To add a new room to a previously confirmed hotel service follow the steps below:

- Call the `HotelValuedAvail` operation.
- Choose the new room.
- Call the `ServiceAdd` operation with the following considerations:
 - Set the `ServiceAddRQ/Service/@SPUI` attribute with the SPUI of the service you want to add the room.
 - Set the `ServiceAddRQ/Service/AvailableRoom` with the new selected room.
- Call `PurchaseConfirm` operation to commit the changes to the system.

Note that you need the purchase detailed information to get the correct purchase token of the purchase. (see `PurchaseDetail` operation).

6. How can I remove only a room from a confirmed hotel service?

To remove a room from a confirmed hotel service follow the steps below:

- Call the `PurchaseDetail` operation.
- Call the `ServiceAdd` operation with the following considerations:
 - Set the `ServiceAddRQ/Service/@SPUI` attribute with the SPUI of the purchase.
 - Set the `ServiceAddRQ/@purchaseToken` with the value of `PurchaseDetailRS/Purchase/@purchaseToken`.
 - Set the `ServiceAddRQ/Service/ContractList` with the value of `PurchaseDetailRS/Purchase/ServiceList/Service/ContractList` from the service you want to remove.
 - Set the `ServiceAddRQ/Service/HotelInfo` with the value of `PurchaseDetailRS/Purchase/ServiceList/Service/HotelInfo` from the service you want to remove.
 - Set the `ServiceAddRQ/Service/AvailableRoom/HotelRoom` element you want to remove.
 - Set the `ServiceAddRQ/Service/AvailableRoom/HotelRoom/@status` attribute to 'CANCELLED'.
 - Set the `ServiceAddRQ/Service/AvailableRoom/HotelRoom/@modifiesSHRUI` attribute to the SHRUI you want to remove.
- Call `PurchaseConfirm` operation to commit the changes to the system.
 - Set the `PurchaseConfirmRQ/ConfirmationData/@purchaseToken` to the `PurchaseToken` you want to confirm.

Note that removing the last room from a hotel service is equivalent to calling the `ServiceRemove` operation, it will remove the service from the purchase. If the service is the only service contained in the purchase it is equivalent to removing the last service of the purchase and you will get a purchase containing no services but with the cancellation fees (when applicable).

Note that you only need to provide `ServiceAddRQ/Service/AvailableRoom/HotelRoom/Board` and `ServiceAddRQ/Service/AvailableRoom/HotelRoom/RoomType` inside the `ServiceAddRQ/Service/AvailableRoom/HotelRoom` element.

7. How can I modify a confirmed hotel service?

To modify a confirmed hotel service follow the steps below:

- Call `PurchaseDetail` operation to get the correct `purchaseToken` of the purchase.
- Call the `HotelValuedAvail` operation providing the desired parameters.
- Choose a `HotelValuedAvailRS/ServiceHotel` element containing the new desired service hotel data.
- Call the `ServiceAdd` operation with the following considerations:
 - Set the `ServiceAddRQ/Service/@SPUI` attribute with the SPUI of the service you want to modify.
 - Set the selected `HotelValuedAvailRS/ServiceHotel` element to the `ServiceAddRQ/Service`.
- Call `PurchaseConfirm` operation to commit the changes to the system.

Note that you can only modify a confirmed hotel service if the old hotel code and the contract are the same as the new one.

Note that by using this method you can modify in one go all the service data (check-in date, check-out date, room type, hotel board,...) except hotel code.

8. How can I modify the hotel board of a confirmed hotel service?

To modify the board type of a confirmed hotel service follow the steps below:

- Call `PurchaseDetail` operation to get the correct `purchaseToken` of the purchase.
- Call the `ServiceAdd` operation with the following considerations:
 - Set the `ServiceAddRQ/Service/@SPUI` attribute with the SPUI of the service you want to modify.
 - Set the `ServiceAddRQ/Service/AvailableRoom/HotelRoom/Board` with the desired board type.
 - Set the `ServiceAddRQ/Service/AvailableRoom/HotelRoom/@modifiesSHRUI` attribute with the old SHRUI
- Call `PurchaseConfirm` operation to commit the changes to the system.

9. How can I modify the number of nights of a confirmed hotel service?

To modify the number of nights of a confirmed hotel service follow the steps below:

- Call `PurchaseDetail` operation to get the correct `purchaseToken` of the purchase.
- Call the `ServiceAdd` operation with the following considerations:
 - Set the `ServiceAddRQ/Service/@SPUI` attribute with the SPUI of the service you want to modify.
 - Set the `ServiceAddRQ/Service/DateFrom` element with the desired check-in date.
 - Set the `ServiceAddRQ/Service/DateTo` element with the desired check-out date.
 - Set the `ServiceAddRQ/Service/AvailableRoom/HotelRoom/@modifiesSHRUI` attribute with the old SHRUI.
- Call `PurchaseConfirm` operation to commit the changes to the system.

10. Direct payment and net prices

This new version of XML specification includes some new functionalities as commissionable hotel services and net prices. To offer you this type of services, your xml support have to discharge this functionalities on your Hotelbeds internal configuration.

By default, if you have activated these functionalities, commissionable hotel services will not appear on your response, and net prices will appear on availability and serviceAdd responses.

Two new elements has been created in order to change the default actions:

- ShowDirectPayment (Can be sent on HotelValuedAvail request)
- ShowNetPrices (Can be sent on HotelValuedAvail and ServiceAdd requests)

ShowDirectPayment

If you want to receive commisionable product you should send the element HotelValuedAvailRQ/ShowDirectPayment with value Y. If you don't send this element, or you send it with value N, you will not receive this type of product.

On availability response, if ServiceHotel element is present and you sent value Y on HotelValuedAvailRQ/ShowDirectPayment the following elements must be always present for hotel payment services (ServiceHotel/DirectPayment = Y):

- ServiceHotel/AvailableRoom/HotelRoom/Price/Commission
- ServiceHotel/AvailableRoom/HotelRoom/Price/NetPrice
- ServiceHotel/AvailableRoom/HotelRoom/Price/SellingPrice

Note: Commissionable product always contains Commission, NetPrice and SellingPrice elements.

If you continue the booking process for a direct payment service, you will receive these elements on all your booking responses (ServiceAddRS, PurchaseConfirmRS and PurchaseDetailRS).

- Purchase/ServiceList/Service/NetPrice
- Purchase/ServiceList/Service/Commission
- Purchase/ServiceList/Service/SellingPrice

ShowNetPrice

By default, you will receive net prices on availability and serviceAdd responses. If you want to not receive net prices you should send the elements HotelValuedAvailRQ/ShowNetPrice and ServiceAddRQ/ShowNetPrice with value N.

Exception: Net prices are always returned for commissionable product.

Binding Prices

WHAT ARE BINDING PRICES?

Binding prices are mandatory rates established by the hotelier that cannot be modified, neither upwards nor downwards.

HOW DO I ACTIVATE BINDING PRICES?

To benefit from this improvement, we need to activate it on a client-per-client basis.

We can do this in two separate phases:

- 1- On the test account, to allow you to make the necessary development.
- 2- On the live environment.

When it's activated you should do the following modifications:

Availability listRQ

There are no changes to be made on the request.

Availability listRS

When binding prices NOT activated, the request returns:

```
<Amount>102.02</Amount>
```

- With binding prices activated:

```
<amount>102.02</amount>
<net_price>102.02</net_price>
<commission>0</commission>
<selling_price mandatory="Y">120</selling_price>
```

OR

```
<amount>125.9</amount>
<net_price>125.9</net_price>
<commission>0</commission>
<selling_price mandatory="N">125.9</selling_price>
```

Tag SellingPrice mandatory="Y" Product subjected to binding price

Tag SellingPrice mandatory="N": Product not subjected to binding price

Tag description

```
<Amount>148.000</Amount>
Selling price through the XML
```

```
<NetPrice>118.310</NetPrice> (NEW TAG)
Net Price.
```

Remember that the selling price is meant for the final customer. If this product is sold to a third party other than the final customer, they must to be informed about the nature and amount of this value.

<Commission>17.000</Commission>
% Commission (VAT not included)..

<SellingPrice mandatory="Y">148.000</SellingPrice>(NEW TAG)
Mandatory attribute determines if the price is binding (Y) or not (N).

ValuationRQ

There are no changes to be made on the request.

ValuationRS

No changes on this response.

The service price will be the <Amount> price or the <SellingPrice>, but not the <net price>.

The Net price will be used for invoices on our side for binding and for normal prices.

ConfirmationRQ

There are no changes to be made on the request.

ConfirmationRS

No changes on this response.

The service price will be the <Amount> price or the <SellingPrice>, but not the <net price>.

The Net price will be used for invoices on our side for binding and for normal prices.

EXAMPLES OF BINDING PRICE RESPONSES

[...]

<room>

```
<code>DBL.FM</code>
<description>DOUBLE FAMILY ROOM</description>
<availability>20</availability>
<price>
  <occupancy>
    <adults>2</adults>
    <children>1</children>
    <rooms>1</rooms>
  </occupancy>
  <board_type>
    <code>BB</code>
    <description>BED AND BREAKFAST</description>
  </board_type>
  <amount>122.44</amount>
  <net_price>122.44</net_price>
  <commission>0</commission>
  <selling_price mandatory="Y">140.4</selling_price>
</price>
```

</room>

[...]

[...]

<room>

```
<code>TPL.ST</code>
<description>TRIPLE STANDARD</description>
<availability>6</availability>
<price>
  <occupancy>
    <adults>2</adults>
    <children>1</children>
    <rooms>1</rooms>
  </occupancy>
  <board_type>
    <code>R0</code>
    <description>ROOM ONLY</description>
  </board_type>
  <amount>125.9</amount>
  <net_price>125.9</net_price>
  <commission>0</commission>
  <selling_price mandatory="N">125.9</selling_price>
</price>
<price>
  <occupancy>
    <adults>2</adults>
    <children>1</children>
    <rooms>1</rooms>
  </occupancy>
  <board_type>
    <code>BB</code>
    <description>BED AND BREAKFAST </description>
  </board_type>
  <amount>156.42</amount>
  <net_price>156.42</net_price>
  <commission>0</commission>
  <selling_price mandatory="N">156.42</selling_price>
</price>
</room>
[...]
```

FREQUENTLY ASKED QUESTIONS

Please bear in mind that binding prices are only available for European rates at the moment. Shortly, they will be available on the American market.

What is a binding price?

It's a compulsory selling price (established by the hotelier) at a rate that cannot be modified, neither upwards or downwards. It's the final selling price of the product.

What are the benefits of binding prices?

There are certain properties that are only available with binding prices. So by activating it you will increase your product portfolio and receive access to top hotel chains at excellent prices.

Your profit will be the difference between the Sellingprice and the NetPrice/Amount.

If you sell the product to a third party/organisation, you will receive the difference between the Sellingprice and the NetPrice/Amount minus the amount you negotiate with the third party.

Why are the Net and Amount tags repeated if they have the same price?

For binding pricing products, the Net and Amount tags always carry the same price. For other type of payment methods (e.g. Direct Payment) prices differ, thus the necessity for both tags.

Why is the commission at "0"?

We offer net rates, so commission will always be 0. When commission-based rates become available we will inform you in case you might be interested.

Binding Prices

WHAT ARE BINDING PRICES?

Binding prices are mandatory rates established by the hotelier that cannot be modified, neither upwards nor downwards.

HOW DO I ACTIVATE BINDING PRICES?

To benefit from this improvement, we need to activate it on a client-per-client basis.

We can do this in two separate phases:

- 1- On the test account, to allow you to make the necessary development.
- 2- On the live environment.

Once activated, you need to implement the following modifications.

HotelValuedAvailRQ

Add the new version tag

version="1.2.2" or version="2010/02" (newest)

The XML feed should look as follows:

```
<HotelValuedAvailRQ echoToken="DummyEchoToken"
  sessionId="DummySessionId" version="1.2.2"
  xmlns="http://www.hotelbeds.com/schemas/2005/06/messages"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
  www.hotelbeds.com/schemas/2005/06/messages HotelValuedAvailRQ.xsd">
  <Language>ENG</Language>
  <Credentials>
    <User>TEST</User>
    <Password>TEST</Password>
  </Credentials>
  [...]
```

HotelValuedAvailRS

When binding prices NOT activated, the request returns:

```
<Price>
<Amount>151.970</Amount>
</Price>
```

When binding prices activated, the request returns:

```
<Price>
  <Amount>151.970</Amount>
  <SellingPrice mandatory="Y">187.250</SellingPrice>
  <NetPrice>151.970</NetPrice>
  <Commission>0.000</Commission>
</Price>
```

Remember that the selling price is meant for the final customer. If this product is sold to a third party other than the final customer, they must to be informed about the nature and amount of this value.

OR

```
<Price>  
  <Amount>151.970</Amount>  
  <SellingPrice mandatory="N">151.970</SellingPrice>  
  <NetPrice>151.970</NetPrice>  
  <Commission>0.000</Commission>  
</Price>
```

Tag SellingPrice mandatory="Y" Product subjected to binding price

Tag SellingPrice mandatory="N": Product not subjected to binding price

Note: <Amount>, <NetPrice> and <SellingPrice>, will be the same price when SellingPrice mandatory="N"

Tag description

```
<Amount>148.000</Amount>  
Selling price through the XML
```

```
<NetPrice>118.310</NetPrice> (NEW TAG)  
Net Price.
```

```
<Commission>17.000</Commission>  
% Commission (VAT not included).
```

```
<SellingPrice mandatory="Y">148.000</SellingPrice> (NEW TAG)  
Mandatory attribute determines if the price is binding (Y) or not (N).
```

ServiceAddRQ

Add the new version tag

version="1.2.2" or version="2010/02" (newest)

ServiceAdd RS

No changes on this response.

The service price will be the <Amount> price or the <SellingPrice>, but not the <net price>.

PurchaseConfirmRQ

Add the new version tag

version="1.2.2" or version="2010/02" (newest)

PurchaseConfirmRS

No changes on this response.

The service price will be the <Amount> price or the <SellingPrice>, but not the <net price>.

EXAMPLES OF BINDING PRICE RESPONSES

```
[...]
<AvailableRoom>
<HotelOccupancy>
  <RoomCount>1</RoomCount>
<Occupancy>
  <AdultCount>2</AdultCount>
  <ChildCount>0</ChildCount>
</Occupancy>
</HotelOccupancy>
<HotelRoom SHRUI="z/IEoSCzV7UUb50BxEa4mQ==" availCount="999" onRequest="N">
  <Board type="SIMPLE" code="SH-E10" shortname="RO">ROOM ONLY</Board>
  <RoomType type="SIMPLE" code="DBL-E10" characteristic="ST">DOUBLE STANDARD</
RoomType>
  <Price>
    <Amount>151.970</Amount>
    <SellingPrice mandatory="Y">187.250</SellingPrice>
    <NetPrice>151.970</NetPrice>
    <Commission>0.000</Commission>
  </Price>
</HotelRoom>
</AvailableRoom>
```

```
[...]
<AvailableRoom>
<HotelOccupancy>
  <RoomCount>1</RoomCount>
<Occupancy>
  <AdultCount>2</AdultCount>
  <ChildCount>0</ChildCount>
</Occupancy>
</HotelOccupancy>
<HotelRoom SHRUI="Blf+AAhnByk8sGs/7GD26g==" availCount="999" onRequest="N">
  <Board type="SIMPLE" code="HD-E10" shortname="BB">BED AND BREAKFAST</Board>
  <RoomType type="SIMPLE" code="DBL-E10" characteristic="ST">DOUBLE STANDARD</
RoomType>
  <Price>
    <Amount>189.320</Amount>
    <SellingPrice mandatory="N">189.320</SellingPrice>
    <NetPrice>189.320</NetPrice>
    <Commission>0.000</Commission>
  </Price>
</HotelRoom>
</AvailableRoom>
[...]
```


FREQUENTLY ASKED QUESTIONS

Please bear in mind that binding prices are only available for European rates at the moment. Shortly, they will be available on the American market.

What is a binding price?

It's a compulsory selling price (established by the hotelier) at a rate that cannot be modified, neither upwards or downwards. It's the final selling price of the product.

What are the benefits of binding prices?

There are certain properties that are only available with binding prices. So by activating it you will increase your product portfolio and receive access to top hotel chains at excellent prices.

Your profit will be the difference between the Sellingprice and the NetPrice/Amount.

If you sell the product to a third party/organisation, you will receive the difference between the Sellingprice and the NetPrice/Amount minus the amount you negotiate with the third party.

Why are the Net and Amount tags repeated if they have the same price?

For binding pricing products, the Net and Amount tags always carry the same price. For other type of payment methods (e.g. Direct Payment) prices differ, thus the necessity for both tags.

Why is the commission at "0"?

We offer net rates, so commission will always be 0. When commission-based rates become available we will inform you in case you might be interested.

LibeRATE - Direct Payment at the hotel

WHAT IS IT?

With Liberate, your customers can pay the booking directly to the hotel, earning you a commission on each booking made.

PRICING POLICIES

Wholesalers

Prices are final, not net.

3 currency types

Hotel's currency

Client's currency

Liberate's currency (currency has to be set up with the sales manager to confirm it)

In the valuation process the response includes totals, net prices and the price breakdown in the hotel's currency.

Commissions will appear in Liberate's currency.

Equivalent prices will appear in Liberate's currency.

Retailers

Final price doesn't change.

2 currency types

Hotel's currency

Client's currency

In the valuation process the response includes totals, net prices and the price breakdown in the hotel's currency.

Commissions will appear in Liberate's currency.

Equivalent prices will appear in Liberate's currency.

HOW CAN I ACTIVATE LIBERATE?

We will handle the activation process in two phases:

1- On the test account, allowing you to make the needed systems development.

2- On the live environment.

Once activated, you should add the following tags in order to start receiving direct payment products:

AvailabilityRQ

New tag: `<ShowDirectPayment>Y</ShowDirectPayment>`

The XML returns the following information:

```
<barceloDS_requests>
  <request type="valued availability list" id="1">
    <language_code>ESP</language_code>
```

With the Liberate model there can only be one service per booking. If you attempt to add more than one service to the shopping cart, the system will return an error.

```

<user>INTERESP</user>
<agency>
  <primary>xxx</primary>
  <secondary>xx</secondary>
  <detail>x</detail>
  <branch>x</branch>
</agency>
<contract/>
<check_in_date>20100906</check_in_date>
<check_out_date>20100908 </check_out_date>
<location>
  <destination_code>LON</destination_code>
  <zone_code/>
</location>
<establishment>
  <code/>
  <category/>
</establishment>
<board_type_code/>
<occupancy>
  <adults>2</adults>
  <children>0</children>
  <rooms>1</rooms>
</occupancy>
<show_direct_payment>Y</show_direct_payment>
</request>
</barceloDS_requests>

```

AvailabilityRS

The possible values of this new element are 'Y' and 'N'.

<DirectPayment>Y</DirectPayment> The hotel service is commissionable
 <DirectPayment>N</DirectPayment> The hotel service is not commissionable.

The mandatory attribute (Y) determines if the price is mandatory and binding, i.e. the sale price cannot be changed.

```

<amount>27.28</amount>
<net_price>24.55</net_price>
<commission>10</commission>
<selling_price mandatory="Y">27.28</selling_price>

```

ValuationRQ

There are no changes in the XML valuation request.

ValuationRS

New tag: <accepted_card_types>

This new tag will contain all card types accepted by the hotel.

In the following example a hotel accepts only MasterCard and Visa

```
<accepted_card_types>
<card>
<type>MC</type>
<description>Master Card</description>
</card>
<card>
<type>VI</type>
<description>Visa</description>
</card>
</accepted_card_types>
```

New tag: <currency>

The currency returned in valuation response may not always be the same as the returned in the availability response. <currency> shows the currency of the hotel.

```
<currency>
<code>GBP</code>
<description>GREAT BRITAIN POUND</description>
</currency>
<board_type>
<code>BB</code>
<description>ALOJAMIENTO Y DESAYUNO</description>
</board_type>
<net_price>82.02</net_price>
<commission>15.5</commission>
<selling_price mandatory="Y">100</selling_price>
```

<currency> only applies to the following elements
<agency_commission>, <commission_vat>, <pvp_equivalent>
New tag=<PvpEquivalent>

<PvpEquivalent> will return the amount of the reservation in the client's currency. This value will be calculated by the exchange rate of the hotel's and client's currency at the time the booking is confirmed.

New paths for Agency commission amount

```
(barceloDS_responses/response/file/commission)
net price
(barceloDS_responses/response/file/net_price)
selling price
(barceloDS_responses/response/file/selling_price).
```

Please see the example below,

```
<agency_currency>
<code>EUR</code>
<description>Euro</description>
</agency_currency>
```

```
<agency_commission>15.5</agency_commission>
<commission_vat>2.48</commission_vat>
<pvp_equivalent>86.23</pvp_equivalent>
```

CONFIRMATION

To confirm Liberate you should invoke confirmation operation through HTTPS (Hypertext Transfer Protocol Secure). If you don't do it you will obtain a security error.

Below, are the valid URLs for the LIVE and TEST environments

LIVE url: <https://api.interface-xml.com/barceloDS/interface/xml>

TEST url: <https://testapi.interface-xml.com/barceloDS/interface/xml>

PURCHASECONFIRMREQ

New tag: <PaymentData >

The elements below are all mandatory except <CardCVC>, which depends on the type of credit card:

```
<PaymentCard>
  <CardType>MC</CardType>
  <CardNumber>1111222233334444</CardNumber>
  <CardHolderName>Mr. Peter Smith</CardHolderName>
  <ExpiryDate>1210</ExpiryDate>
  <CardCVC>123</CardCVC>
</PaymentCard>
```

Optional tag: <ContactData>

```
<ContactData>
  <Email>palomar@yahoo.es</
Email>
  <PhoneNumber>34971225533</
PhoneNumber>
</ContactData>
```

All these PaymentCard elements are mandatory always except CardCvc which depends on the card type.

The credit card must expire after the check-in date, otherwise you will have an error before finalising the booking.

VOUCHER DETAILS

<supplier> (barceloDS_responses/response/file/supplier) will always have the following value: "The Customer / El Cliente".

<vat_number> (barceloDS_responses/response/file/supplier/vat_number) will always have the following dummy value "00000000".

```
<supplier>
  <name>The Customer / El
Cliente</name>
  <vat_number>00000000</
vat_number>
  <reference>258747</reference>
</supplier>
```



The final price is always given in the currency of the hotel. The final price can also be given in the client's currency. The amount will be calculated at the time of finalising the booking.

LibeRATE - Direct Payment at the hotel

WHAT IS IT?

With Liberate, your customers can pay the booking directly to the hotel, earning you a commission on each booking made.

PRICING POLICIES

Wholesalers

Prices are final, not net.

3 currency types

Hotel's currency

Client's currency

Liberate's currency (currency has to be set up with the sales manager to confirm it)

In the valuation process the response includes totals, net prices and the price breakdown in the hotel's currency.

Commissions will appear in Liberate's currency.

Equivalent prices will appear in Liberate's currency.

Retailers

Final price doesn't change.

2 currency types

Hotel's currency

Client's currency

In the valuation process the response includes totals, net prices and the price breakdown in the hotel's currency.

Commissions will appear in Liberate's currency.

Equivalent prices will appear in Liberate's currency.

HOW CAN I ACTIVATE LIBERATE?

We will handle the activation process in two phases:

1- On the test account, allowing you to make the needed systems development.

2- On the live environment.

Once activated, you should add the following tags in order to start receiving direct payment products:

HotelValuedAvailRQ

New tag=<2010/02>

New tag= <ShowDirectPayment>Y</ShowDirectPayment>

The XML returns the following information:

```
<HotelValuedAvailRQ echoToken="DummyEchoToken" version="2010/02"
  sessionId="DummySessionId">
```

With the Liberate model there can only be one service per booking. If you attempt to add more than one service to the shopping cart, the system will return an error.

```

xmlns="http://www.hotelbeds.com/schemas/2005/06/messages"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
www.hotelbeds.com/schemas/2005/06/messages HotelValuedAvailRQ.xsd">
  <Language>ENG</Language>
  <Credentials>
    <User>xxxxxxxx</User>
    <Password>xxxxxxxx</Password>
  </Credentials>
  <PaginationData pageNumber="1"/>
  <CheckInDate date="20101128"/>
  <CheckOutDate date="20101130"/>
  <Destination code="LON" type="SIMPLE"/>
  <OccupancyList>
    <HotelOccupancy>
      <RoomCount>1</RoomCount>
      <Occupancy>
        <AdultCount>2</AdultCount>
        <ChildCount>0</ChildCount>
      </Occupancy>
    </HotelOccupancy>
  </OccupancyList>
  <ShowDirectPayment>Y</ShowDirectPayment>
</HotelValuedAvailRQ>

```

HotelValuedAvailRS

The possible values of this new element are 'Y' and 'N'.

<DirectPayment>Y</DirectPayment>	The hotel service is commissionable
<DirectPayment>N</DirectPayment>	The hotel service is not commissionable.

The mandatory attribute (Y) determines if the price is mandatory and binding, i.e. the sale price cannot be changed.

```

<Price>
<Amount>582.070</Amount>
<SellingPrice mandatory="Y">582.070</SellingPrice>
<NetPrice>496.210</NetPrice>
<Commission>12.500</Commission>
</Price>

```

ServiceAddRQ

New tag=<2010/02>

ServiceAddRS

New tag=<AcceptedCardTypes>

In the following example a hotel accepts only MasterCard and Visa:

```

<AcceptedCardTypes>
  <Card type="MC">Master Card</Card>

```

```
<Card type="VI">Visa</Card>
</AcceptedCardTypes>
```

New tag=<currency>

The currency returned in valuation response may not always be the same as the returned in the availability response. <currency> shows the currency of the hotel.

New tag=<AdditionalCostList>

New tag=<PvpEquivalent>

<AdditionalCostList> will return <PvpEquivalent>, which determines the amount of the reservation in the client's currency. This value will be calculated by the exchange rate of the hotel's and client's currency at the time the booking.

The total amount to be paid by the client will be determined by <TotalAmount>.

Example for a reservation for a reservation in a hotel in London (GBP)

```
<Currency code="GBP">United Kingdom Pound</Currency>
<TotalAmount>481.760</TotalAmount>
<NetPrice>410.700</NetPrice>
<Commission>12.500</Commission>
<SellingPrice mandatory="Y">481.760</SellingPrice>
```

<AdditionalCostList> will return the client's currency

```
<AdditionalCostList>
<Currency code="EUR">Euro</Currency>
<PvpEquivalent>581.280</PvpEquivalent>
<AdditionalCost type="AG_COMMISSION">
<Price>
<Amount>72.660</Amount>
</Price>
</AdditionalCost>
<AdditionalCost type="COMMISSION_VAT">
<Price>
<Amount>13.080</Amount>
</Price>
</AdditionalCost>
</AdditionalCostList>
```

CONFIRMATION

All confirmations must invoke HTTPS (Hypertext Transfer Protocol Secure). HTTP will return a security error.

HTTPS URLs should look as follows:

LIVE account URL: <https://api.interface-xml.com/appservices/http/FrontendService>
 TEST account URL: <https://testapi.interface-xml.com/appservices/http/FrontendService>

Note: We don't store credit card details, we just run basic checks to ensure they are valid. Contact your Hotelbeds sales manager for more information on how hotels handle credit card transactions and details.

PurchaseConfirmRQ

New tag: version="2010/02"

New tag: <PaymentData>

The credit card types accepted by the hotel are in the Serviceadd requests under <PaymentData>

For example,

(PurchaseConfirmRQ/ConfirmationData/ConfirmationServiceDataList/ServiceData/PaymentData) returns the following:

```
<PaymentCard>
  <CardType>MC</CardType>
  <CardNumber>1111222233334444</CardNumber>
  <CardHolderName>Mr. Peter Smith</CardHolderName>
  <ExpiryDate>1210</ExpiryDate>
  <CardCVC>123</CardCVC>
</PaymentCard>
```

New optional tag: <ContactData>

```
<ContactData>
  <Email>palomar@yahoo.es</
Email>
  <PhoneNumber>34971225533</
PhoneNumber>
</ContactData>
```

All these PaymentCard elements are mandatory always except CardCvc which depends on the card type. The credit card must expire after the check-in date, otherwise you will have an error before finalising the booking.

Voucher details



The final price is always given in the currency of the hotel. The final price can also be given in the client's currency. The amount will be calculated at the time of finalising the booking.



Car Service XML Interface Specification **developer's guide**

v1.0.1
June 1010

Hotelbeds Accommodation & Destination Services
Camí de Son Fangós, 100 - Torre A, 5ª Planta
07007 - Palma de Mallorca
Spain

I.	Document Control	3
II.	Operations	4
II.1	CarCountryList.....	4
II.1.1	Request business rules	4
II.1.2	Request Observations	4
II.1.3	Request Acceptance Test	4
II.1.4	Response business rules	4
II.1.5	Response Observations	5
II.2	CarInfoSet	5
II.2.1	Request business rules	5
II.2.2	Request Observations	5
II.2.3	Request Acceptance Test	5
II.2.4	Response business rules	5
II.2.5	Response Observations	5
II.3	CarValuedAvail	6
II.3.1	Request business rules	6
II.3.2	Request Observations	6
II.3.3	Request Acceptance Test	6
II.3.4	Response business rules	7
II.3.5	Response Observations	7
III.	Particularities of common operations.....	8
III.1	ServiceAdd.....	8
III.1.1	Request Business Rules	8
III.1.2	Request Acceptance Test	8
III.1.3	Response Business Rules	8
III.2	ServiceRemove	9
III.2.1	Response Business Rules	9
III.3	PurchaseConfirm.....	9
III.3.1	Request Important Notices.....	9
III.3.2	Request Acceptance Test	9
III.3.3	Response Business Rules	9
III.4	PurchaseDetail.....	10
III.4.1	Response Business Rules	10
III.5	PurchaseList.....	10
III.5.1	Response Business Rules	10
III.6	PurchaseCancel	10
III.6.1	Response Business Rules	10

I. Document Control

1.0.1	02/06/10	CancelProtection section removed	3
1.0.0	14/04/08	New documentation file	3

II. Operations

II.1 CarCountryList

Get a list of available countries for car services containing their destinations and zones.

CarCountryList

Details:	xsddocs/CarCountryListRQ.html	xsddocs/CarCountryListRS.html
Schema:	xml/CarCountryListRQ.xsd	xml/CarCountryListRS.xsd
Example:	xml/CarCountryListRQ.xml	xml/CarCountryListRS.xml

II.1.1 REQUEST BUSINESS RULES

Language	Mandatory
Destination/@type	Must have the SIMPLE value

II.1.2 REQUEST OBSERVATIONS

CarCountryListRQ/Destination element is used to filter the response by destination.
CarCountryListRQ/CountryCode element is used to filter the response by country code.

If none of CarCountryListRQ/Destination or CarCountryListRQ/CountryCode elements are provided, all countries for car services are returned in the response.

If CarCountryListRQ/CountryCode is provided, only this country with all its destinations will be returned.

If CarCountryListRQ/Destination is provided, all zones for this destination are returned.

II.1.3 REQUEST ACCEPTANCE TEST

The following elements and attributes are useless:

- CarCountryListRQ/ExtraParamList
- CarCountryListRQ/Destination/Name
- CarCountryListRQ/Destination/ZoneList

II.1.4 RESPONSE BUSINESS RULES

If CarCountryListRS/Country element is present the following elements and attributes must be always present:

- CarCountryListRS/Country/Name

If CarCountryListRS/Country/DestinationList element is present the following elements and attributes must be always present:

- CarCountryListRS/Country/DestinationList/Destination/Name

If CarCountryListRS/Country/DestinationList/Destination/ZoneList element is present the following elements and attributes must be always present:

- CarCountryListRS/Country/DestinationList/Destination/ZoneList/Zone

II.1.5 RESPONSE OBSERVATIONS

If one or more `CarCountryListRS/Country` elements are present

`CarCountryListRS/@totalItems` attribute contains the total number of countries returned in the response.

If no `CarCountryListRS/Country` element is present and `CarCountryListRS/@totalItems` attribute is 0, no countries were found that match the search criteria.

II.2 CarInfoSet

Get the car offices, car groups and car special equipment available at a specific destination.

CarInfoSet

Details:	xsddocs/CarInfoSetRQ.html	xsddocs/CarInfoSetRS.html
Schema:	xml/CarInfoSetRQ.xsd	xml/CarInfoSetRS.xsd
Example:	xml/CarInfoSetRQ.xml	xml/CarInfoSetRS.xml

II.2.1 REQUEST BUSINESS RULES

Language	Mandatory
Destination/ZoneList/Zone	At least an element

II.2.2 REQUEST OBSERVATIONS

If more than one `CarInfoSetRQ/Destination/ZoneList/Zone` element is provided, only the first one will be taken as request parameter, the other elements in the list will be ignored.

II.2.3 REQUEST ACCEPTANCE TEST

The following elements and attributes are useless:

- `CarInfoSetRQ/ExtraParamList`
- `CarInfoSetRQ/Destination/Name`
- `CarInfoSetRQ/Destination/ZoneList/Content`

Furthermore:

- Only first `CarInfoSetRQ/Destination/ZoneList/Zone` is taken in consideration.

II.2.4 RESPONSE BUSINESS RULES

If `CarInfoSetRS/CarOfficeList` element is present the following elements and attributes must be always present:

- `CarInfoSetRS/CarOfficeList/Code`
- `CarInfoSetRS/CarOfficeList/Name`
- `CarInfoSetRS/CarOfficeList/Destination/Name`
- `CarInfoSetRS/CarOfficeList/Destination/ZoneList/Zone`

II.2.5 RESPONSE OBSERVATIONS

If no `CarInfoSetRS/ErrorList` or `CarInfoSetRS/CarOfficeList` or

`CarInfoSetRS/CarGroupList` elements are present no car info is available for the provided destination.

II.3 CarValuedAvail

Get a list of available car services according to the requested parameters. The returned prices are final prices.

CarValuedAvail

Details:	xsddocs/CarValuedAvailRQ.html	xsddocs/CarValuedAvailRS.html
Schema:	xml/CarValuedAvailRQ.xsd	xml/CarValuedAvailRS.xsd
Example:	xml/CarValuedAvailRQ.xml	xml/CarValuedAvailRS.xml

II.3.1 REQUEST BUSINESS RULES

Language	Mandatory
RentalDate/@date	At least today
RentalDate/@time	Mandatory
RentalDate/@date	Earlier than ReturnDate/@date
ReturnDate/@time	Mandatory
RentalDestination/ZoneList/Zone	Mandatory
ReturnDestination/ZoneList/Zone	Mandatory
CarCategory	Mandatory

II.3.2 REQUEST OBSERVATIONS

If more than one CarValuedAvailRQ/RentalDestination/ZoneList/Zone or CarValuedAvailRQ/ReturnDestination/ZoneList/Zone element is provided, only the first one will be taken as request parameter, the other elements in the list will be ignored.

To get a list of available rental and return destinations use CarCountryList operation (see CarCountryList).

To get a list of available car categories or special equipment use CarInfoSet operation (see CarInfoSet).

II.3.3 REQUEST ACCEPTANCE TEST

The following elements and attributes are useless:

- CarValuedAvailRQ/ExtraParamList
- CarValuedAvailRQ/Contract
- CarValuedAvailRQ/RentalDestination/Name
- CarValuedAvailRQ/RentalDestination/ZoneList/Zone content
- CarValuedAvailRQ/ReturnDestination/ZoneList/Zone content
- CarValuedAvailRQ/SpecialEquipmentList/ServiceCarSpecialEquip/Name
- CarValuedAvailRQ/SpecialEquipmentList/ServiceCarSpecialEquip/Description

Furthermore:

- Only first CarValuedAvailRQ/RentalDestination/ZoneList/Zone is taken in consideration.
- Only first CarValuedAvailRQ/ReturnDestination/ZoneList/Zone is taken in consideration.

II.3.4 RESPONSE BUSINESS RULES

One of CarValuedAvailRS/ServiceCar or CarValuedAvailRS/ErrorMessageList element must be present in the response unless the @totalItems attribute is 0.

If CarValuedAvailRS/ServiceCar element is present the following elements and attributes must be always present:

- CarValuedAvailRS/ServiceCar/@availToken
- CarValuedAvailRS/ServiceCar/ContractList
- CarValuedAvailRS/ServiceCar/DateFrom/@time
- CarValuedAvailRS/ServiceCar/DateTo/@time
- CarValuedAvailRS/ServiceCar/Currency
- CarValuedAvailRS/ServiceCar/TotalAmount
- CarValuedAvailRS/ServiceCar/CarGroup
- CarValuedAvailRS/ServiceCar/RentalOffice/Code
- CarValuedAvailRS/ServiceCar/ReturnOffice/Code
- CarValuedAvailRS/ServiceCar/CarInfo/Code
- CarValuedAvailRS/ServiceCar/CarInfo/Name
- CarValuedAvailRS/ServiceCar/CarInfo/Vendor

II.3.5 RESPONSE OBSERVATIONS

If one or more CarValuedAvailRS/ServiceCar elements are present:

- /@totalItems attribute contains the total number of car services returned in the response.

If no CarValuedAvailRS/ServiceCar element is present and /@totalItems attribute is 0, no hotel services were found matching the search criteria.

Additional service car rate description information may be provided in CarValuedAvailRS/ServiceCar/ServiceExtraInfoList/ExtendedData element:

- Type value: EXT_ADDITIONAL_INFO
- Name value: INFO_CAR_RATE_DESCRIPTION

III. Particularities of common operations

III.1 ServiceAdd

In this section you will see the particularities of this operation for Car service. If you want to see the general specification for ServiceAdd, please see [HotelBeds XML Interface Specification-v2.4.4](#).

III.1.1 REQUEST BUSINESS RULES

When you try to add a Car Service:

Service/DateFrom/@time	Mandatory
Service/DateTo	Mandatory
Service/DateTo/@time	Mandatory
Service/CarInfo/Code	Mandatory

III.1.2 REQUEST ACCEPTANCE TEST

The following elements and attributes are useless:

- ServiceAddRQ/Service/CarInfo/Name
- ServiceAddRQ/Service/CarInfo/DescriptionList
- ServiceAddRQ/Service/CarInfo/ImageList
- ServiceAddRQ/Service/CarInfo/Contact
- ServiceAddRQ/Service/CarInfo/TransmissionType
- ServiceAddRQ/Service/CarInfo/AirConditioning
- ServiceAddRQ/Service/CarInfo/Vendor
- ServiceAddRQ/Service/CarInfo/DoorCount
- ServiceAddRQ/Service/CarInfo/Abs
- ServiceAddRQ/Service/CarInfo/Airbag
- ServiceAddRQ/Service/CarInfo/PowerSteering
- ServiceAddRQ/Service/CarInfo/SeatCount
- ServiceAddRQ/Service/CarInfo/Driver/AdditionalInfo
- ServiceAddRQ/Service/CarInfo/Driver/Document/DocIssueAuthority
- ServiceAddRQ/Service/CarInfo/Driver/Document/DocIssueLocation
- ServiceAddRQ/Service/CarInfo/Driver/Document/DocType
- ServiceAddRQ/Service/CarInfo/Driver/Document/EffectiveDate
- ServiceAddRQ/Service/CarInfo/Driver/Document/ExpireDate
- ServiceAddRQ/Service/CarInfo/Driver/Document/DocHolderName
- ServiceAddRQ/Service/CarInfo/Driver/Document/DocLimitations
- ServiceAddRQ/Service/RentalOffice
- ServiceAddRQ/Service/ReturnOffice
- ServiceAddRQ/Service/SpecialEquipmentList

III.1.3 RESPONSE BUSINESS RULES

If ServiceAddRS/Purchase element is present and ServiceAddRS/Purchase/ServiceList/Service/@type is ServiceCar the following elements and attributes must be provided:

- ServiceAddRS/Purchase/ServiceList/Service/@SPUI
- ServiceAddRS/Purchase/ServiceList/Service/Status
- ServiceAddRS/Purchase/ServiceList/Service/Supplier
- ServiceAddRS/Purchase/ServiceList/Service/DateFrom/@time
- ServiceAddRS/Purchase/ServiceList/Service/DateTo/@time

III.2 ServiceRemove

In this section you will see the particularities of this operation for Car service. If you want to see the general specification for ServiceRemove, please see [HotelBeds XML Interface Specification-v2.4.4](#).

III.2.1 RESPONSE BUSINESS RULES

If ServiceRemoveRS/Purchase element is present and ServiceRemoveRS/Purchase/ServiceList/Service/@type is ServiceCar the following elements and attributes must be provided:

- ServiceRemoveRS/Purchase/ServiceList/Service/@SPUI
- ServiceRemoveRS/Purchase/ServiceList/Service/Status
- ServiceRemoveRS/Purchase/ServiceList/Service/Supplier
- ServiceRemoveRS/Purchase/ServiceList/Service/DateFrom/@time
- ServiceRemoveRS/Purchase/ServiceList/Service/DateTo/@time

III.3 PurchaseConfirm

In this section you will see the particularities of this operation for Car service. If you want to see the general specification for PurchaseCancel, please see [HotelBeds XML Interface Specification-v2.4.4](#).

III.3.1 REQUEST IMPORTANT NOTICES

Driver is mandatory for car services if it was not specified when the car service was added.

III.3.2 REQUEST ACCEPTANCE TEST

The following elements and attributes are useless:

- PurchaseConfirmRQ/ConfirmationData/Confirmation/ServiceDataList/ServiceData/CommentList

III.3.3 RESPONSE BUSINESS RULES

If PurchaseConfirmRS/Purchase element is present and PurchaseConfirmRS/Purchase/ServiceList/Service/@type is ServiceCar the following elements and attributes must be provided:

- PurchaseConfirmRS/Purchase/ServiceList/Service/@SPUI
- PurchaseConfirmRS/Purchase/ServiceList/Service/Status
- PurchaseConfirmRS/Purchase/ServiceList/Service/Supplier
- PurchaseConfirmRS/Purchase/ServiceList/Service/DateFrom/@time
- PurchaseConfirmRS/Purchase/ServiceList/Service/DateTo/@time
- PurchaseConfirmRS/Purchase/ServiceList/Service/Driver/LastName

III.4 PurchaseDetail

In this section you will see the particularities of this operation for Car service. If you want to see the general specification for PurchaseDetail, please see [HotelBeds XML Interface Specification-v2.4.4](#).

III.4.1 RESPONSE BUSINESS RULES

If PurchaseDetailRS/Purchase element is present and PurchaseDetailRS/Purchase/ServiceList/Service/@type is ServiceCar the following elements and attributes must be provided:

- PurchaseDetailRS/Purchase/ServiceList/Service/@SPUI
- PurchaseDetailRS/Purchase/ServiceList/Service/Status
- PurchaseDetailRS/Purchase/ServiceList/Service/Supplier
- PurchaseDetailRS/Purchase/ServiceList/Service/DateFrom/@time
- PurchaseDetailRS/Purchase/ServiceList/Service/DateTo/@time

III.5 PurchaseList

In this section you will see the particularities of this operation for Car service. If you want to see the general specification for PurchaseList, please see [HotelBeds XML Interface Specification-v2.4.4](#).

III.5.1 RESPONSE BUSINESS RULES

If PurchaseListRS/Purchase element is present and PurchaseListRS/Purchase/ServiceList/Service/@type is ServiceCar the following elements and attributes must be provided:

- PurchaseListRS/Purchase/ServiceList/Service/Status
- PurchaseListRS/Purchase/ServiceList/Service/DateFrom/@time
- PurchaseListRS/Purchase/ServiceList/Service/DateTo/@time
- PurchaseListRS/Purchase/ServiceList/Service/Driver/LastName

III.6 PurchaseCancel

In this section you will see the particularities of this operation for Car service. If you want to see the general specification for PurchaseCancel, please see [HotelBeds XML Interface Specification-v2.4.4](#).

III.6.1 RESPONSE BUSINESS RULES

If PurchaseCancelRS/Purchase element is present and PurchaseCancelRS/Purchase/ServiceList/Service/@type is ServiceCar the following elements and attributes must be provided:

- PurchaseCancelRS/Purchase/ServiceList/Service/@SPUI
- PurchaseCancelRS/Purchase/ServiceList/Service/Status
- PurchaseCancelRS/Purchase/ServiceList/Service/Supplier
- PurchaseCancelRS/Purchase/ServiceList/Service/DateFrom/@time
- PurchaseCancelRS/Purchase/ServiceList/Service/DateTo/@time
- PurchaseCancelRS/Purchase/ServiceList/Service/Driver/LastName

XML care hire integration

WHAT IS THE XML CAR HIRE INTEGRATION?

Our XML car service integration allows you to offer rent-a-car services worldwide, running requests with over 25 providers in the world's top tourist destinations.

A complete back office integration allows for full control of product search queries, modifications and cancellations, making the booking process easier for clients with multi-provider systems.

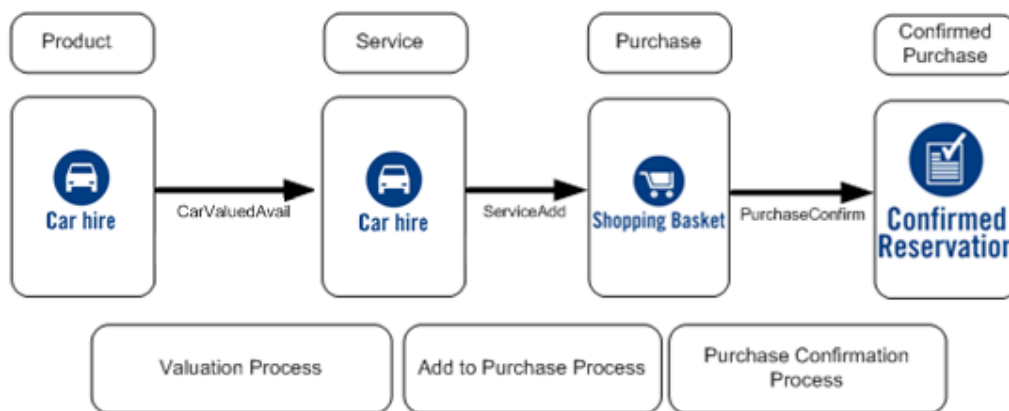
For more information, our extended XML car service integration documentation package is constantly updated, offering an in-depth description of all the functionalities, with practical examples and a FAQ section.

Interface development and integration

As part of the interface integration process, we'll provide a test environment with our entire product portfolio loaded where to run the necessary pre-live tests.

Once the XML integration is complete, we'll run a certification process of your application in order to minimise the risk of any platform errors, ensuring that your website, XML feed and system connection is correct and ready to deliver the sales.

WORKFLOW XML FEED



REQUIRED XML REQUESTS

CarCountryListRQ

<CarCountryList> returns a list of available countries for car services, including the destinations and area.

```

    <Country code="ES">
    <Name>SPAIN </Name>
    <DestinationList>
    <Destination type="SIMPLE" code="PMI">
    <Name>Majorca</Name>
    <ZoneList>
    <Zone type="SIMPLE" code="864">Aeropuerto Palma de Mallorca - Son San Juan</Zone>
    <Zone type="SIMPLE" code="365">Alcudia - Av Nicolau Riera Marsa 1</Zone>
  
```

```

<Zone type="SIMPLE" code="363">Cala D Or Mallorca - Avd Bienvenidos,74</Zone>
<Zone type="SIMPLE" code="1343">Cala Millor - Sol Naixent 14</Zone>
<Zone type="SIMPLE" code="615">Paseo Maritimo, 16 (07014 Palma De Mallorca)</Zone>
<Zone type="SIMPLE" code="616">Santa Ponsa - Av Rey Don Jaime 98 Local 3</Zone>
</ZoneList>
</Destination>
</DestinationList>
</Country>

```

CarInfoSetRQ

< CarInfoSet > returns the rent-a-car office locations, car types and car special equipment extras available in each destination.

```

<CarOfficeList>
<CarOffice xsi:type="ProductCarOffice">
<Code>21055</Code>
<Name>Aeropuerto Palma de Mallorca - Son San Juan</Name>
<Contact>
<Address/>
</Contact>
<Destination type="SIMPLE" code="PMI">
<Name>Majorca</Name>
<ZoneList>
<Zone type="SIMPLE" code="864">Aeropuerto Palma de Mallorca - Son San Juan</Zone>
</ZoneList>
</Destination>
<TermsAndCond type="TANDCES">
http://production.rent-at-avis.com/avisonline/terms.nsf/TermsByCountryAndLngCategories/ES-ES-Common?OpenDocument&INTERNATIONAL=Y
</TermsAndCond>
</CarOffice>
</CarOfficeList>
<CarGroupList>
<CarGroup code="2">Mini - Smart for Two or similar</CarGroup>
<CarGroup code="3">Economy - Nissan Micra, Polo or similar</CarGroup>
<CarGroup code="4">Compact - Opel Astra, Peugeot 308 or similar</CarGroup>
<CarGroup code="5">Intermediate - Opel Zafira, VW Golf or similar</CarGroup>
<CarGroup code="6">Minivan - VW Touran, Mercedes E or similar</CarGroup>
<CarGroup code="1">Standard - Alfa 159, Chrysler or similar</CarGroup>
</CarGroupList>

```

CarValuedAvailRQ

<CarValuedAvail > returns a list of features available in each car (transmission type, air conditioning, number of seats, etc.). Note: the returned prices are always final.

```

<DateFrom date="20101130" time="1000"/>
<DateTo date="20101202" time="1000"/>

```

```
<Currency code="EUR"/>
<TotalAmount>122.220</TotalAmount>

<TransmissionType>MANUAL</TransmissionType>
<AirConditioning>Y</AirConditioning>
<Vendor code="11">AVIS</Vendor>
<DoorCount>3</DoorCount>
<ABS>Y</ABS>
<Airbag>Y</Airbag>
<PowerSteering>Y</PowerSteering>
<SeatCount>5</SeatCount>
</CarInfo>
<CarGroup code="5">Intermediate - Opel Zafira, VW Golf or similar</CarGroup>
<RentalOffice xsi:type="ProductCarOffice">
<Code>864</Code>
<TermsAndCond type="terms_and_cond">
http://production.rent-at-avis.com/avisonline/terms.nsf/TermsByCountryAndLngCategories/ES-ES-Common?OpenDocument&INTERNATIONAL=Y
</TermsAndCond>
</RentalOffice>
<ReturnOffice xsi:type="ProductCarOffice">
<Code>864</Code>
</ReturnOffice>
```

Other requests:

ServiceAdd/Remove

PurchaseList/Detail/Confirm/Cancel/Flush

PRODUCT UPDATES

The product updates for tickets will need the following tables/.csv files:

- CarDestinations.csv (CarDestinations table)
- Countries.csv and also CountryIDs (Countries and CountryIDs tables)
- Destinations.csv and also DestinationsID.csv (Destinations and DestinationsIDs tables)

PRICING AND MARKUPS

Prices are always net.

Example price presentation returned via the XML feed

```
<ServiceExtraInfoList>
<ExtendedData type="EXT_ADDITIONAL_INFO">
<Name>INFO_CAR_RATE_DESCRIPTION</Name>
<Value>
Local tax (tax), collision damage waiver (cdw), theft protection (tp), vehicle registration fee
(vrf), airport/city surcharge, unlimited mileage.
</Value>
</ExtendedData>
</ServiceExtraInfoList>
<CarInfo xsi:type="ProductCar">
```

```

<Code>947</Code>
<Name>Renault Clio Confort 1.5</Name>
<ImageList>
<Image>
<Type>S</Type>
<Url>
http://www.hotelbeds.com/giata/avis/small/es0_b_sml01.gif
</Url>
<Image>
<Type>L</Type>
<Url>
http://www.hotelbeds.com/giata/avis/big/es0_b_lrg01.jpg
</Url>
</Image>
</ImageList>
<TransmissionType>MANUAL</TransmissionType>
<AirConditioning>Y</AirConditioning>
<Vendor code="11">AVIS</Vendor>
<DoorCount>3</DoorCount>
<ABS>N</ABS>
<Airbag>Y</Airbag>
<PowerSteering>Y</PowerSteering>
<SeatCount>5</SeatCount>
</CarInfo>

CommentList>
<Comment type="SERVICE">
In case the car is to be returned in a different office than the pick-up one, Avis might invoice
a "one-way fee" to the client. The amount of this fee can be consulted by calling one of the
following telephone numbers: Barcelona Call Centre: +34 93 344 3700 Manchester Call
Centre: +44 0870 6060100
</Comment>
</CommentList>
<DateFrom date="20101029" time="1115"/>
<DateTo date="20101031" time="1005"/>
<Currency code="EUR"/>
<TotalAmount>102.700</TotalAmount>
<ServiceExtraInfoList>
<ExtendedData type="EXT_ADDITIONAL_INFO">
<Name>INFO_CAR_RATE_DESCRIPTION</Name>
<Value>
Local tax (tax), collision damage waiver (cdw), theft protection (tp), vehicle registration fee
(vrf), airport/city surcharge, unlimited mileage.
</Value>
</ExtendedData>
</ServiceExtraInfoList>
<CarInfo xsi:type="ProductCar">
<Code>1093</Code>
<Name>Opel Astra 1.6</Name>

```

Pick up Rental country* SPAIN Rental destinations* Majorca Rental office* Cala Millor - Sol Naixent 14 From* 27/10/2010 03 15	Return Return country.* SPAIN Return destination* Majorca Return office* Paseo Maritimo, 16 (07014 Palma De Mall. To* 29/10/2010 hh mm
--	---

Groups: Mini - Smart for Two or similar Delete Search


* Mandatory fields

- Mini - Smart for Two or similar
- Economy - Nissan Micra, Polo or similar
- Compact - Opel Astra, Peugeot 308 or similar
- Intermediate - Opel Zafira, VW Golf or similar
- Standard - Alfa 159, Chrysler or similar
- Minivan - VW Touran, Mercedes E or similar

The final price of the service is the price you get in the valuation step (serviceAdd)

Rental details


Alternative car groups




Opel Astra 1.6 or similar

Price: 102,70 €

In case the car is to be returned in a different office than the pick-up one, Avis might invoice a "one-way fee" to the client. The amount of this fee can be consulted by calling one of the following telephone numbers: Barcelona Call Centre: +34 93 344 3700 Manchester Call Centre: +44 0870 6060100






Renault Clio Confort 1.5 or similar

Price: 80,04 €

In case the car is to be returned in a different office than the pick-up one, Avis might invoice a "one-way fee" to the client. The amount of this fee can be consulted by calling one of the following telephone numbers: Barcelona Call Centre: +34 93 344 3700 Manchester Call Centre: +44 0870 6060100



Offer includes:


- Local tax (tax), collision damage waiver (cdw), theft protection (tp), vehicle registration fee (vrf), airport/city surcharge, unlimited mileage.

Please consult the minimum required age of the driver, in the Terms and Conditions.
 Credit card is required to pick up the car.
 The tariff does not include additional services the driver might contract. The amount of these extra services can be consulted in the T&C

Price: 102,7 €

VOUCHER

Below, there's an example of a ticket voucher with the relevant information, as well as some XML examples with information required.



Client: Test Test
Locator: 102-1782998
***Agency contact:** 971 584 654
Agency reference: TEST

Service: RENTACAR
***Voucher No:** 102-1782998
***Account No:** AV875137750019
Avis Locator: 00000000US0
AWD No: X062201
IATA Number: 0174856E

Car	Rate	From	To	Days
Compact - Opel Astra, Peugeot 308 or similar (Opel Astra 1.6) o similar	7U	29/10/2010 11:15	31/10/2010 10:05	1 dia 22 horas 50 minutos

Remarks
Special Offer, free upgrade with categories B & C. Indicated car models are guide only, actual model may vary by destination booked. Voucher must be provided at the arrival. Booking date from 15th February to 30th April 2008. Exclusive offer for Mainland, Balearic and Canary islands. Subject to availability.

Rental location
Barcelona - Aeropuerto De Barcelona (El Prat) - 2934

Return location
Barcelona - Aeropuerto De Barcelona (El Prat) - 2934

Rate description
Local tax (tax), collision damage waiver (cdw), theft protection (tp), vehicle registration fee (vrf), airport/city surcharge, unlimited mileage. In case the car is to be returned in a different country than the one of collection, Avis might invoice a "one-way-fee" to the client. The amount of this fee can be consulted by calling one of the following telephone numbers:

terms and conditions:

CUSTOMER NOTE: A VALID CREDIT CARD WILL BE REQUIRED AT THE RENTAL LOCATION WHEN YOU PICK UP YOUR CAR TO WHICH EXTRAS (E.G. ADDITIONAL INSURANCES) CAN BE CHARGED.

RENTAL SALES AGENT NOTE: ALL THE INFORMATION ON THIS VOUCHER MUST MATCH THE WIZARD RESERVATION. IF THERE ARE ANY CHANGES IT IS NOT VALID.

CAR RENTAL VOUCHER – TERMS AND CONDITIONS

"Avis" means Avis Rent a Car System, Inc. its affiliates and licensees. "Customer" means the individual whose details are set out in the "Renter's Name" box above. Issuer means the person who issues this voucher to the customer.

1.VOUCHER
This voucher does not constitute a reservation but is simply a record of the Customer's application for vehicle rental services.

2.TERMS OF RENTAL
When the Customer presents this voucher to obtain and/or pay for vehicle rental services offered by Avis he will be required to enter into a separate rental agreement with Avis. The agreement will govern the Customer's rights and responsibilities in relation to the vehicle rented. If the Customer has any queries or problems, he will need to take them up with the Avis company he rents from in the country in which the rental takes place or with the Avis company in the country of residence. [This voucher does not constitute a contract between the Customer and the Issuer for the Issuer to provide vehicle rental services.] Avis may refuse to rent to persons (a) under the applicable minimum age, (b) not in possession of a valid recognised driver's license, (c) who in Avis' opinion constitute an unreasonable risk or (d) unable to satisfy standard credit or cash qualification requirements applied by Avis from time to time.

```
LOCATOR: <Reference>
<FileNumber>1762832</FileNumber>
<IncomingOffice code="102"/>
</Reference>
```

```
AGENCY REFERENCE: <AgencyReference>TEST</AgencyReference>
```

```
DATES FROM/TO: <DateFrom date="20101122"/>
<DateTo date="20101122"/>
```

```
AVIS INFO: <ServiceExtraInfoList>
<ExtendedData type="EXT_ADDITIONAL_INFO">
<Name>INFO_AVIS_ACCOUNT_NUMBER</Name>
```

```
<Value>AV875137750019</Value>
</ExtendedData>
<ExtendedData type="EXT_ADDITIONAL_INFO">
<Name>INFO_AVIS_AWD_NUMBER</Name>
<Value>X062201</Value>
</ExtendedData>
<ExtendedData type="EXT_ADDITIONAL_INFO">
<Name>INFO_AVIS_IATA_CODE</Name>
<Value>0174856E</Value>
</ExtendedData>
<ExtendedData type="EXT_ADDITIONAL_INFO">
<Name>INFO_AVIS_VOUCHER_NUMBER</Name>
<Value>102-1782998</Value>
</ExtendedData>
<ExtendedData type="EXT_ADDITIONAL_INFO">
<Name>INFO_AVIS_RATE_DESCRIPTION</Name>
<Value>
Local tax (tax), collision damage waiver (cdw), theft protection (tp), vehicle registration fee
(vrf), airport/city surcharge, unlimited mileage.
</Value>
</ExtendedData>
<ExtendedData type="EXT_ADDITIONAL_INFO">
<Name>INFO_AVIS_RATE_CODE</Name>
<Value>7U</Value>
</ExtendedData>
</ServiceExtraInfoList>
CAR INFO: <CarGroup code="4">Compact - Opel Astra, Peugeot 308 or similar</CarGroup>
<Driver type="AD">
<CustomerId>1</CustomerId>
<Age>30</Age>
<Name>test</Name>
<LastName>test</LastName>
</Driver>
<RentalOffice xsi:type="ProductCarOffice">
<Code>2934</Code>
<Name>Barcelona - Aeropuerto De Barcelona (El Prat)</Name>
</RentalOffice>
-
<ReturnOffice xsi:type="ProductCarOffice">
<Code>2934</Code>
<Name>Barcelona - Aeropuerto De Barcelona (El Prat)</Name>
</ReturnOffice>
</Service>
```

Amendments to XML bookings

Below are the necessary tags needed to make an amendment.

To find a purchase to amend use <PurchaseList> and <PurchaseDetail>.

To add new services individually use <ServiceAdd>.

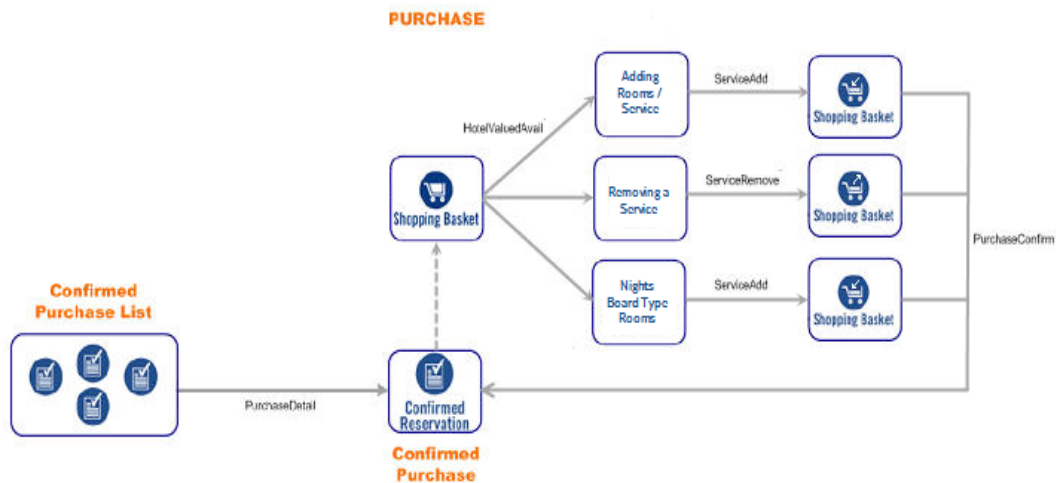
To remove an existing service use <ServiceRemove>.

To confirm the purchase with the new details use <PurchaseConfirm>.

Please note that you can only add to a purchase valued services and that Agency References, Holder or Paxes names can't be amended.

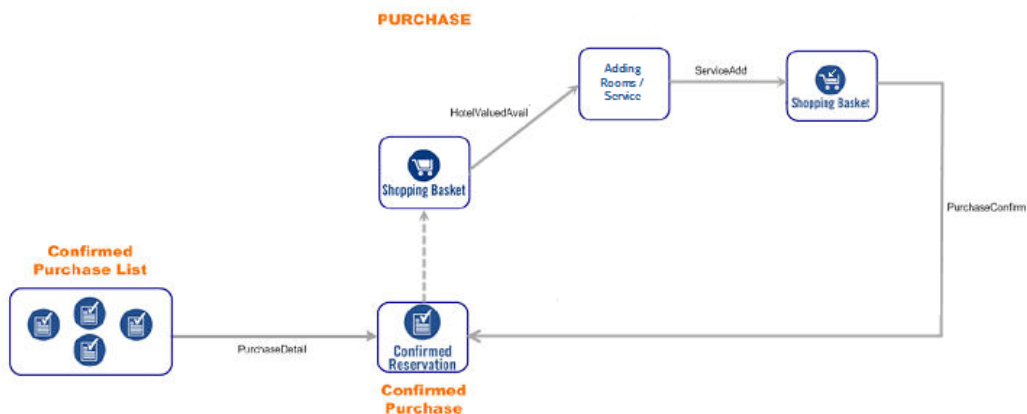
GENERAL AMENDMENTS WORKFLOW

If you want more information in XML format for all types of amendments, you'll find it in the file ROOMS_SERVICES.rar, also available in this USB.



ADDING ROOMS OR SERVICES TO A CONFIRMED PURCHASE

Workflow



You must find the confirmed booking (PurchaseDetail.xml), and check for availability in order to include the new service and / or room into the purchase (HotelValuedAvailRQ.xml and HotelValuedAvailRS.xml).

The example below shows an availability check for a single room in a different hotel:

```
<ServiceHotel xsi:type="ServiceHotel" availToken="atj1nqH0XtjewM4bR6ZfFg9H">
  <ContractList>
    <Contract>
      <Name>CG-TODOS</Name>
      <IncomingOffice code="102"/>
      <Classification code="NOR">Online Price</Classification>
    </Contract>
  </ContractList>
  <DateFrom date="20101121"/>
  <DateTo date="20101123"/>
  <Currency code="EUR">Euro</Currency>
  <HotelInfo xsi:type="ProductHotel">
    <Code>57</Code>
    <Name>Barcelona Universal</Name>
    <Category type="SIMPLE" code="4EST" shortname="4*">4 STARS</Category>
    <Destination type="SIMPLE" code="BCN">
      <Name>Barcelona</Name>
      <ZoneList>
        <Zone type="SIMPLE" code="44">Parallel</Zone>
      </ZoneList>
    </Destination>
    <ChildAge ageFrom="0" ageTo="12"/>
  </HotelInfo>
  <AvailableRoom>
    <HotelOccupancy>
      <RoomCount>1</RoomCount>
      <Occupancy>
        <AdultCount>1</AdultCount>
        <ChildCount>0</ChildCount>
      </Occupancy>
    </HotelOccupancy>
    <HotelRoom SHRUI="wGQirnwsB/xpEMT1gOo78Q==" availCount="10"
onRequest="N">
      <Board type="SIMPLE" code="BB-E10" shortname="BB">BED AND BREAKFAST</
Board>
      <RoomType type="SIMPLE" code="DUS-E10" characteristic="ST">DOUBLE/ TWIN
INDIVIDUAL USE STANDARD</RoomType>
      <Price>
        <Amount>204.500</Amount>
      </Price>
    </HotelRoom>
  </AvailableRoom>
</ServiceHotel>
```

After selecting the room, you need to retrieve the purchaseToken from the PurchaseDetail response and include it in <ServiceAdd> header (ServiceAddRQ.xml):

```
<ServiceAddRQ echoToken="DummyEchoToken" purchaseToken="TS095107518"
  xmlns="http://www.hotelbeds.com/schemas/2005/06/messages"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
  www.hotelbeds.com/schemas/2005/06/messages ServiceAddRQ.xsd">
[...]
```

To confirm the modified booking use <PurchaseConfirm>. In this case, as you are only amending a confirmed booking, you don't need to include any information in <ServiceData>.

Below is the <PurchaseConfirm> request (PurchaseConfirmRQ.xml):

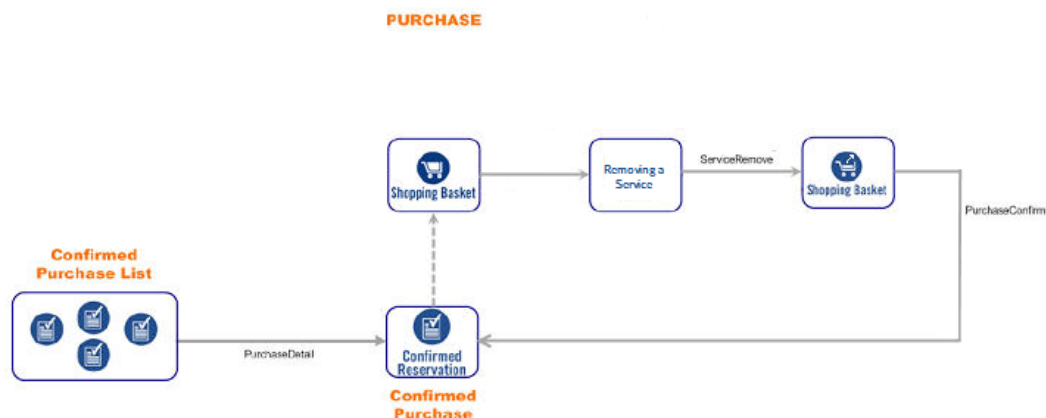
```
<PurchaseConfirmRQ echoToken="DummyEchoToken"
  xmlns="http://www.hotelbeds.com/schemas/2005/06/messages"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.hotelbeds.com/schemas/2005/06/messages
  PurchaseConfirmRQ.xsd">
  <Language>ENG</Language>
  <Credentials>
    <User>TEST</User>
    <Password>TEST</Password>
  </Credentials>
  <ConfirmationData purchaseToken="TS101734622">
  </ConfirmationData>
</PurchaseConfirmRQ>
```

Please note that as amendments to the Holder and AgencyReference are not allowed, this information is never included in amendments.

<purchaseToken> should be the same throughout the amendment process since you get it in PurchaseDetail.

After sending the confirmation (PurchaseConfirmRS.xml), the booking will be modified regarding the first one you had (please check PurchaseDetail_After_Amendment.xml).

REMOVING A SERVICE



To remove a service from a confirmed purchase, it's not necessary to check for availability as we are going to modify an existent service.

In the example below we are going to remove the service with SPUI="102#H#1" (PurchaseDetail.xml).

To do that, we need:

1-a <ServiceRemove> request specifying the SPUI we want to remove and
2-the <purchaseToken> we just got from <PurchaseDetail> (ServiceRemoveRQ.xml):

```
<ServiceRemoveRQ SPUI="102#H#1" echoToken="DummyEchoToken"
purchaseToken="TS105117805"
xmlns="http://www.hotelbeds.com/schemas/2005/06/messages"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.hotelbeds.com/schemas/2005/06/messages
ServiceRemoveRQ.xsd">
  <Language>ENG</Language>
  <Credentials>
    <User>TEST</User>
    <Password>TEST</Password>
  </Credentials>
</ServiceRemoveRQ>
```

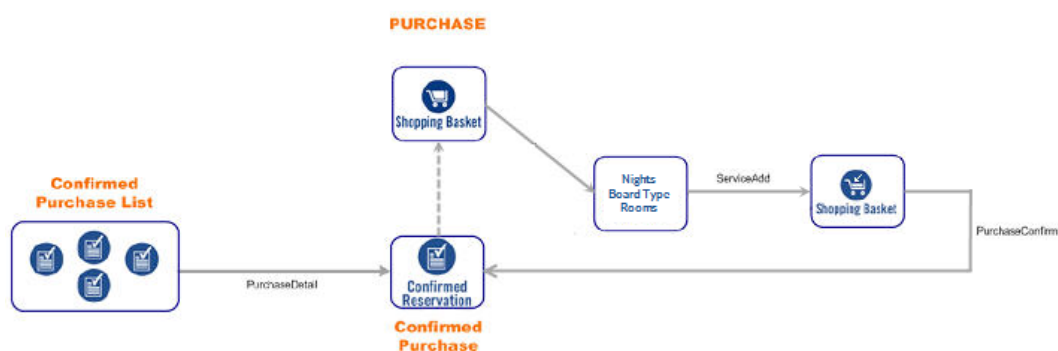
We'll receive the response from <ServiceRemove> with the selected room with status CANCELLED:

```
<HotelRoom SHRUI="fE4c7qKsm08+Gyn9dTjUQA=="
availCount="1" status="CANCELLED">
  <Board type="SIMPLE" code="RO-E10">ROOM ONLY</Board>
  <RoomType type="SIMPLE" code="DBL-E10"
characteristic="ST">DOUBLE STANDARD</RoomType>
  <Price>
    <Amount>0.000</Amount>
  </Price>
```

Note that <purchaseToken> should be the same in all the amendment process since you get it in <PurchaseDetail>.

NOTE: If the room you are going to cancel has cancellation fees at the moment of cancellation, you will receive these cancellation fees in confirmation response as a supplement and they will be added to the final price of the booking. This applies to all services cancelled through the XML.

MODIFYING NIGHTS / BOARD TYPE / ROOMS



MODIFYING NIGHTS

To remove a service from a confirmed purchase, it's not necessary to check for availability as we are going to modify an existent service.

In the example below we are going to remove the service with SPUI="102#H#2" (PurchaseDetail.xml).

To make the amendment, you need to include the new booking details in the <ServiceAdd> request (ServiceAdd_RQ.xml and ServiceAdd_RS.xml).

In this case, the booking is from 21/11/2010 to 23/11/2010. We're going to amend it with the following details: check-in 20/11/2010, check out 25/11/2010.

In the <ServiceAdd> request you should inform the <purchaseToken> in the header of the request.

```
<ServiceAddRQ echoToken="DummyEchoToken" purchaseToken="TS095325953"
  xmlns="http://www.hotelbeds.com/schemas/2005/06/messages"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
www.hotelbeds.com/schemas/2005/06/messages ServiceAddRQ.xsd">
  <Language>ENG</Language>
  <Credentials>
    <User>TEST</User>
    <Password>TEST</Password>
  </Credentials>
  <Service SPUI="102#H#2" xsi:type="ServiceHotel">
    <ContractList>
      <Contract>
        <Name>CG-TODOS</Name>
        <IncomingOffice code="102"/>
      </Contract>
    </ContractList>
    <DateFrom date="20101120"/>
    <DateTo date="20101125"/>
    <Currency code="EUR"/>
    <HotelInfo xsi:type="ProductHotel">
      <Code>149</Code>
      <Destination code="BCN" type="SIMPLE"/>
    </HotelInfo>
    <AvailableRoom>
    <HotelOccupancy>
      <RoomCount>1</RoomCount>
      <Occupancy>
        <AdultCount>1</AdultCount>
        <ChildCount>0</ChildCount>
      </Occupancy>
    </HotelOccupancy>
    <HotelRoom modifiesSHRUI="N7IIaUsI42IXGwkMNH5HQw==" onRequest="N">
      <Board code="BB-E10" type="SIMPLE"/>
      <RoomType characteristic="ST" code="DUS-E10" type="SIMPLE"/>
    </HotelRoom>
    </AvailableRoom>
  </Service>
</ServiceAddRQ>
```

Once you receive the <ServiceAdd> response with Status MODIFIED and the dates changed, you can continue with the confirmation of the modification.

EXTRACT FROM SERVICE ADD RESPONSE:

```
<Purchase purchaseToken="TS095325953" timeToExpiration="608594">
  <Reference>
    <FileNumber>1538064</FileNumber>
    <IncomingOffice code="102"/>
  </Reference>
  <Status>MODIFIED</Status>
  <Agency>
    <Code>3345</Code>
    <Branch>1</Branch>
  </Agency>
  <Language>ENG</Language>
  <CreationDate date="20100921"/>
  <CreationUser>TEST</CreationUser>
  <Holder type="AD">
    <Age>0</Age>
    <Name>TEST</Name>
    <LastName>TEST</LastName>
  </Holder>
  <AgencyReference>TEST AGENCYREF</AgencyReference>
  <ServiceList>
    <Service xsi:type="ServiceHotel" SPUI="102#H#2">
      <Reference>
        <FileNumber>1538064-H2</FileNumber>
        <IncomingOffice code="102"/>
      </Reference>
      <Status>MODIFIED</Status>
      <ContractList>
        <Contract>
          <Name>CG-TODOS</Name>
          <IncomingOffice code="102"/>
        </Contract>
      </ContractList>
      <Supplier name="HOTELBEDS PRODUCT,S.L.U." vatNumber="B38877676"/>
      <DateFrom date="20101120"/>
      <DateTo date="20101125"/>
    </Service>
  </ServiceList>
  [...]
</Purchase>
```

PURCHASE CONFIRM REQUEST:

```
<PurchaseConfirmRQ echoToken="DummyEchoToken"
  xmlns="http://www.hotelbeds.com/schemas/2005/06/messages"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
  www.hotelbeds.com/schemas/2005/06/messages PurchaseConfirmRQ.xsd">
  <Language>ENG</Language>
```



```
<Credentials>
  <User>TEST</User>
  <Password>TEST</Password>
</Credentials>
<ConfirmationData purchaseToken="TS095325953">
</ConfirmationData>
</PurchaseConfirmRQ>
```

Note that purchaseToken should be the same in all the amendment process since you get it in PurchaseDetail.

MODIFYING BOARD TYPE

To add a new board type to a confirmed purchase, it's not necessary to request availability as we are going to modify an existent service, but we recommend doing it in order for you to know the different board types available.

In the example below we are going to add the new board type to the service with SPUI="102#H#2" (PurchaseDetail.xml).

To do the amendment, you need to include in the <ServiceAdd> request the new information about board type (ServiceAdd_RQ.xml and ServiceAdd_RS.xml).

In the example below, the booking has a bed and breakfast basis and we are going to change it to room only basis.

In the <ServiceAdd> request you should inform the <purchaseToken> in the header of the request.

```
<ServiceAddRQ echoToken="DummyEchoToken" purchaseToken="TS121310425"
xmlns="http://www.hotelbeds.com/schemas/2005/06/messages"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.
hotelbeds.com/schemas/2005/06/messages ServiceAddRQ.xsd">
  <Language>ENG</Language>
  <Credentials>
    <User>TEST</User>
    <Password>TEST</Password>
  </Credentials>
  <Service SPUI="102#H#2" xsi:type="ServiceHotel">
    <ContractList>
      <Contract>
        <Name>CG-TODOS</Name>
        <IncomingOffice code="102"/>
      </Contract>
    </ContractList>
    <DateFrom date="20101120"/>
    <DateTo date="20101125"/>
    <Currency code="EUR"/>
    <HotelInfo xsi:type="ProductHotel">
      <Code>149</Code>
      <Destination code="BCN" type="SIMPLE"/>
    </HotelInfo>
    <AvailableRoom>
```

```

<HotelOccupancy>
<RoomCount>1</RoomCount>
<Occupancy>
<AdultCount>1</AdultCount>
<ChildCount>0</ChildCount>
</Occupancy>
</HotelOccupancy>
<HotelRoom modifiesSHRUI="Ku+IPPtsb4JOwnNX/dG7gQ==" onRequest="N">
<Board code="RO-E10" type="SIMPLE"/>
<RoomType characteristic="ST" code="DUS-E10" type="SIMPLE"/>
</HotelRoom>
</AvailableRoom>
</Service>
</ServiceAddRQ>

```

Once you receive the <ServiceAdd> response with Status MODIFIED and the dates changed, you can continue with the confirmation of the modification.

EXTRACT FROM SERVICE ADD RESPONSE:

```

<Purchase purchaseToken="TS121310425" timeToExpiration="1752808">
<Reference>
<FileNumber>1538064</FileNumber>
<IncomingOffice code="102"/>
</Reference>
<Status>MODIFIED</Status>
[...]
<AvailableRoom>
<HotelOccupancy>
<RoomCount>1</RoomCount>
<Occupancy>
<AdultCount>1</AdultCount>
<ChildCount>0</ChildCount>
<GuestList>
<Customer type="AD">
<CustomerId>3</CustomerId>
<Age>30</Age>
<Name>TEST</Name>
<LastName>TEST</LastName>
</Customer>
</GuestList>
</Occupancy>
</HotelOccupancy>
<HotelRoom SHRUI="Ku+IPPtsb4JOwnNX/dG7gQ==" availCount="1" status="MODIFIED">
<Board type="SIMPLE" code="RO-E10">ROOM ONLY</Board>
<RoomType type="SIMPLE" code="DUS-E10" characteristic="ST">DOUBLE/ TWIN INDIVIDUAL
USE STANDARD</RoomType>
<Price>
<Amount>622.050</Amount>
</Price>

```

PURCHASE CONFIRM REQUEST:

```
<PurchaseConfirmRQ echoToken="DummyEchoToken"
  xmlns="http://www.hotelbeds.com/schemas/2005/06/messages"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
www.hotelbeds.com/schemas/2005/06/messages PurchaseConfirmRQ.xsd">
  <Language>ENG</Language>
  <Credentials>
    <User>TEST</User>
    <Password>TEST</Password>
  </Credentials>
  <ConfirmationData purchaseToken=" TS121310425">
  </ConfirmationData>
</PurchaseConfirmRQ>
```

Note that purchaseToken should be the same in all the amendment process since you get it in PurchaseDetail.

MODIFYING A ROOM

To modify a room from a confirmed purchase you have to request availability, in order to know if there is availability in other rooms. It's also possible to make the amendment without requesting availability if you already know the rooms you want.

In the example below we are going to modify a room with SPUI="102#H#2" (PurchaseDetail.xml).

As you're going to add a new room, you need to make a request to <HotelValuedAvail> (HotelValuedAvail_RQ.xml and HotelValuedAvail_RS.xml).

In order to modify the room, it's necessary to include the new room information in the ServiceAdd request (ServiceAdd_RQ.xml and ServiceAdd_RS.xml). In this case, the booking is for a double room and we are going to change it into a triple room.

In the <ServiceAdd> request you should inform the <purchaseToken> in the header of the request.

```
<ServiceAddRQ echoToken="DummyEchoToken" purchaseToken="TS011458703"
  xmlns="http://www.hotelbeds.com/schemas/2005/06/messages"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.
hotelbeds.com/schemas/2005/06/messages ServiceAddRQ.xsd">
  <Language>ENG</Language>
  <Credentials>
    <User>TEST</User>
    <Password>TEST</Password>
  </Credentials>
  <Service SPUI="102#H#1" availToken="0GXWTA3M4upvTOjCb/fvxAcQ"
  xsi:type="ServiceHotel">
  <ContractList>
  <Contract>
  <Name>CG-BAR 23</Name>
  <IncomingOffice code="102"/>
```

```

</Contract>
</ContractList>
<DateFrom date="20101121"/>
<DateTo date="20101123"/>
<Currency code="EUR"/>
<HotelInfo xsi:type="ProductHotel">
<Code>69</Code>
<Destination code="BCN" type="SIMPLE"/>
</HotelInfo>
<AvailableRoom>
<HotelOccupancy>
<RoomCount>1</RoomCount>
<Occupancy>
<AdultCount>2</AdultCount>
<ChildCount>0</ChildCount>
<GuestList>
<Customer type="AD">
<Age>45</Age>
</Customer>
</GuestList>
</Occupancy>
</HotelOccupancy>
<HotelRoom modifiesSHRUI="fE4c7qKsm08+Gyn9dTjUQA==" onRequest="N">
<Board code="RO-E10" type="SIMPLE"/>
<RoomType characteristic="ST" code="TPL-E10" type="SIMPLE"/>
</HotelRoom>
</AvailableRoom>
</Service>
</ServiceAddRQ>

```

The <PurchaseConfirm> request is the same as in the other cases, but with the new purchaseToken received in <PurchaseDetail> response.

OTHER TYPES OF AMENDMENTS

You can also modify rooms of the same type. For example, cancelling 2 rooms from a confirmed booking of 3 rooms.

For this type of amendment, you don't need to make <HotelValuedAvail> request. You can send directly the <ServiceAdd> with the information about purchase token in the header of the request.

```

<ServiceAddRQ echoToken="DummyEchoToken" purchaseToken="TS014548810"
xmlns="http://www.hotelbeds.com/schemas/2005/06/messages"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.
hotelbeds.com/schemas/2005/06/messages ServiceAddRQ.xsd">
<Language>ENG</Language>
<Credentials>
<User>TEST</User>

```

```
<Password>TEST</Password>
</Credentials>
<Service SPUI="102#H#1" xsi:type="ServiceHotel">
<ContractList>
<Contract>
<Name>CG-BAR 23</Name>
<IncomingOffice code="102"/>
</Contract>
</ContractList>
<DateFrom date="20101121"/>
<DateTo date="20101123"/>
<Currency code="EUR"/>
<HotelInfo xsi:type="ProductHotel">
<Code>69</Code>
<Destination code="BCN" type="SIMPLE"/>
</HotelInfo>
<AvailableRoom>
<HotelOccupancy>
<RoomCount>2</RoomCount>
<Occupancy>
<AdultCount>2</AdultCount>
<ChildCount>0</ChildCount>
<GuestList>
<Customer type="AD">
<Age>45</Age>
</Customer>
</GuestList>
</Occupancy>
</HotelOccupancy>
<HotelRoom modifiesSHRUI="yyyaDMjLD7LeR2Me5jC9RA==" status="MODIFIED">
<Board code="RO-E10" type="SIMPLE"/>
<RoomType characteristic="ST" code="TPL-E10" type="SIMPLE"/>
</HotelRoom>
</AvailableRoom>
</Service>
</ServiceAddRQ>
```

EXTRACT FROM SERVICE ADD RESPONSE:

```
<Purchase purchaseToken="TS014548810" timeToExpiration="1697304">
<Reference>
<FileNumber>1538404</FileNumber>
<IncomingOffice code="102"/>
</Reference>
<Status>MODIFIED</Status>
[...]
```

```
<AvailableRoom>
<HotelOccupancy>
<RoomCount>2</RoomCount>
```

```

<Occupancy>
<AdultCount>2</AdultCount>
<ChildCount>0</ChildCount>
<GuestList>
<Customer type="AD">
<CustomerId>3</CustomerId>
<Age>30</Age>
<Name>TEST</Name>
<LastName>TEST</LastName>
</Customer>
<Customer type="AD">
<CustomerId>4</CustomerId>
<Age>30</Age>
</Customer>
<Customer type="AD">
<CustomerId>5</CustomerId>
<Age>30</Age>
</Customer>
<Customer type="AD">
<CustomerId>6</CustomerId>
<Age>30</Age>
</Customer>
</GuestList>
</Occupancy>
</HotelOccupancy>
<HotelRoom SHRUI="1ULfz5lsfwS4cyByOKYhoQ==" availCount="2" status="MODIFIED">
<Board type="SIMPLE" code="RO-E10">ROOM ONLY</Board>
<RoomType type="SIMPLE" code="TPL-E10" characteristic="ST">TRIPLE STANDARD</
RoomType>
<Price>
<Amount>383.720</Amount>
</Price>
[...]
```

You can also choose which of the paxes will stay in the booking by selecting them at the <ServiceAdd> step of the amendment:

```

<ServiceAddRQ echoToken="DummyEchoToken" purchaseToken="TS053747461"
xmlns="http://www.hotelbeds.com/schemas/2005/06/messages"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.
hotelbeds.com/schemas/2005/06/messages ServiceAddRQ.xsd">
[...]
```

```

<AvailableRoom>
<HotelOccupancy>
<RoomCount>1</RoomCount>
<Occupancy>
<AdultCount>2</AdultCount>
<ChildCount>0</ChildCount>
<GuestList>
```

```
<Customer type="AD">
<CustomerId>1</CustomerId>
<Age>0</Age>
<Name>TEST</Name>
<LastName>TEST</LastName>
</Customer>
<Customer type="AD">
<CustomerId>6</CustomerId>
<Age>0</Age>
<Name>TESTA</Name>
<LastName>TEST</LastName>
</Customer>
</GuestList>
</Occupancy>
</HotelOccupancy>
<HotelRoom modifiesSHRUI="HWRdSMARtc7ho5zgat+zlg==" status="MODIFIED">
<Board code="HD-E10" type="SIMPLE"/>
<RoomType characteristic="ST" code="DBL-E10" type="SIMPLE"/>
</HotelRoom>
</AvailableRoom>
</Service>
</ServiceAddRQ>
```



Transfer Service XML Interface Specification **developer's guide**

v1.0.1

June 10

Hotelbeds Accommodation & Destination Services
Camí de Son Fangós, 100 - Torre A, 5ª Planta
07007 - Palma de Mallorca
Spain

I.	Document Control	3
II.	Operations.....	4
II.1	TransferValuedAvail.....	4
II.1.1	Request business rules.....	4
II.1.2	Request Observations	5
II.1.3	Request Acceptance Test.....	5
II.1.4	Response business rules.....	5
II.1.5	Response Observations	6
III.	Particularities of common operations	7
III.1	ServiceAdd.....	7
III.1.1	Request Business Rules.....	7
III.1.2	Request Observations	7
III.1.3	Request Acceptance Test.....	7
III.1.4	Response Business Rules.....	8
III.2	ServiceRemove	8
III.2.1	Response Business Rules.....	8
III.3	PurchaseConfirm	9
III.3.1	Request important Notices.....	9
III.3.2	Request Acceptance Test.....	9
III.3.3	Response Business Rules.....	10
III.4	PurchaseDetail.....	10
III.4.1	Response Business Rules.....	10
III.5	PurchaseList	10
III.5.1	Response Business Rules.....	10
III.6	PurchaseCancel.....	12
III.6.1	Response Business Rules.....	12
IV.	FAQ	13

I. Document Control

1.0.1	02/06/10	<ul style="list-style-type: none"> - New optional element (HotelbedsCommonTypes.xsd): ServiceTransfer/CancellationPolicy - Modified section Response Business Rules from ServiceAdd operation. - Modified section Response Business Rules from PurchaseConfirm operation. - Modified section Response Business Rules from PurchaseDetail operation. - CancelProtection section removed. 	1
1.0.0	14/04/08	New documentation file	1

II. Operations

II.1 TransferValuedAvail

Get a list of available transfer services according to the requested parameters. The returned prices are final prices.

TransferValuedAvail

Details:	xsd docs/TransferValuedAvailRQ.html	xsd docs/TransferValuedAvailRS.html
Schema:	xml/TransferValuedAvailRQ.xsd	xml/TransferValuedAvailRS.xsd
Example:	xml/TransferValuedAvailRQ.xml	xml/TransferValuedAvailRS.xml

II.1.1 REQUEST BUSINESS RULES

Language	Mandatory
AvailData/ServiceDate/@date	At least today
AvailData/ServiceDate/@time	Mandatory

If TransferValuedAvailRQ/AvailData/Occupancy/GuestList/Customer/@type attribute value is CH, element TransferValuedAvailRQ/AvailData/Occupancy/GuestList/ Customer/Age will be mandatory.

If two TransferValuedAvailRQ/AvailData elements are provided one must have the TransferValuedAvailRQ/AvailData/@type attribute set to IN and the other to OUT.

If TransferValuedAvailRQ/AvailData/PickupLocation/@type is ProductTransferHotel:

- TransferValuedAvailRQ/AvailData/PickupLocation/Code element is mandatory.
- TransferValuedAvailRQ/AvailData/PickupLocation/TransferZone/Code element is mandatory.

If TransferValuedAvailRQ/AvailData/PickupLocation/@type is ProductTransferTerminal:

- TransferValuedAvailRQ/AvailData/PickupLocation/Code element is mandatory.
- TransferValuedAvailRQ/AvailData/PickupLocation/DateTime/@date element is mandatory.
- TransferValuedAvailRQ/AvailData/PickupLocation/DateTime/@time element is mandatory.

If TransferValuedAvailRQ/AvailData/PickupLocation/@type is ProductZone:

- TransferValuedAvailRQ/AvailData/PickupLocation/Code element is mandatory.

If TransferValuedAvailRQ/AvailData/DestinationLocation/@type is ProductTransferHotel:

- TransferValuedAvailRQ/AvailData/DestinationLocation/Code element is mandatory.
- TransferValuedAvailRQ/AvailData/DestinationLocation/TransferZone/Code element is mandatory.

If TransferValuedAvailRQ/AvailData/DestinationLocation/@type is ProductTransferTerminal:

- TransferValuedAvailRQ/AvailData/DestinationLocation/Code element is mandatory.
- TransferValuedAvailRQ/AvailData/DestinationLocation/DateTime/@date element is mandatory.

- TransferValuedAvailRQ/AvailData/DestinationLocation/DateTime/@time element is mandatory.

If TransferValuedAvailRQ/AvailData/DestinationLocation/@type is ProductZone:

- TransferValuedAvailRQ/AvailData/DestinationLocation/Code element is mandatory.

II.1.2 REQUEST OBSERVATIONS

II.1.3 REQUEST ACCEPTANCE TEST

The following elements and attributes are useless:

- TransferValuedAvailRQ/ExtraParamList
- TransferValuedAvailRQ/AvailData/Occupancy/GuestList/Customer/CustomerId
- TransferValuedAvailRQ/AvailData/Occupancy/GuestList/Customer/Name
- TransferValuedAvailRQ/AvailData/Occupancy/GuestList/Customer/LastName
- TransferValuedAvailRQ/AvailData/Occupancy/GuestList/Customer/AdditionalInf
- TransferValuedAvailRQ/AvailData/Occupancy/GuestList/Customer/Document
- TransferValuedAvailRQ/AvailData/Occupancy/GuestList/Customer/BirthDate
- TransferValuedAvailRQ/AvailData/Occupancy/GuestList/Customer/CountryCode
- TransferValuedAvailRQ/AvailData/PickupLocation/Name
- TransferValuedAvailRQ/AvailData/PickupLocation/DescriptionList
- TransferValuedAvailRQ/AvailData/PickupLocation/ImageList
- TransferValuedAvailRQ/AvailData/PickupLocation/Contact
- TransferValuedAvailRQ/AvailData/PickupLocation/TransferZone/Name
- TransferValuedAvailRQ/AvailData/PickupLocation/TransferZone/DescriptionList
- TransferValuedAvailRQ/AvailData/PickupLocation/TransferZone/ImageList
- TransferValuedAvailRQ/AvailData/PickupLocation/TransferZone/Contact
- TransferValuedAvailRQ/AvailData/DestinationLocation/Name
- TransferValuedAvailRQ/AvailData/DestinationLocation/DescriptionList
- TransferValuedAvailRQ/AvailData/DestinationLocation/ImageList
- TransferValuedAvailRQ/AvailData/DestinationLocation/Contact
- TransferValuedAvailRQ/AvailData/DestinationLocation/TransferZone/Name
- TransferValuedAvailRQ/AvailData/DestinationLocation/TransferZone/DescriptionList
- TransferValuedAvailRQ/AvailData/DestinationLocation/TransferZone/ImageList
- TransferValuedAvailRQ/AvailData/DestinationLocation/TransferZone/Contact

Furthemore:

- TransferValuedAvailRQ/AvailData/Occupancy/GuestList/Customer/Age is useless for adult customers.

II.1.4 RESPONSE BUSINESS RULES

If TransferValuedAvailRS/ServiceTransfer element is present the following elements and attributes must be provided:

- TransferValuedAvailRS/ServiceTransfer/@availToken
- TransferValuedAvailRS/ServiceTransfer/@transferType
- TransferValuedAvailRS/ServiceTransfer/Currency
- TransferValuedAvailRS/ServiceTransfer/DestinationLocation/Code
- TransferValuedAvailRS/ServiceTransfer/ContractList
- TransferValuedAvailRS/ServiceTransfer/Paxes
- TransferValuedAvailRS/ServiceTransfer/Paxes/AdultCount
- TransferValuedAvailRS/ServiceTransfer/Paxes/ChildCount

II.1.5 RESPONSE OBSERVATIONS

If TicketValuationRS/ServiceTicket/Paxes/GuestList/Customer/@type attribute value is CH, element TicketValuationRQ/ServiceOccupancy/GuestList/Customer/Age will be mandatory.

III. Particularities of common operations

III.1 ServiceAdd

In this section you will see the particularities of this operation for Transfer service. If you want to see the general specification for ServiceAdd, please see [HotelBeds XML Interface Specification-v2.4.4](#).

III.1.1 REQUEST BUSINESS RULES

When you try to add a TransferService:

Service/DateFrom/@time	Mandatory
Service/TransferInfo/Code	Mandatory
Service/Paxes/GuestList/Customer/Age	Mandatory if Service/Paxes/GuestList/Customer/@type attribute is CH
Service/TravelInfo/TravelNumber	Mandatory if Service/TravelInfo element is present

III.1.2 REQUEST OBSERVATIONS

To add transfer services you must specify at ServiceTransfer/TravelInfo the elements TravelNumber, DepartInfo and ArrivalInfo. If you don't provide this information in this step, you must provide it in the PurchaseConfirm request operation call.

In addition, if transfer origin or transfer destination is a zone, you must specify at ServiceTransfer/PickupLocation or at ServiceTransfer/DestinationLocation the elements HotelName, HotelAddress and HotelPostalCode. If you don't provide this information in this step, you must provide it in the PurchaseConfirm request operation call. If you don't do it, confirmation request will fail.

III.1.3 REQUEST ACCEPTANCE TEST

The following elements and attributes are useless:

- ServiceAddRQ/Service/DateTo
- ServiceAddRQ/TransferInfo/Name
- ServiceAddRQ/TransferInfo/DescriptionList
- ServiceAddRQ/TransferInfo/ImageList
- ServiceAddRQ/TransferInfo/Contact
- ServiceAddRQ/Paxes/GuestList/Customer/CustomerId
- ServiceAddRQ/Paxes/GuestList/Customer/Name
- ServiceAddRQ/Paxes/GuestList/Customer/LastName
- ServiceAddRQ/Paxes/GuestList/Customer/AdditionalInfo
- ServiceAddRQ/Paxes/GuestList/Customer/AdditionalInfo
- ServiceAddRQ/Paxes/GuestList/Customer/Document
- ServiceAddRQ/Paxes/GuestList/Customer/BirthDate
- ServiceAddRQ/Paxes/GuestList/Customer/CountryCode
- ServiceAddRQ/Paxes/GuestList/Customer/Age is useless for adult customers
- ServiceAddRQ/Service/PickupLocation/Name
- ServiceAddRQ/Service/PickupLocation/DescriptionList
- ServiceAddRQ/Service/PickupLocation/ImageList
- ServiceAddRQ/Service/PickupLocation/Contact

- ServiceAddrQ/Service/DestinationLocation/Name
- ServiceAddrQ/Service/DestinationLocation/DescriptionList
- ServiceAddrQ/Service/DestinationLocation/ImageList
- ServiceAddrQ/Service/DestinationLocation/Contact
- ServiceAddrQ/Service/TravelInfo/ArrivalInfo/Name
- ServiceAddrQ/Service/TravelInfo/ArrivalInfo/DescriptionList
- ServiceAddrQ/Service/TravelInfo/ArrivalInfo/ImageList
- ServiceAddrQ/Service/TravelInfo/ArrivalInfo/Contact
- ServiceAddrQ/Service/TravelInfo/ArrivalInfo/TransferZone
- ServiceAddrQ/Service/TravelInfo/ArrivalInfo/DateTime
- ServiceAddrQ/Service/TravelInfo/ArrivalInfo/Country
- ServiceAddrQ/Service/TravelInfo/DepartInfo/Name
- ServiceAddrQ/Service/TravelInfo/DepartInfo/DescriptionList
- ServiceAddrQ/Service/TravelInfo/DepartInfo/ImageList
- ServiceAddrQ/Service/TravelInfo/DepartInfo/Contact
- ServiceAddrQ/Service/TravelInfo/DepartInfo/TransferZone
- ServiceAddrQ/Service/TravelInfo/DepartInfo/DateTime
- ServiceAddrQ/Service/TravelInfo/DepartInfo/Country
- ServiceAddrQ/Service/CancellationPolicy

III.1.4 RESPONSE BUSINESS RULES

If ServiceAddrRS/Purchase element is present and ServiceAddrRS/Purchase/ServiceList/Service/@type is ServiceTransfer the following elements and attributes must be provided:

- ServiceAddrRS/Purchase/ServiceList/Service/@SPUI
- ServiceAddrRS/Purchase/ServiceList/Service/@transferType
- ServiceAddrRS/Purchase/ServiceList/Service/Status
- ServiceAddrRS/Purchase/ServiceList/Service/Supplier
- ServiceAddrRS/Purchase/ServiceList/Service/ContractList
- ServiceAddrRS/Purchase/ServiceList/Service/PickupLocation/Code
- ServiceAddrRS/Purchase/ServiceList/Service/DestinationLocation/Code

In addition, if the transfer service returned has got cancellation fees, the following element must be provided:

- ServiceAddrRS/Purchase/ServiceList/Service/CancellationPolicy

III.2 ServiceRemove

In this section you will see the particularities of this operation for Transfer service. If you want to see the general specification for ServiceRemove, please see [HotelBeds XML Interface Specification-v2.4.4](#).

III.2.1 RESPONSE BUSINESS RULES

If ServiceRemoveRS/Purchase element is present and ServiceRemoveRS/Purchase/ServiceList/Service/@type is ServiceTransfer the following elements and attributes must be provided:

- ServiceRemoveRS/Purchase/ServiceList/Service/@SPUI
- ServiceRemoveRS/Purchase/ServiceList/Service/@transferType
- ServiceRemoveRS/Purchase/ServiceList/Service/Status
- ServiceRemoveRS/Purchase/ServiceList/Service/Supplier
- ServiceRemoveRS/Purchase/ServiceList/Service/ContractList
- ServiceRemoveRS/Purchase/ServiceList/Service/PickupLocation/Code
- ServiceRemoveRS/Purchase/ServiceList/Service/DestinationLocation/Code

III.3 PurchaseConfirm

In this section you will see the particularities of this operation. If you want to see the general specification for PurchaseConfirm, please see [HotelBeds XML Interface Specification-v2.4.4](#).

III.3.1 REQUEST IMPORTANT NOTICES

To confirm transfer services you can use operation ServiceAdd call including the element ServiceTransfer/TravelInfo. If you don't do it, there is another way to indicate this information. You can use the operation PurchaseConfirm specifying at ConfirmationServiceDataTransfer the elements TravelNumber, DepartInfo and ArrivalInfo.

In addition, if transfer origin or transfer destination is a zone, you can use operation ServiceAdd call including the elements HotelName, HotelAddress and HotelPostalCode at ServiceTransfer/PickupLocation or at ServiceTransfer/DestinationLocation. If you don't provide this information in this step, you must provide it in the PurchaseConfirm request operation call. If you don't do it, confirmation request will fail.

III.3.2 REQUEST ACCEPTANCE TEST

The following elements and attributes are useless:

- PurchaseConfirmRQ/ConfirmationData/ConfirmationServiceDataList/ServiceData/TravelInfo/DepartInfo/Name
- PurchaseConfirmRQ/ConfirmationData/ConfirmationServiceDataList/ServiceData/TravelInfo/DepartInfo/DescriptionList
- PurchaseConfirmRQ/ConfirmationData/ConfirmationServiceDataList/ServiceData/TravelInfo/DepartInfo/ImageList
- PurchaseConfirmRQ/ConfirmationData/ConfirmationServiceDataList/ServiceData/TravelInfo/DepartInfo/Contact
- PurchaseConfirmRQ/ConfirmationData/ConfirmationServiceDataList/ServiceData/TravelInfo/DepartInfo/TransferZone
- PurchaseConfirmRQ/ConfirmationData/ConfirmationServiceDataList/ServiceData/TravelInfo/DepartInfo/Country
- PurchaseConfirmRQ/ConfirmationData/ConfirmationServiceDataList/ServiceData/TravelInfo/ArrivalInfo/Name
- PurchaseConfirmRQ/ConfirmationData/ConfirmationServiceDataList/ServiceData/TravelInfo/ArrivalInfo/DescriptionList
- PurchaseConfirmRQ/ConfirmationData/ConfirmationServiceDataList/ServiceData/TravelInfo/ArrivalInfo/ImageList
- PurchaseConfirmRQ/ConfirmationData/ConfirmationServiceDataList/ServiceData/TravelInfo/ArrivalInfo/Contact
- PurchaseConfirmRQ/ConfirmationData/ConfirmationServiceDataList/ServiceData/TravelInfo/ArrivalInfo/TransferZone
- PurchaseConfirmRQ/ConfirmationData/ConfirmationServiceDataList/ServiceData/TravelInfo/ArrivalInfo/Country

III.3.3 RESPONSE BUSINESS RULES

If `PurchaseConfirmRS/Purchase` element is present and `PurchaseConfirmRS/Purchase/ServiceList/Service/@type` is `ServiceTransfer` the following elements and attributes must be provided:

- `PurchaseConfirmRS/Purchase/ServiceList/Service/@SPUI`
- `PurchaseConfirmRS/Purchase/ServiceList/Service/@transferType`
- `PurchaseConfirmRS/Purchase/ServiceList/Service/Status`
- `PurchaseConfirmRS/Purchase/ServiceList/Service/Supplier`
- `PurchaseConfirmRS/Purchase/ServiceList/Service/ContractList`
- `PurchaseConfirmRS/Purchase/ServiceList/Service/PickupLocation/Code`
- `PurchaseConfirmRS/Purchase/ServiceList/Service/DestinationLocation/Code`

In addition, if the transfer service returned has got cancellation fees, the following element must be provided:

- `ServiceAddRS/Purchase/ServiceList/Service/CancellationPolicy`

III.4 PurchaseDetail

In this section you will see the particularities of this operation for Transfer service. If you want to see the general specification for `PurchaseDetail`, please see [HotelBeds XML Interface Specification-v2.4.4](#).

III.4.1 RESPONSE BUSINESS RULES

If `PurchaseDetailRS/Purchase` element is present and `PurchaseDetailRS/Purchase/ServiceList/Service/@type` is `ServiceTransfer` the following elements and attributes must be provided:

- `PurchaseDetailRS/Purchase/ServiceList/Service/@SPUI`
- `PurchaseDetailRS/Purchase/ServiceList/Service/@transferType`
- `PurchaseDetailRS/Purchase/ServiceList/Service/Status`
- `PurchaseDetailRS/Purchase/ServiceList/Service/Supplier`
- `PurchaseDetailRS/Purchase/ServiceList/Service/ContractList`
- `PurchaseDetailRS/Purchase/ServiceList/Service/PickupLocation/Code`
- `PurchaseDetailRS/Purchase/ServiceList/Service/DestinationLocation/Code`

In addition, if the transfer service returned has got cancellation fees, the following element must be provided:

- `ServiceAddRS/Purchase/ServiceList/Service/CancellationPolicy`

III.5 PurchaseList

In this section you will see the particularities of this operation for Transfer service. If you want to see the general specification for `PurchaseList`, please see [HotelBeds XML Interface Specification-v2.4.4](#).

III.5.1 RESPONSE BUSINESS RULES

If `PurchaseListRS/Purchase` element is present and `PurchaseListRS/Purchase/ServiceList/Service/@type` is `ServiceTransfer` the following elements and attributes must be provided:

- `PurchaseListRS/Purchase/ServiceList/Service/@transferType`
- `PurchaseListRS/Purchase/ServiceList/Service/Status`
- `PurchaseListRS/Purchase/ServiceList/Service/ContractList`

- PurchaseListRS/Purchase/ServiceList/Service/PickupLocation/Code
- PurchaseListRS/Purchase/ServiceList/Service/DestinationLocation/Code

III.6 PurchaseCancel

In this section you will see the particularities of this operation for Transfer service. If you want to see the general specification for PurchaseCancel, please see [HotelBeds XML Interface Specification-v2.4.4](#).

III.6.1 RESPONSE BUSINESS RULES

If PurchaseCancelRS/Purchase/ServiceList/Service/@type is ServiceTransfer the following elements and attributes must be provided:

- PurchaseCancelRS/Purchase/ServiceList/Service/@SPUI
- PurchaseCancelRS/Purchase/ServiceList/Service/@transferType
- PurchaseCancelRS/Purchase/ServiceList/Service/Status
- PurchaseCancelRS/Purchase/ServiceList/Service/Supplier
- PurchaseCancelRS/Purchase/ServiceList/Service/ContractList
- PurchaseCancelRS/Purchase/ServiceList/Service/PickupLocation/Code
- PurchaseCancelRS/Purchase/ServiceList/Service/DestinationLocation/Code

IV. FAQ

1. How can I add a complete transfer service to a purchase?

If you send a `TransferValuedAvail` request, providing two `AvailData` elements, one for transfer type `IN` and another one for transfer type `OUT`, you will obtain a response containing a list of transfer services. Each service corresponds with one transfer type.

To add a complete transfer service (`IN` and `OUT`) to a purchase follow the steps below:

- Call the `TransferValuedAvail` operation providing two `AvailData` elements, one for transfer type `IN` and another one for transfer type `OUT`.
- Choose a `TransferValuedAvailRS/ServiceTransfer` element you want to add to the purchase (Ex: choose a transfer with attribute `@transferType = IN`) and call operation `ServiceAdd` providing this service.
- Call again `ServiceAdd` operation providing the other `TransferValuedAvailRS/ServiceTransfer` you want to add to the purchase (Ex: choose a transfer with attribute `@transferType = OUT`) and provide the `ServiceAddRS/@purchaseToken` you received adding the other `ServiceTransfer`.

XML ticket service integration

WHAT IS THE XML TICKET SERVICE INTEGRATION?

Our XML ticket service integration offers you the possibility to book the best entrance tickets in the world's top destinations: culture, sports, leisure, adventure, tours and much more will be available through an integrated XML interface.

The XML tickets feed includes a shopping cart allowing for multiple bookings in different destinations and the possibility to add activities/tickets to your hotel bookings.

A complete back office integration allows for full control of product search queries, modifications and cancellations, making the booking process easier for clients with multi-provider systems.

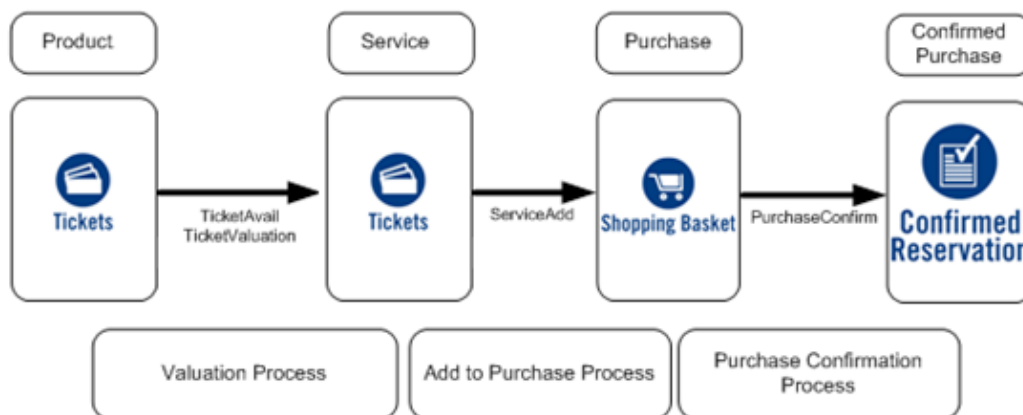
For more information, our extended XML tickets documentation package is constantly updated, offering an in-depth description of all the functionalities, with practical examples and a FAQ section.

Interface development and integration

As part of the interface integration process, we'll provide a test environment with our entire product portfolio loaded where to run the necessary pre-live tests.

Once the XML integration is complete, we'll run a certification process of your application in order to minimise the risk of any platform errors, ensuring that your website, XML feed and system connection is correct and ready to deliver the sales.

WORKFLOW XML FEED



REQUIRED XML REQUESTS

TicketAvailRQ

< TicketAvail > returns an availability list:

```

.....
<AvailableModality code="0#2 DIAS">
<Name>2 Days Pass</Name>
<Contract>
<Name>BCN CARD 2010</Name>
<IncomingOffice code="102"/>
</Contract>
.....
<Type code="P">Only once</Type>
  
```

```

<Mode code="P">Person</Mode>
<OperationDateList>
<OperationDate date="20101028" minimumDuration="1" maximumDuration="2"/>
<OperationDate date="20101029" minimumDuration="1" maximumDuration="1"/>

```

TicketValuationRQ

<TicketValuation> returns the price of a ticket depending on the parameters chosen (length of the service provided and/or on the days requested)

```

.....
<DateFrom date="20101029"/>
<DateTo date="20101029"/>
<Currency code="EUR">Euro</Currency>
<TotalAmount>44.000</TotalAmount>
<AdditionalCostList>
<AdditionalCost type="AG_COMMISSION">
<Price>
<Amount>0.00</Amount>
</Price>
</AdditionalCost>
<AdditionalCost type="COMMISSION_VAT">
<Price>
<Amount>0.00</Amount>
</Price>
</AdditionalCost>
</AdditionalCostList>
<TicketInfo xsi:type="ProductTicket">
<Code>BIKEBCN</Code>
<Name>Bike Tour</Name>
<CompanyCode>E10</CompanyCode>
<TicketClass>T</TicketClass>

```

TicketDetailRQ

TicketDetail returns relevant information related to the ticket.

In the < TicketDetail > request you must use the ticket code and the contract name given in the < TicketAvail > response.

```

<Feature xsi:type="ProductFeatureData" group="8">
<Code>1</Code>
<Name>Is lunch/dinner included on this tour?</Name>
<Description>No.</Description>
</Feature>
</FeatureList>
</TicketFeature>
<TicketFeature xsi:type="ProductFeatureGroup" group="10">
<Code>10</Code>
<Name>Drinks</Name>
<FeatureList>
<Feature xsi:type="ProductFeatureData" group="10">

```

```
<Code>1</Code>  
<Name>Are drinks included?</Name>  
<Description>Yes. In the tasting room you can try the different ranges of cellar Freixenet wine  
and/or champagne</Description>  
</Feature>  
</FeatureList>  
</TicketFeature>
```

TicketClassificationListRQ

< TicketClassificationList > returns a list the different themes of tickets/activities.

< TicketClassificationList > returns a breakdown of all the types of activities/tickets for the destination requested

```
<Classification code="DISCO">Night clubs</Classification>  
<Classification code="RELAX">Relax Take it easy</Classification>  
<Classification code="PARTE">Theme Parks</Classification>  
<Classification code="RESTA">Restaurants</Classification>  
<Classification code="DEPOR">Sport events</Classification>  
<Classification code="PANOR">Scenic tours</Classification>  
<Classification code="OUTAC">Outdoor Activities</Classification>  
<Classification code="CITYP">City pass</Classification>  
<Classification code="CULTU">Cultural</Classification>  
<Classification code="SHDIN">Dinner shows</Classification>  
<Classification code="HELIC">Air Tours</Classification>
```

Other requests:

ServiceAdd/Remove

PurchaseList/Detail/Confirm/Cancel/Flush

PRODUCT UPDATES

The product updates for tickets will need the following tables/.csv files:

- TicketClassification
- TicketClassificationDescription
- TicketDestinations
- Countries(also CountryIDs)
- Destinations(also DestinationsID)

PRICING AND MARKUPS

With our XML ticket feed you'll receive net prices as well as the retail or gate price for each ticket, allowing you to compare rates and establish your markup.


It's important that you always run a < TicketValuation> request, as the XML ticket feed will return several services with variable lengths of stay and/or prices depending on the date.

*Note: the final price of the service is the price you get in the valuation step.

All prices are per number of pax, whether its adult, child or infant. The service price is the total amount for all paxes.

Example price presentation returned via the XML feed,

Sitges and Cava Freixenet - Morning



Want to immerse yourself in the magical world of champagne? Come discover the Cava Freixenet. This guided tour begins with a short audiovisual presentation of the history of a family business. Then ... [more](#)

Type	Date	Days	Gate Price	Saving	Hotelbeds	Final price
General entrance	<div style="border: 2px solid red; padding: 2px;"> Fri - 08/10/2010 Fri - 08/10/2010 Sat - 09/10/2010 Sun - 10/10/2010 </div>	1	50,00 €	30,04%	34,98 €	Final price

NOTE: children older than 7 pay the adult price.

```

<PriceList>
<Price>
<Amount>34.980</Amount>
<Description>ADULT PRICE</Description>
</Price>
<Price>
<Amount>0.000</Amount>
<Description>CHILD PRICE</Description>
</Price>
<Price>
<Amount>0.000</Amount>
<Description>INFANT PRICE</Description>
</Price>
<Price>
<Amount>34.980</Amount>
<Description>SERVICE PRICE</Description>
</Price>
<Price>
<Amount>50.000</Amount>
<Description>TICKET OFFICE PRICE</Description>
</Price>
<Price>
<Amount>0.000</Amount>
<Description>CHILD TICKET OFFICE PRICE</Description>
</Price>
</PriceList>

```

VOUCHER

Below, there's an example of a ticket voucher with the relevant information, as well as some XML examples with the information required.



Client: TEST TEST
 Locator: 102-1762832
 Agency reference: TEST

Hotelbeds contact information: ES - ISLAS BALEARES
 CM. Complejo Mirall Balear, Camí
 Son Fangós, 100 - PALMA DE
 MALLORCA
 e-destinations2@hotelbeds.com

Total pax: 1 From: 22/10/2010 To: 22/10/2010

Service	Address	Opening times
Sitges and Cava Freixenet - Morning	c/ -	-

Ticket type	Adults	Children	Age(s)	Guest Name(s)
General entrance	1	0		Test Test

Contract info:
 Meeting point : Ronda Universitat 5, Barcelona Excursion departure at 08:45 hrs local time (meeting time at 08:15hrs) on Friday, Saturday and Sunday. In case of Emergency, please call the following phone number: + 34 933 17 64 54 or +34,660,409,789.
Remarks:



With guarantee of Hotelbeds reference: 102-1762832-OE1
 Payable through HOTELBEDS SPAIN, S.L.U., CIF UNKNOWN, acting as agent for the service operating company, details of which can be provided upon request.

LOCATOR: <Reference>
 <FileNumber>1762832</FileNumber>
 <IncomingOffice code="102"/>
 </Reference>

AGENCY REFERENCE: <AgencyReference>TEST</AgencyReference>
 DATES FROM/TO: <DateFrom date="20101122"/>
 <DateTo date="20101122"/>

TICKED TYPE: <TicketInfo xsi:type="ProductTicket">
 <Code>GENENTRAN</Code>
 <Name>General Entrance</Name>
 <CompanyCode>E10</CompanyCode>
 <TicketClass>T</TicketClass>
 PAXES/NAMES: <Paxes>
 <AdultCount>1</AdultCount>
 <ChildCount>0</ChildCount>
 <GuestList>
 <Customer type="AD">
 <CustomerId>1</CustomerId>
 <Age>45</Age>
 <Name>Test</Name>
 <LastName>Test</LastName>
 </Customer>
 </GuestList>
 </Paxes>

CONTRACT INFO: <Comment type="CONTRACT">

Client can exchange voucher at Mini Golf Fantasia's ticket office.

Timetable:

Until 26/03/2010 opened from Friday to Wednesday from 10.30 hrs. to 18.00 hrs.

As from 27/03/2010 opened every day from 10.30 hrs. to 22.00 hrs.

</Comment>

PAYABLE THROUGH: <PaymentData>

<PaymentType code="P"/>

-

<Description>

Payment for this pro-forma invoice should be made in full to Hotelbeds S.L.U., Bank: BBVA(Plaza del Olivar, s/n - CP 07002 - Palma de Mallorca - BALEARES) Account:ES73 0182-4899-14-0200712789, SWIFT:BBVAESMMXXX, 7 days prior to clients arrival. Please indicate our reference number when making payment.Thank you for your cooperation.

</Description>

</PaymentData>



Ticket Service XML Interface Specification **developer's guide**

v1.0.2

June 10

Hotelbeds Accommodation & Destination Services
Camí de Son Fangós, 100 - Torre A, 5ª Planta
07007 - Palma de Mallorca
Spain

I.	Document Control	3
II.	Operations.....	4
II.1	TicketAvail	4
II.1.1	Request business rules	4
II.1.2	Request Observations	4
II.1.3	Request Acceptance Test	4
II.1.4	Response business rules	5
II.1.5	Response Observations	5
II.2	TicketClassificationList	6
II.2.1	Request business rules	6
II.2.2	Request Observations	6
II.2.3	Request Acceptance Test	6
II.2.4	Response business rules	6
II.2.5	Response Observations	6
II.3	TicketDetail.....	7
II.3.1	Request business rules	7
II.3.2	Request Observations	7
II.3.3	Request Acceptance Test	7
II.3.4	Response business rules	7
II.3.5	Response Observations	7
II.4	TicketValuation	8
II.4.1	Request business rules	8
II.4.2	Request Observations	8
II.4.3	Request Acceptance Test	8
II.4.4	Response business rules	8
II.4.5	Response Observations	8
III.	Particularities of common operations	10
III.1	ServiceAdd.....	10
III.1.1	Request Business Rules	10
III.1.2	Request Acceptance Test	10
III.1.3	Response Business Rules	10
III.1.4	Response Observations	11
III.2	ServiceRemove	11
III.2.1	Response Business Rules	11
III.3	PurchaseConfirm.....	11
III.3.1	Request Business Rules	11
III.3.2	Response Business Rules	11
III.3.3	Response Observations	12
III.3.4	Request Acceptance Test	12
III.4	PurchaseDetail.....	13
III.4.1	Response Business Rules	13
III.4.2	Response Observations	13
III.5	PurchaseList.....	13
III.5.1	Response Business Rules	13
III.6	PurchaseCancel	13
III.6.1	Response Business Rules	14
IV.	FAQ	15

I. Document Control

1.0.2	02/06/10	CancelProtection section removed	4
1.0.1	11/08/08	Modified FAQ <i>How is calculated the final price of a ticket service?</i>	4
1.0.0	14/04/08	New Ticket documentation file	4

II. Operations

II.1 TicketAvail

Get a list of available ticket services according to the requested parameters.

TicketAvail	
Details:	xsddocs/TicketAvailRQ.html xsddocs/TicketAvailRS.html
Schema:	xml/TicketAvailRQ.xsd xml/TicketAvailRS.xsd
Example:	xml/TicketAvailRQ.xml xml/TicketAvailRS.xml

II.1.1 REQUEST BUSINESS RULES

Language	Mandatory
Destination/@type	SIMPLE value
DateFrom/@date	At least today
DateTo/@date	Later than DateFrom/@date

II.1.2 REQUEST OBSERVATIONS

If TicketAvailRQ/ServiceOccupancy/GuestList/Customer/@type attribute value is CH, element TicketAvailRQ/ServiceOccupancy/GuestList/Customer/Age will be mandatory.

If TicketAvailRQ/TicketZone element is present then TicketAvailRQ/TicketZone/Code will be mandatory.

II.1.3 REQUEST ACCEPTANCE TEST

The following elements and attributes are useless:

- TicketAvailRQ/ExtraParamList
- TicketAvailRQ/ServiceOccupancy/GuestList/Customer/CustomerId
- TicketAvailRQ/ServiceOccupancy/GuestList/Customer/Name
- TicketAvailRQ/ServiceOccupancy/GuestList/Customer/LastName
- TicketAvailRQ/ServiceOccupancy/GuestList/Customer/AdditionalInfo
- TicketAvailRQ/ServiceOccupancy/GuestList/Customer/Document
- TicketAvailRQ/ServiceOccupancy/GuestList/Customer/BirthDate
- TicketAvailRQ/ServiceOccupancy/GuestList/Customer/CountryCode
- TicketAvailRQ/Destination/Name
- TicketAvailRQ/Destination/ZoneList
- TicketAvailRQ/DateFrom/@time
- TicketAvailRQ/DateTo/@time
- TicketAvailRQ/TicketZone/Name
- TicketAvailRQ/TicketZone/DescriptionList
- TicketAvailRQ/TicketZone/ImageList
- TicketAvailRQ/TicketZone/Contact
- TicketAvailRQ/Contract/IncomingOffice/Description
- TicketAvailRQ/Contract/IncomingOffice/ContactInfo
- TicketAvailRQ/Contract/IncomingOffice/FiscalNumber
- TicketAvailRQ/Contract/Sequence
- TicketAvailRQ/Contract/Classification

- TicketAvailRQ/Contract/CommentList

Furthermore:

- TicketAvailRQ/ServiceOccupancy/GuestList/Customer/Age is useless for adult customers.

II.1.4 RESPONSE BUSINESS RULES

One of TicketAvailRS/ServiceTicket or TicketAvailRS/ErrorList element must be present in the response unless the TicketAvailRS/@totalItems attribute is 0.

If TicketAvailRS/ServiceTicket element is present the following elements and attributes must be always present:

- TicketAvailRS/PaginationData
- TicketAvailRS/ServiceTicket/@availToken
- TicketAvailRS/ServiceTicket/AvailableModality/PriceList
- TicketAvailRS/ServiceTicket/AvailableModality/OperationDateList
- TicketAvailRS/ServiceTicket/AvailableModality/ChildAge
- TicketAvailRS/ServiceTicket/AvailableModality/ChildAge/@ageFrom
- TicketAvailRS/ServiceTicket/AvailableModality/ChildAge/@ageTo

If TicketAvailRS/ServiceTicket/Paxes/GuestList/Customer/@type attribute value is CH, element TicketAvailRS/ServiceTicket/Paxes/GuestList/Customer/Age will be mandatory.

II.1.5 RESPONSE OBSERVATIONS

If no TicketAvailRS/ServiceTicket element is present and /@totalItems attribute is 0, no ticket services were found matching the search criteria.

If one or more TicketAvailRS/ServiceTicket elements are present:

- /@totalItems attribute contains the total number of ticket services returned in the response.

II.2 TicketClassificationList

Get a list of available classifications for ticket services.

TicketClassificationList

Details:	xsddocs/TicketClassificationListRQ.html	xsddocs/TicketClassificationListRS.html
Schema:	xml/TicketClassificationListRQ.xsd	xml/TicketClassificationListRS.xsd
Example:	xml/TicketClassificationListRQ.xml	xml/TicketClassificationListRS.xml

II.2.1 REQUEST BUSINESS RULES

Language	Mandatory
Destination/@type	SIMPLE value

II.2.2 REQUEST OBSERVATIONS

II.2.3 REQUEST ACCEPTANCE TEST

The following elements and attributes are useless:

- TicketClassificationListRQ/ExtraParamList
- TicketClassificationListRQ/Destination/Name
- TicketClassificationListRQ/Destination/ZoneList

II.2.4 RESPONSE BUSINESS RULES

One of TicketClassificationListRS/Classification or TicketClassificationListRS/ErrorMessage element must be present in the response unless the @totalItems attribute is 0.

II.2.5 RESPONSE OBSERVATIONS

If no TicketClassificationListRS/Classification element is present and /@totalItems attribute is 0, no ticket classifications were found matching the search criteria.

If one or more TicketClassificationListRS/Classification elements are present:

- /@totalItems attribute contains the total number of classifications returned in the response.

II.3 TicketDetail

Get detailed information for the specified ticket product.

TicketDetail	
Details:	xsddocs/TicketDetailRQ.html xsddocs/TicketDetailRS.html
Schema:	xml/TicketDetailRQ.xsd xml/TicketDetailRS.xsd
Example:	xml/TicketDetailRQ.xml xml/TicketDetailRS.xml

II.3.1 REQUEST BUSINESS RULES

Language	Mandatory
----------	-----------

II.3.2 REQUEST OBSERVATIONS

Don't provide at element TicketDetailRQ/ModalityCode the same value as TicketAvailRS/ServiceTicket/AvailableModality/@code. Two first characters from modality received at TicketAvailRS must be deleted. Example:

TicketAvailRS/ServiceTicket/AvailableModality/@code value is "0#ARENAL"
TicketDetailRQ/ModalityCode must be "ARENAL"

II.3.3 REQUEST ACCEPTANCE TEST

The following elements and attributes are useless:

- TicketDetailRQ/ExtraParamList
- TicketDetailRQ/Contract/IncomingOffice/Description
- TicketDetailRQ/Contract/IncomingOffice/ContactInfo
- TicketDetailRQ/Contract/IncomingOffice/FiscalNumber
- TicketDetailRQ/Contract/Sequence
- TicketDetailRQ/Contract/Classification
- TicketDetailRQ/Contract/CommentList

II.3.4 RESPONSE BUSINESS RULES

If TicketDetailRS/Ticket element is present the following elements and attributes must be always present:

- TicketDetailRS/Ticket/Name
- TicketDetailRS/Ticket/DescriptionList
- TicketDetailRS/Ticket/Destination/Name
- TicketDetailRS/Ticket/TicketFeature/FeatureList/Feature/Description

II.3.5 RESPONSE OBSERVATIONS

II.4 TicketValuation

Get a final valuation of a ticket service for the desired dates and occupancy. This service comes from a previous ticket availability.

TicketValuation

Details:	xsddocs/TicketValuationRQ.html	xsddocs/TicketValuationRS.html
Schema:	xml/TicketValuationRQ.xsd	xml/TicketValuationRS.xsd
Example:	xml/TicketValuationRQ.xml	xml/TicketValuationRS.xml

II.4.1 REQUEST BUSINESS RULES

Language	Mandatory
ServiceOccupancy/GuestList	Mandatory

II.4.2 REQUEST OBSERVATIONS

If TicketValuationRQ/ServiceOccupancy/GuestList/Customer/@type attribute value is CH, element TicketValuationRQ/ServiceOccupancy/GuestList/Customer/Age will be mandatory.

II.4.3 REQUEST ACCEPTANCE TEST

The following elements and attributes are useless:

- TicketValuationRQ/ExtraParamList
- TicketValuationRQ/ServiceOccupancy/GuestList/Customer/CustomerId
- TicketValuationRQ/ServiceOccupancy/GuestList/Customer/Name
- TicketValuationRQ/ServiceOccupancy/GuestList/Customer/LastName
- TicketValuationRQ/ServiceOccupancy/GuestList/Customer/AdditionalInfo
- TicketValuationRQ/ServiceOccupancy/GuestList/Customer/Document
- TicketValuationRQ/ServiceOccupancy/GuestList/Customer/BirthDate
- TicketValuationRQ/ServiceOccupancy/GuestList/Customer/CountryCode

Furthemore:

- TicketValuationRQ/ServiceOccupancy/GuestList/Customer/Age is useless for adult customers.

II.4.4 RESPONSE BUSINESS RULES

If TicketValuationRS/ServiceTicket element is present the following elements and attributes must be always present:

- TicketValuationRS/ServiceTicket/Paxes/AdultCount
- TicketValuationRS/ServiceTicket/Paxes/ChildCount
- TicketValuationRS/ServiceTicket/Paxes/GuestList

II.4.5 RESPONSE OBSERVATIONS

If TicketValuationRS/ServiceTicket/Paxes/GuestList/Customer/@type attribute value is CH, element TicketValuationRQ/ServiceOccupancy/GuestList/Customer/Age will be mandatory.

If service has got contract observations, you will receive element TicketValuationRS/ServiceTicket/ServiceDetailList. This list contains questions about service that you must response.

Question: ServiceDetailList/ServiceDetail/Name

Response: ServiceDetailList/ServiceDetail/Description

You must provide the response to each question at PurchaseConfirm operation. You will receive again this information at ServiceAdd response.

III. Particularities of common operations

III.1 ServiceAdd

In this section you will see the particularities of this operation for Ticket service. If you want to see the general specification for ServiceAdd, please see [HotelBeds XML Interface Specification-v2.4.4](#).

III.1.1 REQUEST BUSINESS RULES

Service/DateTo	Mandatory
Service/TicketInfo/Code	Mandatory
Service/Paxes/GuestList/ Customer/ Age	Mandatory if Service/Paxes/GuestList/ Customer/@type attribute is CH.
Service/AvailableModality	Only the first element will be taken in consideration. If you want to provide more than one modality a new service must be added for each new modality.

III.1.2 REQUEST ACCEPTANCE TEST

The following elements and attributes are useless:

- ServiceAddRQ/Service/DateTo/@time
- ServiceAddRQ/Service/TicketInfo/Name
- ServiceAddRQ/Service/TicketInfo/DescriptionList
- ServiceAddRQ/Service/TicketInfo/ImageList
- ServiceAddRQ/Service/TicketInfo/Contact
- ServiceAddRQ/Service/TicketInfo/TicketZone
- ServiceAddRQ/Service/TicketInfo/Classification
- Only first ServiceAddRQ/Service/AvailableModality is taken in consideration
- ServiceAddRQ/Service/AvailableModality/PriceList
- ServiceAddRQ/Service/AvailableModality/Type
- ServiceAddRQ/Service/AvailableModality/Mode
- ServiceAddRQ/Service/AvailableModality/OperationDateList
- ServiceAddRQ/Service/AvailableModality/ChildAge
- ServiceAddRQ/Service/Paxes/GuestList/ Customer/ CustomerId
- ServiceAddRQ/Service/Paxes/GuestList/ Customer/ Name
- ServiceAddRQ/Service/Paxes/GuestList/ Customer/ LastName
- ServiceAddRQ/Service/Paxes/GuestList/ Customer/ AdditionalInfo
- ServiceAddRQ/Service/Paxes/GuestList/ Customer/ Document
- ServiceAddRQ/Service/Paxes/GuestList/ Customer/ BirthDate
- ServiceAddRQ/Service/Paxes/GuestList/ Customer/ CountryCode
- ServiceAddRQ/Service/Paxes/GuestList/ Customer/ Age useless for adult customers
- ServiceAddRQ/Service/CancellationPolicyList

III.1.3 RESPONSE BUSINESS RULES

If ServiceAddRS/Purchase/ServiceList/Service/@type is ServiceTicket the following elements and attributes must be provided:

- ServiceAddRS/Purchase/ServiceList/Service/@SPUI
- ServiceAddRS/Purchase/ServiceList/Service/Status
- ServiceAddRS/Purchase/ServiceList/Service/Supplier
- ServiceAddRS/Purchase/ServiceList/Service/TicketInfo/Code
- ServiceAddRS/Purchase/ServiceList/Service/TicketInfo/Name
- ServiceAddRS/Purchase/ServiceList/Service/TicketInfo/Destination/Name

III.1.4 RESPONSE OBSERVATIONS

If service has got contract observations, you will receive element `ServiceAddRS/Purchase/ServiceList/Service/ServiceDetailList`. This list contains questions about service that you must response.

Question: `ServiceDetailList/ServiceDetail/Name`

Response: `ServiceDetailList/ServiceDetail/Description`

You must provide the response to each question at `PurchaseConfirm` operation.

III.2 ServiceRemove

In this section you will see the particularities of this operation for Ticket service. If you want to see the general specification for `ServiceRemove`, please see [HotelBeds XML Interface Specification-v2.4.4](#).

III.2.1 RESPONSE BUSINESS RULES

If `ServiceRemoveRS/Purchase` element is present and `ServiceRemoveRS/Purchase/ServiceList/Service/@type` is `ServiceTicket` the following elements and attributes must be provided:

- `ServiceRemoveRS/Purchase/ServiceList/Service/@SPUI`
- `ServiceRemoveRS/Purchase/ServiceList/Service/Status`
- `ServiceRemoveRS/Purchase/ServiceList/Service/Supplier`
- `ServiceRemoveRS/Purchase/ServiceList/Service/TicketInfo/Code`
- `ServiceRemoveRS/Purchase/ServiceList/Service/TicketInfo/Name`
- `ServiceRemoveRS/Purchase/ServiceList/Service/TicketInfo/Destination/Name`

III.3 PurchaseConfirm

In this section you will see the particularities of this operation for Ticket service. If you want to see the general specification for `PurchaseConfirm`, please see [HotelBeds XML Interface Specification-v2.4.4](#).

III.3.1 REQUEST BUSINESS RULES

If `ServiceAddRS/Purchase/ServiceList/Service/ServiceDetailList` element was present, the following elements and attributes must be provided:

- `PurchaseConfirmRQ/Purchase/ServiceList/Service/ServiceDetailList/ServiceDetail/@code`
- `PurchaseConfirmRQ/Purchase/ServiceList/Service/ServiceDetailList/ServiceDetail/Name`
- `PurchaseConfirmRQ/Purchase/ServiceList/Service/ServiceDetailList/ServiceDetail/Description`

The element `Description` must contains the response to question about ticket service (element `Name`).

III.3.2 RESPONSE BUSINESS RULES

If `PurchaseConfirmRS/Purchase` element is present and `PurchaseConfirmRS/Purchase/ServiceList/Service/@type` is `ServiceTicket` the following elements and attributes must be provided:

- `PurchaseConfirmRS/Purchase/ServiceList/Service/@SPUI`

- PurchaseConfirmRS/Purchase/ServiceList/Service/Status
- PurchaseConfirmRS/Purchase/ServiceList/Service/Supplier
- PurchaseConfirmRS/Purchase/ServiceList/Service/TicketInfo/Code
- PurchaseConfirmRS/Purchase/ServiceList/Service/TicketInfo/Name
- PurchaseConfirmRS/Purchase/ServiceList/Service/TicketInfo/Destination/Name

III.3.3 RESPONSE OBSERVATIONS

If PurchaseConfirmRS/Purchase/ServiceList/Service/@type is ServiceTicket and service has got contract observations, the following elements and attributes must be provided:

- PurchaseConfirmRS/Purchase/ServiceList/Service/ServiceDetailList/ServiceDetail/@code
- PurchaseConfirmRS/Purchase/ServiceList/Service/ServiceDetailList/ServiceDetail/Name
- PurchaseConfirmRS/Purchase/ServiceList/Service/ServiceDetailList/ServiceDetail/Description

III.3.4 REQUEST ACCEPTANCE TEST

The following elements and attributes are useless:

- PurchaseConfirmRQ/ConfirmationData/Confirmation/ServiceDataList/ServiceData/CommentList

III.4 PurchaseDetail

In this section you will see the particularities of this operation for Ticket service. If you want to see the general specification for PurchaseDetail, please see [HotelBeds XML Interface Specification-v2.4.4](#).

III.4.1 RESPONSE BUSINESS RULES

If PurchaseDetailRS/Purchase element is present and PurchaseDetailRS/Purchase/ServiceList/Service/@type is ServiceTicket the following elements and attributes must be provided:

- PurchaseDetailRS/Purchase/ServiceList/Service/@SPUI
- PurchaseDetailRS/Purchase/ServiceList/Service/Status
- PurchaseDetailRS/Purchase/ServiceList/Service/Supplier
- PurchaseDetailRS/Purchase/ServiceList/Service/TicketInfo/Code
- PurchaseDetailRS/Purchase/ServiceList/Service/TicketInfo/Name
- PurchaseDetailRS/Purchase/ServiceList/Service/TicketInfo/Destination/Name

III.4.2 RESPONSE OBSERVATIONS

If PurchaseDetailRS/Purchase/ServiceList/Service/@type is ServiceTicket and service has got contract observations, the following elements and attributes must be provided:

- PurchaseDetailRS/Purchase/ServiceList/Service/ServiceDetailList/ServiceDetail/@code
- PurchaseDetailRS/Purchase/ServiceList/Service/ServiceDetailList/ServiceDetail/Name
- PurchaseDetailRS/Purchase/ServiceList/Service/ServiceDetailList/ServiceDetail/Description

III.5 PurchaseList

In this section you will see the particularities of this operation for Ticket service. If you want to see the general specification for PurchaseList, please see [HotelBeds XML Interface Specification-v2.4.4](#).

III.5.1 RESPONSE BUSINESS RULES

If PurchaseListRS/Purchase element is present and PurchaseListRS/Purchase/ServiceList/Service/@type is ServiceTicket the following elements and attributes must be provided:

- PurchaseListRS/Purchase/ServiceList/Service/Status
- PurchaseListRS/Purchase/ServiceList/Service/TicketInfo/Code
- PurchaseListRS/Purchase/ServiceList/Service/TicketInfo/Name
- PurchaseListRS/Purchase/ServiceList/Service/TicketInfo/Destination/Name

III.6 PurchaseCancel

In this section you will see the particularities of this operation for Ticket service. If you want to see the general specification for PurchaseCancel, please see [HotelBeds XML Interface Specification-v2.4.4](#).

III.6.1 RESPONSE BUSINESS RULES

If `PurchaseCancelRS/Purchase` element is present and `PurchaseCancelRS/Purchase/ServiceList/Service/@type` is `ServiceTicket` the following elements and attributes must be provided:

- `PurchaseCancelRS/Purchase/ServiceList/Service/@SPUI`
- `PurchaseCancelRS/Purchase/ServiceList/Service/Status`
- `PurchaseCancelRS/Purchase/ServiceList/Service/Supplier`
- `PurchaseCancelRS/Purchase/ServiceList/Service/TicketInfo/Code`
- `PurchaseCancelRS/Purchase/ServiceList/Service/TicketInfo/Name`
- `PurchaseCancelRS/Purchase/ServiceList/Service/TicketInfo/Destination/Name`

IV. FAQ

1. How is calculated the final price of a ticket service?

Operation TicketAvail returns a list of ticket services availables. For each of these tickets, you will obtain an AvailableModality element with the following information about the price:

- Type of modality price.

TicketAvailRS/ServiceTicket/AvailableModality/Type possible values are:

-

- Modality price.

TicketAvailRS/ServiceTicket/AvailableModality/Mode possible values are:

-

- List of prices.

TicketAvailRS/ServiceTicket/AvailableModality/PriceList/Price/Description possible values are:

- ADULT PRICE: Adult ticket price.
- CHILD PRICE: Child ticket price.
- INFANT PRICE: Infant ticket price.
- SERVICE PRICE: Total price of ticket service (including all pax).
- TICKET OFFICE PRICE: Adult ticket office price.
- CHILD TICKET OFFICE PRICE: Child ticket office price.

Important note: Price/Amount shows the price of the first day you request. So the Price/Amount can change if you reserve the ticket for another days.

- Child age.

TicketAvailRS/ServiceTicket/AvailableModality/ChildAge. What's the meaning of this element?

- If TicketAvailRQ/ServiceOccupancy/GuestList/Customer/Age < ChildAge/ageFrom then Customer will pay as an infant.
- If ChildAge/ageFrom <= TicketAvailRQ/ServiceOccupancy/GuestList/Customer/Age <= ChildAge/ageTo then Customer will pay as a child.
- If TicketAvailRQ/ServiceOccupancy/GuestList/Customer/Age > ChildAge/ageTo then Customer will pay as an adult.

Using this information, how is calculated the final price of a ticket service?

Prices in modality P (Person)

- Type 'D' (days):

$(\text{TicketAvailRQ} / \text{ServiceOccupancy} / \text{AdultCount} * \text{ADULT PRICE} + \text{TicketAvailRQ} / \text{ServiceOccupancy} / \text{ChildCount} * \text{CHILD PRICE}(\text{case child}) / \text{INFANT PRICE}(\text{case infant}))$

*Number of Days.

- Type 'N' (nights):

$(\text{TicketAvailRQ} / \text{ServiceOccupancy} / \text{AdultCount} * \text{ADULT PRICE} + \text{TicketAvailRQ} / \text{ServiceOccupancy} / \text{ChildCount} * \text{CHILD PRICE}(\text{case child}) / \text{INFANT PRICE}(\text{case infant}))$

*Number of Nights (days - 1).

- Type 'P' (only once):

$(\text{TicketAvailRQ} / \text{ServiceOccupancy} / \text{AdultCount} * \text{ADULT PRICE} + \text{TicketAvailRQ} / \text{ServiceOccupancy} / \text{ChildCount} * \text{CHILD PRICE}(\text{case child}) / \text{INFANT PRICE}(\text{case infant}))$

Prices in modality S (Unit)

- Type 'D' (days): (Price) * Number of Days.
- Type 'N' (nights): (Price) * Number of Nights(days - 1).
- Type 'P' (only once): (Price).



HotelBeds XML Interface Specification **developer's guide**

v2.4.4
August 10

Hotelbeds Accommodation & Destination Services
Camí de Son Fangós, 100 - Torre A, 5ª Planta
07007 - Palma de Mallorca
Spain

I.	Document Control	5
II.	Introduction.....	9
II.1	Documentation organization	9
II.2	XML Services integration documentation organization.....	9
II.3	Version Policy	10
II.3.1	Document version.....	10
II.3.2	Interface XML version	10
II.4	Setup Process.....	10
II.4.1	Access to test environment	10
II.4.2	Access to live environment.....	11
III.	Technical Information.....	12
III.1	Compression in responses	13
III.2	SOAP messages	14
III.2.1	SOAP Message Request Example.....	14
III.2.2	HTTP SOAP Request Example	15
III.2.3	SOAP Message Response Example.....	16
III.2.4	HTTP SOAP Message Response Example.....	16
IV.	Basic Concepts	17
IV.1	Product	17
IV.2	Service.....	17
IV.3	Purchase.....	17
V.	Acceptance Test	18
V.1	What is the Acceptance Test (AT)?	18
V.2	Client requirements.....	18
V.3	Acceptance test considerations	18
V.3.1	Data consistency	18
V.3.2	Request data verification	18
V.3.3	Net traffic.....	18
V.4	Steps of certification process	19
VI.	Best Practices	22
VI.1	SessionId.....	22
VI.2	Search by geographical criteria	22
VI.3	What you should never do.....	22
VI.4	Workflow	22
VII.	Workflows	23
VII.1	Service confirmation workflow	23
VII.1.1	Description	23
VII.2	Purchase modification (add or remove services) workflow	24
VII.2.1	Description	24
VII.3	Purchase cancellation workflow	25
VII.3.1	Description	25
VIII.	Common Operations	26

VIII.1	ServiceAdd.....	26
VIII.1.1	Request business rules	27
VIII.1.2	Request Observations	27
VIII.1.3	Request Acceptance Test	27
VIII.1.4	Response business rules	27
VIII.1.5	Response Observations	27
VIII.2	ServiceRemove.....	28
VIII.2.1	Request business rules	28
VIII.2.2	Request Observations	28
VIII.2.3	Request Acceptance Test	28
VIII.2.4	Response business rules	28
VIII.2.5	Response Observations	28
VIII.3	PurchaseFlush	29
VIII.3.1	Request business rules	29
VIII.3.2	Request Observations	29
VIII.3.3	Request Acceptance Test	29
VIII.3.4	Response business rules	29
VIII.3.5	Response Observations	29
VIII.4	PurchaseConfirm.....	30
VIII.4.1	Request business rules	30
VIII.4.2	Request Observations	30
VIII.4.3	Request Acceptance Test	30
VIII.4.4	Response business rules	30
VIII.4.5	Response Observations	30
VIII.5	PurchaseDetail.....	31
VIII.5.1	Request business rules	31
VIII.5.2	Request Observations	31
VIII.5.3	Request Acceptance Test	31
VIII.5.4	Response business rules	31
VIII.5.5	Response Observations	31
VIII.6	PurchaseList.....	32
VIII.6.1	Request business rules	32
VIII.6.2	Request Observations	32
VIII.6.3	Request Acceptance Test	32
VIII.6.4	Response business rules	32
VIII.6.5	Response Observations	32
VIII.7	PurchaseCancel	33
VIII.7.1	Request business rules	33
VIII.7.2	Request Observations	33
VIII.7.3	Request Acceptance Test	33
VIII.7.4	Response business rules	33
VIII.7.5	Response Observations	34
VIII.8	IncomingOfficeList	35
VIII.8.1	Request business rules	35
VIII.8.2	Request Observations	35
VIII.8.3	Request Acceptance Test	35
VIII.8.4	Response business rules	35
VIII.8.5	Response Observations	35
IX.	FAQ	36
X.	Troubleshooting	40
X.1	I get the HTTP Status code 500 with the following error message: "IndexOutOfBoundsException: Index: 0, Size: 0" error.	40
X.2	I get "Envelope is not a valid tag name." error.	41
X.3	I get the HTTP Status code 500 with the following error message: "Unknown operation"	41

X.4	I get the HTTP Status code 500 with the following error message: "Method 'GET' not allowed, please use 'POST' method"	41
XI.	Conventions.....	42
XI.1	Language Codes	42
XI.2	Error Codes.....	42
XI.2.1	Application errors	42
XI.2.2	Bussines logic errors	42
XI.2.3	Communication errors	42

I. Document Control

2.4.4	02/08/10	<ul style="list-style-type: none"> - Hotel Service document updated to 1.2.0 - New elements (HotelbedsCommonTypes.xsd): <ol style="list-style-type: none"> 1. Service/DirectPayment 2. Service/AcceptedCardTypes 3. Service/NetPrice 4. Service/Commission 5. SellingPrice 6. Service/SellingPrice 7. ServicePrice/NetPrice 8. ServicePrice/Commission 9. ServicePrice/SellingPrice 10. ExtraOverride 11. ServicePrice/ExtraOverride 12. AdditionalCostList/Currency 13. AdditionalCostList/PvpEquivalent 14. PaymentCard 15. ContactData 16. PaymentData/PaymentCard 17. PaymentData/ContactData 18. ConfirmationServiceData/PaymentData 19. ServiceHotel/PackageRate - Added type StringLength0to25 on HotelbedsSimpleTypes.xsd - New element (ServiceAddRQ.xsd): ServiceAddRQ/ShowNetPrice - Updated HotelbedsCommonTypes.html from xsddocs. - Updated HotelValuedAvailRQ.html from xsddocs. - Updated ServiceAddRQ.html from xsddocs. - Updated diagrams from xsddocs - Modified section II.3 Version Policy 	8
2.4.3	02/08/10	<ul style="list-style-type: none"> - Car Service document updated to 1.0.1 - Hotel Service document updated to 1.0.3 - Transfer Service document updated to 1.0.1 - Ticket Service document updated to 1.0.2 - Product Update document updated to Product Update Nex Gen v.1.3.2 - Modified "Clients requirements" from Acceptance Test section. - Modified "Steps of certification process" section. - Added new section "Workflow" on Best Practices. - Deleted "Compression" section from Best Practices. - Deleted "Product content & product images locally stored on clients" section from Best Practices. - New optional elements (HotelbedsCommonTypes.xsd): <ol style="list-style-type: none"> 1. ServiceTicketModality/ContentSequence added. 2. ServiceTransfer/CancellationPolicy added. 3. CoreRequest/@version added. 4. ProductTicket/TicketClass added. 5. ServiceCarSpecialEquip/Amount added. 6. ServiceCarSpecialEquip/Currency added. 7. ServiceCar/FlightNumber added. - Modified ProductZone/HotelAddress and ConfirmationServiceDataTransfer/HotelAddress type definition. - Updated ServiceAddRS_Transfer.xml example from /xml - Updated HotelbedsCommonTypes.html from xsddocs - Updated HotelbedsSimpleTypes.html from xsddocs - Updated diagrams from xsddocs - Modified FAQ (16). - Modified section Request Observations from PurchaseConfirm operation. 	8

		<ul style="list-style-type: none"> - Sections CancelProtectionAdd and CancelProtectionRemove deleted. - Deleted CancelProtectionAdd and CancelProtectionRemove from /xsd docs documentation files and /xml. - Deleted FAQ (8) What is a cancel protection? - Deleted FAQ (9) Cancel protection and service cancellation fees. - Gzip compression is mandatory. Modified section III.1. 	
2.4.2	21/08/08	<ul style="list-style-type: none"> - Ticket Service document updated to 1.0.1 - Hotel Service document updated to 1.0.2 - Product Update document updated to 1.3.1 - Added new FAQ (18). - Fixed section Request Observations from PurchaseConfirm operation. - Fixed complexType_Customer.png from xsddocs/diagrams. 	10
2.4.1	26/06/08	<ul style="list-style-type: none"> - Deleted removed operations from /xsd docs documentation files. - Hotel Service document updated to 1.0.1 	10
2.4.0	04/06/08	<ul style="list-style-type: none"> - Product Update document updated to 1.3.0 - Modified section Request observations from TicketDetail operation. - Modified section Request observations from HotelValuedAvail operation. - Modified section Response Observations from TicketValuation operation. - Modified section Response Observations from ServiceAdd operation (only ticket services). - Modified section Request Business rules from PurchaseConfirm operation (only ticket services). - Modified section Response Observations from PurchaseDetail operation (only ticket services). - Modified section Response Observations from PurchaseConfirm operation (only ticket services). - Fixed section Business Rules from TransferValuedAvail operation. - Added new FAQ (17). - Added new FAQ (16). - Fixed ServiceAddRQ_Transfer_IN.xml and ServiceAddRQ_Transfer_OUT.xml examples. - Modified section Request Business Rules at HotelValuedAvail operation. - Deleted HotelRoomTypeGroupList operation. - Deleted HotelCategoryGroupList operation. - Deleted HotelBoardGroupList operation. - Deleted DestinationGroupList operation. - Deleted infants from specification. <p>HotelbedsCommonTypes.xsd: ProductHotel/InfantAge deleted. ServiceOccupancy/InfantCount deleted. ServiceTicketModality/InfantAge deleted.</p> <p>HotelbedsSimpleTypes.xsd: HotelbedsCustomerType type IN deleted. Xml examples and documentation updated without infants.</p> <ul style="list-style-type: none"> - New simple type: HotelbedsSimpleTypes.xsd: StringLength1to10. - New optional elements: HotelbedsCommonTypes.xsd: Purchase/CommentList added. ConfirmationPurchaseData/CommentList added. - Modified section Request Business Rules at PurchaseConfirm operation. - Modified section Request Important Notices at ServiceAdd and PurchaseConfirm. Added reference to HotelPostalCode. - New optional elements: HotelbedsCommonTypes.xsd: ProductZone/HotelPostalCode added. ConfirmationServiceDataTransfer/HotelPostalCode 	10

		<ul style="list-style-type: none"> added. - Deleted InsuranceValuedAvail operation. - Added note about encoding in section 3 Technical Information. 	
2.3.0	13/12/07	<ul style="list-style-type: none"> - Product Update document updated to 1.2.0 - Acceptance Test document updated to 1.1.0. - Modified Section Workflows. Duplicated serviceAdd request to transfer services, one to type IN and another one to type OUT. - Added new FAQ: "How can I add a complete transfer service to a purchase?". - Added new xml examples ServiceAddRQ_Transfer_IN and ServiceAddRQ_Transfer_OUT. - Added new FAQ about the final price of ticket services. - Modified attribute 'PaginationRequestData/@itemsPerPage' annotation. - Modified section Technical Information. Added note about HTTP parameter. - Modified section 6.1. Fixed IncomingOfficeList operation name. - Fixed IncomingOfficeList request data diagram. - Modified section Request Important Notices at CountryList, CarCountryList, HotelCountryList, TicketCountryList and TransferCountryList operations. - Added ServiceAdd xml examples for Ticket and Transfer services. - Modified section 2.1 and 3. Added note about wsdl files. - Modified section Request Important Notices at ServiceAdd and PurchaseConfirm operations. Described the way to reserve transfer services when transfer origin or transfer destination is a zone. -New optional element: HotelbedsCommonTypes.xsd: ProductZone/HotelName added. ProductZone/HotelAddress added. ConfirmationServiceDataTransfer/HotelName added. ConfirmationServiceDataTransfer/HotelAddress added. - Section 8.20. Added note about hotel contract. - Modified section Request Important Notices at ServiceAdd and PurchaseConfirm operations. Described the way to indicate transfer service travel information. - Element length change: HotelbedsCommonTypes.xsd: Product/Code is now up to 15 characters. Zone/Code is now up to 15 characters. Reference/FileNumber is now up to 267 characters. Zone content is now up to 255 characters. -New optional element: HotelbedsCommonTypes.xsd: Purchase/CreationUser added. -Section 8.19 Fixed. -Section 3.1 Fixed the endpoint URLs. 	33
2.2.0	07/07	<ul style="list-style-type: none"> -Request business rules for TransferValuedAvail fixed. -PurchaseConfirm response important notices fixed. -Namespace error fixed in WDSL. -Request business rules for CarValuedAvail fixed. -New optional attribute: HotelbedsCommonTypes.xsd: Zone/@serviceType. -Element length change: HotelbedsCommonTypes.xsd: AgencyIdentification/Code is now up to 99999999. -Product Update document updated to 1.1.0 -Request important notices for HotelValuedAvail fixed. 	33

2.1.0	23/05/07	<ul style="list-style-type: none"> -General text mistakes fixed. -ServiceAdd: Request important notices fixed. -PurchaseConfirm: Request important notices fixed. -Troubleshooting section added. -Error codes added. -Live URL fixed. -Acceptance Test document updated to 1.0.1. -Product Update document updated to 1.0.2. -Added a list of temporal hotel issues to ProductHotel complexType. -New element: HotelbedsCommonTypes.xsd PaymentData -New element: HotelbedsCommonTypes.xsd Purchase/PaymentData -Element removed: HotelbedsCommonTypes.xsd Purchase/PaymentType -New element: Added two new elements to ServiceTicketModality. -New complex type: HotelbedsCommonTypes.xsd ServiceTypeModalityType -New complex type: HotelbedsCommonTypes.xsd ServiceTypeModalityMode 	33
2.0.0	01/04/07	First release of the XML on-line reservation interface system.	33

II. Introduction

II.1 Documentation organization

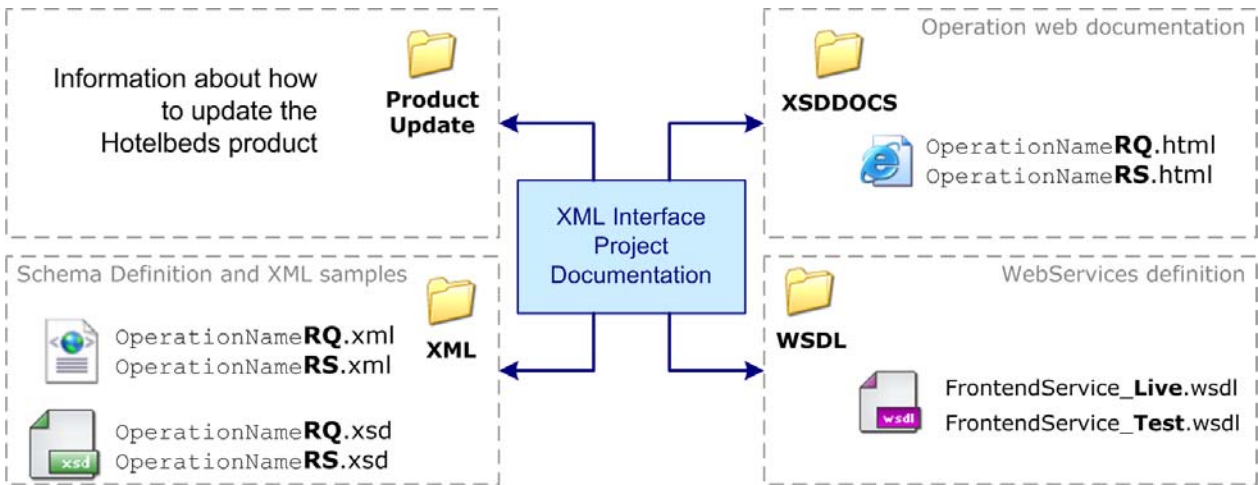


Figure 1 - Folder structure

II.2 XML Services integration documentation organization

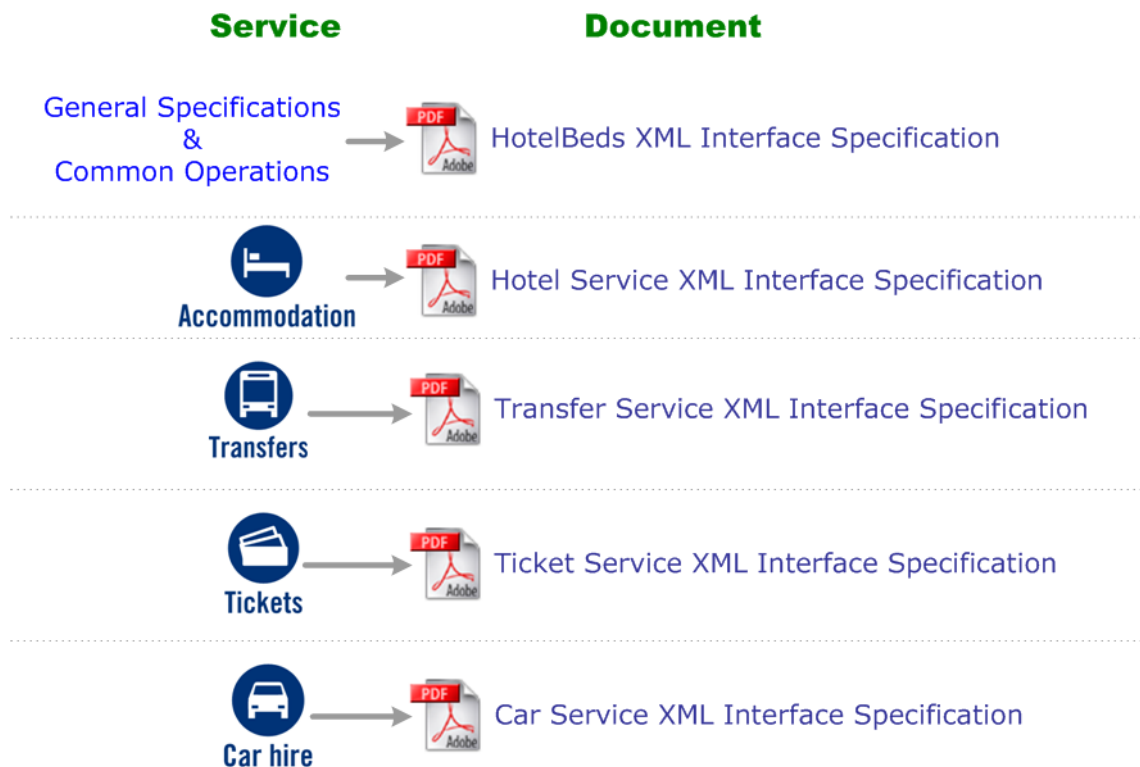


Figure 2 – Documentation files structure

II.3 Version Policy

II.3.1 DOCUMENT VERSION

The version number is divided in three different parts, major version, minor version and release. If the version of the document is 2.4.0 then:

2	4	0
Major version	Minor version	Release

- **Major version**

Indicates the number of the document version, if this version number changes probably a brand new development in the interface is needed. There will not be backwards compatibility.

- **Minor version**

Indicates minor changes in the interface specification like: new functionalities, or changes in the old functionalities. A minor version change can be backwards compatible or not. If new functionalities are added it will be backwards compatible, otherwise if the old functionalities change it will not be backwards compatible.

- **Release**

Indicates a new release of the interface with minor changes like optional fields added to some types, adding a new connection method, but it will always be backwards compatible. It also indicates new documentation (best practices, faqs, changes in appendices, etc...).

II.3.2 INTERFACE XML VERSION

The version number is divided in two different parts, year release and month release.

2010	02
Year release	Month release

To use an specific XML interface version you have to provide an extra attribute in your requests: CoreRequest/@version.

```
<HotelValuedAvailRQ echoToken="DummyEchoToken"
    sessionId="DummySessionId"
    xmlns="http://www.hotelbeds.com/schemas/2005/06/messages"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.hotelbeds.com/schemas/2005/06/messages
    HotelValuedAvailRQ.xsd" version="2010/02">
    ...
```

II.4 Setup Process

In this section you will find how to access the different environments in short and easy steps.

II.4.1 ACCESS TO TEST ENVIRONMENT

- Provide your xml support the ip/ips from which you are planning to access the test server (a maximum of 255 ips can be provided)

- Hotelbeds will configure the system to allow you to access from these ips to the test server
- Configure your interface client using the test server address, in the way that better suits you, web services using WSDL or direct HTTP connection.
- Use the credentials provided by your xml support. These credentials are the same for the live environment. You will not have access to the live environment until you have successfully passed the Acceptance Test (AT) (see Acceptance Test).

II.4.2 ACCESS TO LIVE ENVIRONMENT

- Prior to starting this process you must successfully pass the Acceptance Test (AT) (see Acceptance Test)
- Provide your xml support the ip/ips from which you are planning to access the live server (a maximum of 255 ips can be provided)
- Hotelbeds will grant you the access to the live server
- Configure your interface client using the live server address

III. Technical Information

The interface may be called by any web or classic application supporting standard web protocols like SOAP 1.1 and HTTP 1.1.

You can choose one of the following connection methods:

1. Use the WSDL descriptor and the XML Schema definitions to generate a web services proxy for the interface. A complete XML definition of the interface Web Services operations is available at /wsdl/FrontendService_Live.wsdl or /wsdl/FrontendService_Test.wsdl. These files are identical, the only thing that changes is the port address (live and test respectively). Remember that you must use **wSDL files** we provide at **documentation**, don't generate it.
2. Open a direct HTTP connection to the interface and POST valid xml requests according to the XML Schema definitions. To connect to the interface using HTTP direct connection you must use the following URL's:

	http://212.170.239.71/appservices/http/FrontendService
	http://212.170.239.18/appservices/http/FrontendService

NOTE: If an HTTP request is used, it is assumed the request has a parameter named **xml_request** whose value is a string containing the XML document for the request. If this parameter is sent with another name, you will obtain the error "Unknown operation" (see Troubleshooting).

The requests must be in **UTF-8** (Unicode) encoding. The responses will be in UTF-8 too.

<pre><?xml version="1.0" encoding="UTF-8"?> <soapenv:Envelope soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <soapenv:Body> <hb:getHotelCategoryList xmlns:hb="http://axis.frontend.hydra.hotelbeds.com" xsi:type="xsd:string"> <HotelCategoryListRQ ... > . . . </hb:getHotelCategoryList> </soapenv:Body> </soapenv:Envelope></pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <soapenv:Body> <ns1:getHotelCategoryList xsi:type="xsd:string" xmlns:ns1="http://axis.frontend.hydra.hotelbeds.com"> <HotelCategoryListRS ... > . . . </ns1:getHotelCategoryList> </soapenv:Body> </soapenv:Envelope></pre>
---	--

III.1 Compression in responses

The interface accept compression of the responses. The use of this feature is **mandatory**. To use it please follow the next steps:

	<table border="1"> <tr> <td style="background-color: #cccccc;">:</td> <td></td> </tr> <tr> <td style="background-color: #cccccc;">:</td> <td></td> </tr> </table>	:		:	
:					
:					
	<table border="1"> <tr> <td style="background-color: #cccccc;">:</td> <td style="text-align: right;">e</td> </tr> <tr> <td style="background-color: #cccccc;">:</td> <td style="text-align: right;">e</td> </tr> </table>	:	e	:	e
:	e				
:	e				

III.2 SOAP messages

A SOAP message is an XML document that consists of a mandatory SOAP envelope, an optional SOAP header, and a mandatory SOAP body.

- The Envelope is the top element of the XML document representing the message. The element name MUST be **ENVELOPE**
- The interface doesn't use the header element.
- The Body is a container for mandatory information intended for the interface. SOAP defines one element for the body, which is the Fault element used for reporting errors. The element name MUST be **BODY**

III.2.1 SOAP MESSAGE REQUEST EXAMPLE

III.2.2 HTTP SOAP REQUEST EXAMPLE

```
POST /appservices/ws/FrontendService HTTP/1.1
SOAPAction:
User-Agent: Java/1.4.2_12
Host: 127.0.0.1:8088
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: XXX
```

III.2.3 SOAP MESSAGE RESPONSE EXAMPLE

III.2.4 HTTP SOAP MESSAGE RESPONSE EXAMPLE

The HTTP SOAP response example has been reduced for readability purposes. A real SOAP response should be longer, including more elements.

```
HTTP/1.1 200 OK
Server: Resin/2.1.17
Content-Type: text/xml; charset=utf-8
Transfer-Encoding: chunked
Date: Mon, 17 Jul 2006 15:03:48 GMT
```

IV. Basic Concepts

Three main concepts are used in the interface. Understanding of these three concepts is very important in making a good integration.

IV.1 Product

A product is any available item in the system, it holds descriptive information, and **MAY** be booked. Products will never have information about dates, price, customer, policies, contracts...

- Hotel product
This item will contain descriptive information related to a hotel such as name, chain, category, address, ...
- Car product
This item will contain descriptive information related to a car such as description, features (transmission type, ABS, power steering, seat count...), images, ...
- Ticket product
This item will contain descriptive information related to a ticket such as description, features, ...
- Transfer product
This item will contain descriptive information related to a transfer such as type of transfer (in, out, in/out), transfer vehicle type, ...

IV.2 Service

A service is a product containing information about dates, price, customers, policies, contracts,... and is the minimum booking entity.

e.g.:

- Double hotel room with full board, that has a price of 560€ from August 12 th to August 25 th.
- Car hire from July 12 th to July 15th , economy type, and a price of 135€.

IV.3 Purchase

A purchase is a service container with a status (`SHOPPING_CART`, `BOOKING`, `MODIFIED`, `CANCELLED`), global supplements information, global discount information, global additional costs, a reference number if it is confirmed, currency information and a total price of contained services.

To add a service to a purchase it **MUST** be valued. A purchase will be valid for 30 minutes after you add the first service or can be obtained using `PurchaseDetail` operation. Note that `Purchase/@timeToExpiration` attribute shows the purchase remaining life time in milliseconds. Adding or removing a service to or from a purchase when it is close to becoming invalid, extends purchase's life by five extra minutes.

V. Acceptance Test

V.1 What is the Acceptance Test (AT)?

The acceptance test is the way Hotelbeds ensures that an integration will not compromise the system stability, this is achieved by making some test before allow the client go live. If any problem or issue is detected during the Acceptance Test process the client will be notified.

V.2 Client requirements

- Hotelbeds will need access to the client test server to perform the Acceptance Test for an estimated period of two weeks. You can use either a VPN or a public web site.
- The Acceptance Test will be performed running all the operations developed on the client test server.
- It is mandatory to pass successfully the Acceptance Test before going live.
- In order to do the request data verification you must point your test server to the following urls depending on the frontend you are using:

	<code>http://212.170.239.71/appservices/http/AcceptanceTest</code>
	<code>http://212.170.239.71/appservices/ws/AcceptanceTest</code>

V.3 Acceptance test considerations

V.3.1 DATA CONSISTENCY

The data consistency test prevents you to have inconsistent data inside the Hotelbeds backoffice system and omit mandatory data in the vouchers. All the sensible data shown in your test server (prices, dates,...) during the purchase process will be compared with the data stored in the Hotelbeds backoffice for each service.

Purchase cancellations and modifications (if developed) will also be audited.
The vouchers for all successfully finished purchases will be audited.

V.3.2 REQUEST DATA VERIFICATION

The request data verification test prevents you of sending useless and/or redundant XML data to our system. To avoid massive request or a bad use of operations all the XML requests sent to our system will be monitorized and thoroughly audited during the Acceptance Test.

V.3.3 NET TRAFFIC

In order to finalize the acceptance test you must perform an stress-test. This stress test will be set with at least 30% of your average production server load. It will be performed on every request, and always targeting our live server. Before launching the stress-test you must schedule it with our technical support at xml@hotelbeds.com.

V.4 Steps of certification process

The certification process has 4 steps:

1. We make sure the site will confirm proper bookings. By proper we mean, right prices as well as number of rooms, dates, hotels, type of board, type of room... Customers must be also aware of what hoteliers demand us to show them. In this step you should provide us with a link to your beta site to make some test reservations. We pay special attention to:
 - 1.1. Child ages must be asked on the searching page. They cannot be changed at any step of the booking flow.
 - 1.2. The correct implementation and display of pagination (`<PaginationData itemsPerPage="999" pageNumber="1"/>`)
 - 1.3. The contract remarks (`<Comment type="CONTRACT">`) must be indicated on the booking summary just before booking and also on the voucher. This is critical information from the hotelier for the client and must be indicated together with the room information. The information is provided through the "ServiceAdd" and "PurchaseConfirm" responses. Most hotels in test destination AND (Andorra) have contract remarks.
 - 1.4. In the voucher the supplier information such as supplier name and VAT must be indicated ("Payable through..."). The information is provided through the "PurchaseConfirm" and "PurchaseDetail" responses. Hotelbeds booking reference has to be highlighted. The guests' names (at least one per room), children ages, destination, check in/out dates, room type, board type, hotel details (address, phone and category). For an example, please check voucher.doc.
 - 1.5. Although it's not mandatory, we also recommend showing before confirmation the cancellation policies ("ServiceAdd") and the hotel issues (ExportCSV folder, HotelIssues.csv). Displaying this information will prevent further issues with clients.
 - 1.6. Prices have to be checked in all steps (ValuedAvail, ServiceAdd and PurchaseConfirm). Have in mind that our system has a small percentage (less than 0,1%) of returning a miscalculated price in the search (ValuedAvail) when rates, discounts, supplements or any other condition have been recently changed. Note that the price to be confirmed will be always the one returned in the ServiceAdd response.
 - 1.7. Due to the high volume of requests we experience day by day we have been obliged to ask our clients at developing stage to implement the compression. We also believe you could profit from this feature as it will speed up the communication between your system and ours.

Please note that some destinations offer a lot of products (more than 400). Although time for calculating the availability may be quite fast, the system will spend some extra time sending this information through the Internet connection (700KB-800KB). The system is designed to reduce the amount of time in the transmission step through the compression tool, by reducing the transmission time and response size by about 95%. The compression is part of the standard HTTP protocol (Gzip Algorithm) and you will find a lot of examples on the Internet.
2. Once we are sure customers will be well informed, we analyze the structure of the XML petitions. In order to make this test, you will have to point your system to a certification url that we will provide you, confirm few bookings (we will also provide you the parameters) and email us the reference numbers, in order for us to analyze their code.

HotelBeds IT department will then crosscheck each provided reference number and its details with the logged XML requests. At this point we will review each step of the booking process, each missing or unnecessary parameter or request, and all the recommendations of any type will be underlined to the client. The IT department will be able to request any modification they consider necessary to assure a correct behaviour of the system once the client goes live, such as deleting and adding steps (XML requests) or ask for more tests. A customer's website will not have live access until they receive the good to go from IT, after they are sure that the Acceptance Test has been totally successful.

Some frequent critical errors are:

- 2.1. There are duplicated requests that contain the same information and are sent within a short period of time.
- 2.2. Some steps such as ServiceAddRQ or PurchaseConfirmRQ are repeated unnecessarily. Everytime a ServiceAddRQ is launched without a PurchaseToken this will create a temporary shopping cart that will be stored in memory and the massive creation of them may fill up the memory. On the other hand a duplicated PurchaseConfirmRQ will result in an error once the booking has been confirmed.
- 2.3. Not all the features the website offers to its users are checked with the run test case.
 - a) There are no multiple service bookings (ie. more than one room with the same occupancy, with different occupancy, etc.)
 - b) PurchaseFlush request is not implemented and generates an excessive amount of temporary purchase carts.
 - c) PurchaseCancel is not thrown though it might represent a key action for users.
- 2.4. Unnecessary parameters in any requests that will not be processed. We will inform about those considered useless parameters and suggest to remove them.
- 2.5. The workflow is too confusing because of dozens of Availability requests, unused carts or any other mixed operations that make the Acceptance Test impossible to be reviewed. In this case IT will need all the tests to be repeated again with a simplified process.
- 2.6. Requests that are set to be sent automatically after another one. A common case is a website that sends a PurchaseDetailRQ automatically once the PurchaseConfirmRS has been received while both responses display the same information.
- 2.7. RoomCount element is ignored on Customer list. (Please read FAQ "*How can I search multiple occupancy for a hotel service?*" from Hotel Service document.)
3. In the third step we will simulate the behaviour your site will have once it is live. In other words, you will post a 30% of the number of XML petitions per second you estimate your system will send our production servers, once live, for 10 minutes. In order to do so, you can create a script which posts XML petitions (they can be repeated) in the frequency you think your system will. However, if the amount of XML petitions per minutes that you estimate is low (our IT department decides where the limit is), you might not have to take this test, and you will go straight to live.
4. At this point, you will be already live. However, in order to remain live you will have to confirm a reservation (dates six months in advance) and send us both voucher and price. Bear in mind that you will have to cancel the booking afterwards, otherwise you would be charged. Check with us before cancelling.

Please check the FTP for future revisions of the certification process (document *Certification Process.doc*).

VI. Best Practices

In this section you will find some best practices recommended by our technical staff to make a good integration.

VI.1 SessionId

The different service availabilities are session oriented. The responsibility of generating the session ids relies on the client. For each new request you must generate a different session id. If you are using pagination and retrieving next pages from a previous initial search you must use the same session id, this is because the results are cached using the session id.

VI.2 Search by geographical criteria

Our system is designed to search all the services available in one specific destination for the dates and the occupancies requested. This model allows you to get in a single request all the available services from our inventory. If you need just to check availability for one single product, you can add this information in the availability request, so the system will return you just this product if available. This single product request is useful when you want to recheck availability or when you wish to focus on one particular product.

VI.3 What you should never do

You should NOT use the individual product search to launch multithreaded requests for many individual products.

VI.4 Workflow

To confirm a booking you should send the following requests:

1. Availability
2. ServiceAdd
3. PurchaseConfirm

This is the optimal workflow and we expect you to send it in that way.

There are some inappropriate workflows we will not accept. Here you are some examples:

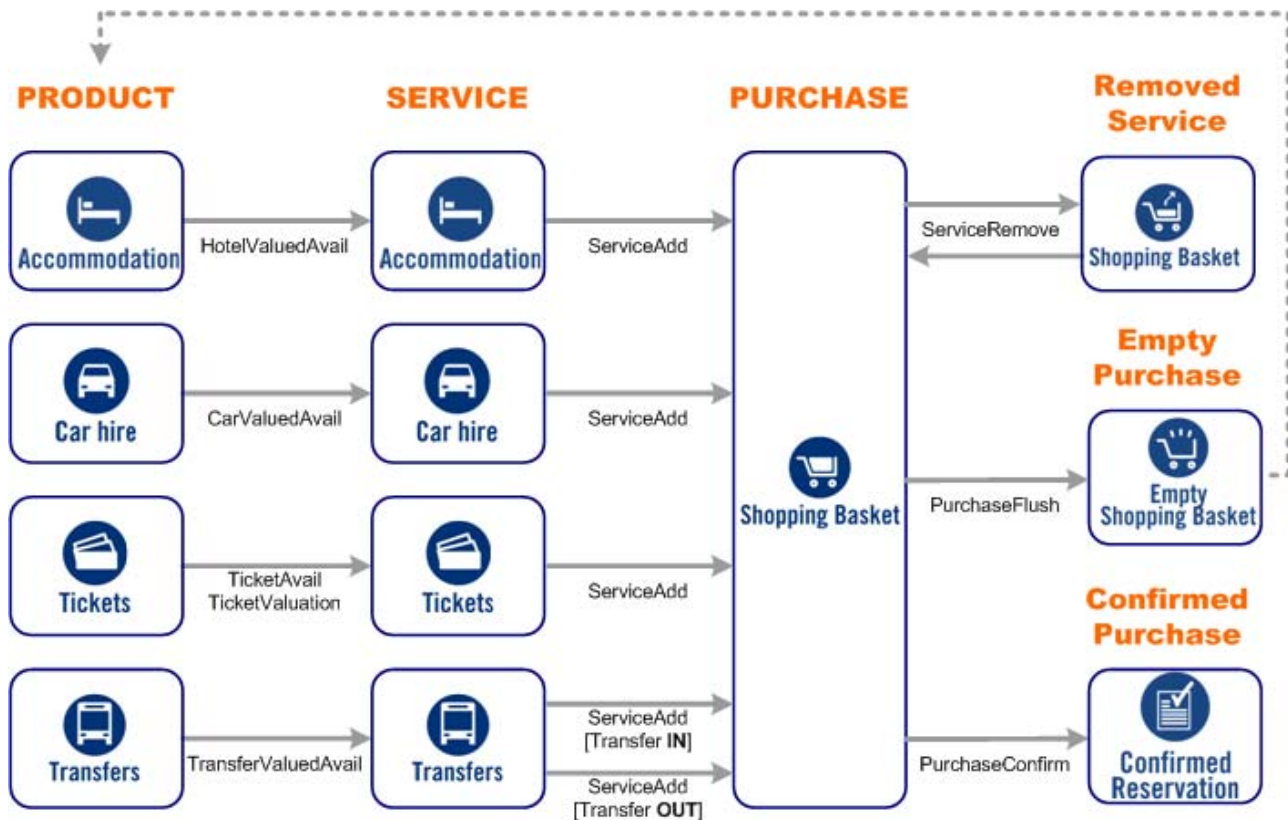
1. Availability
2. ServiceAdd
3. Availability (same as 1)
4. ServiceAdd (same as 2)
5. PurchaseConfirm

1. Availability
2. ServiceAdd
3. ServiceAdd (same as 2)
4. PurchaseConfirm

1. Availability
2. Availability (same as 1)
3. Availability (same as 1)
4. ServiceAdd
5. PurchaseConfirm

VII. Workflows

VII.1 Service confirmation workflow



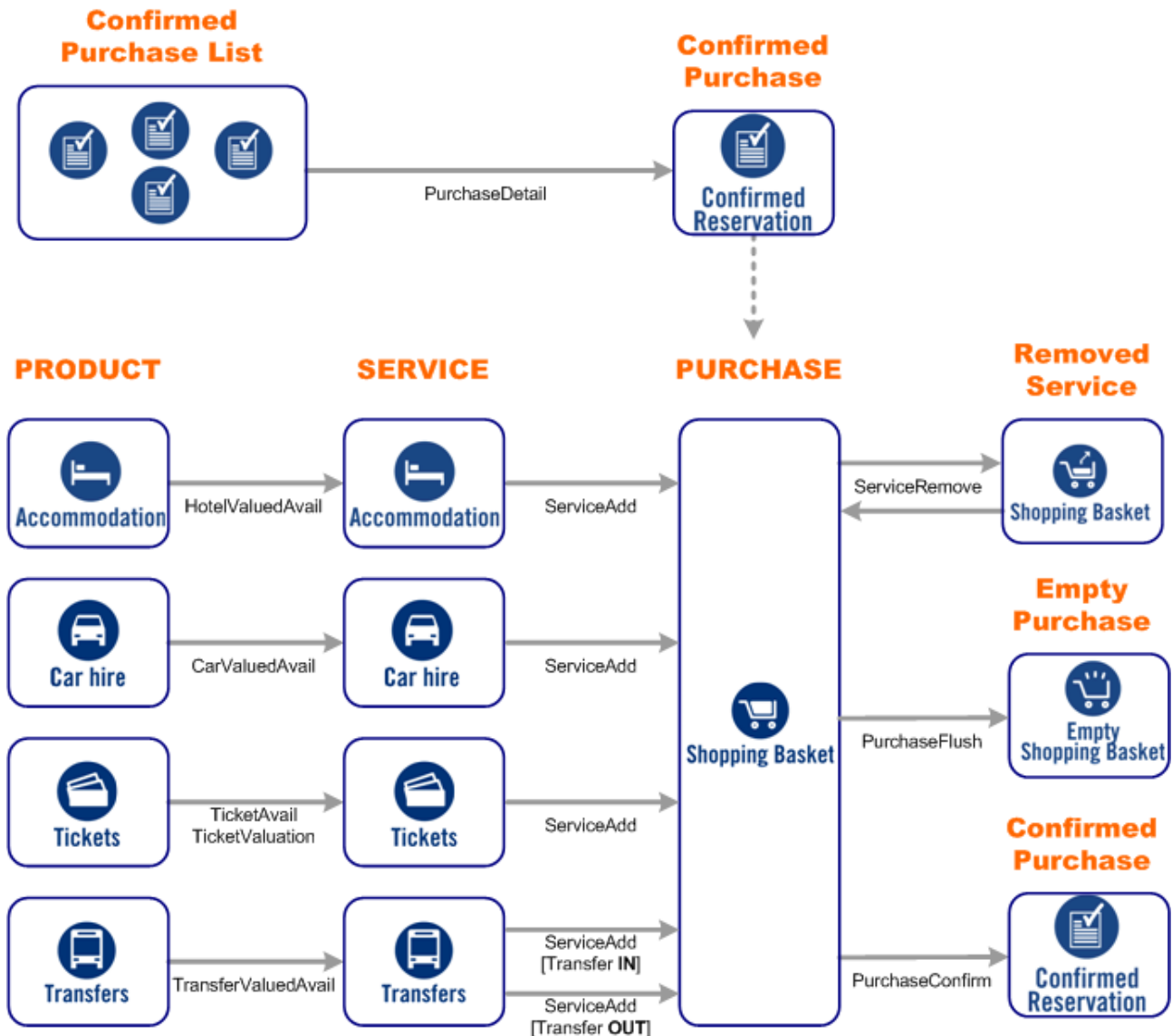
VII.1.1 DESCRIPTION

To confirm a purchase containing multiple or single services you should add new services to the purchase at any time using `ServiceAdd` operation subsequent to new service availability (`CarValuedAvail`, `HotelValuedAvail`, `TicketAvail` - `TicketValuation`, `TransferValuedAvail`).

- Remove an existing service at any time using `ServiceRemove` operation.
- Empty the purchase at any time using `PurchaseFlush` operation.
- Once you have the purchase containing all the desired services you can confirm it using `PurchaseConfirm` operation.

Please note that you can only add to a purchase valued services, the `TicketAvail` operation **does not** return valued ticket services. `TicketValuation` operation **must** be called before adding a ticket service to the purchase.

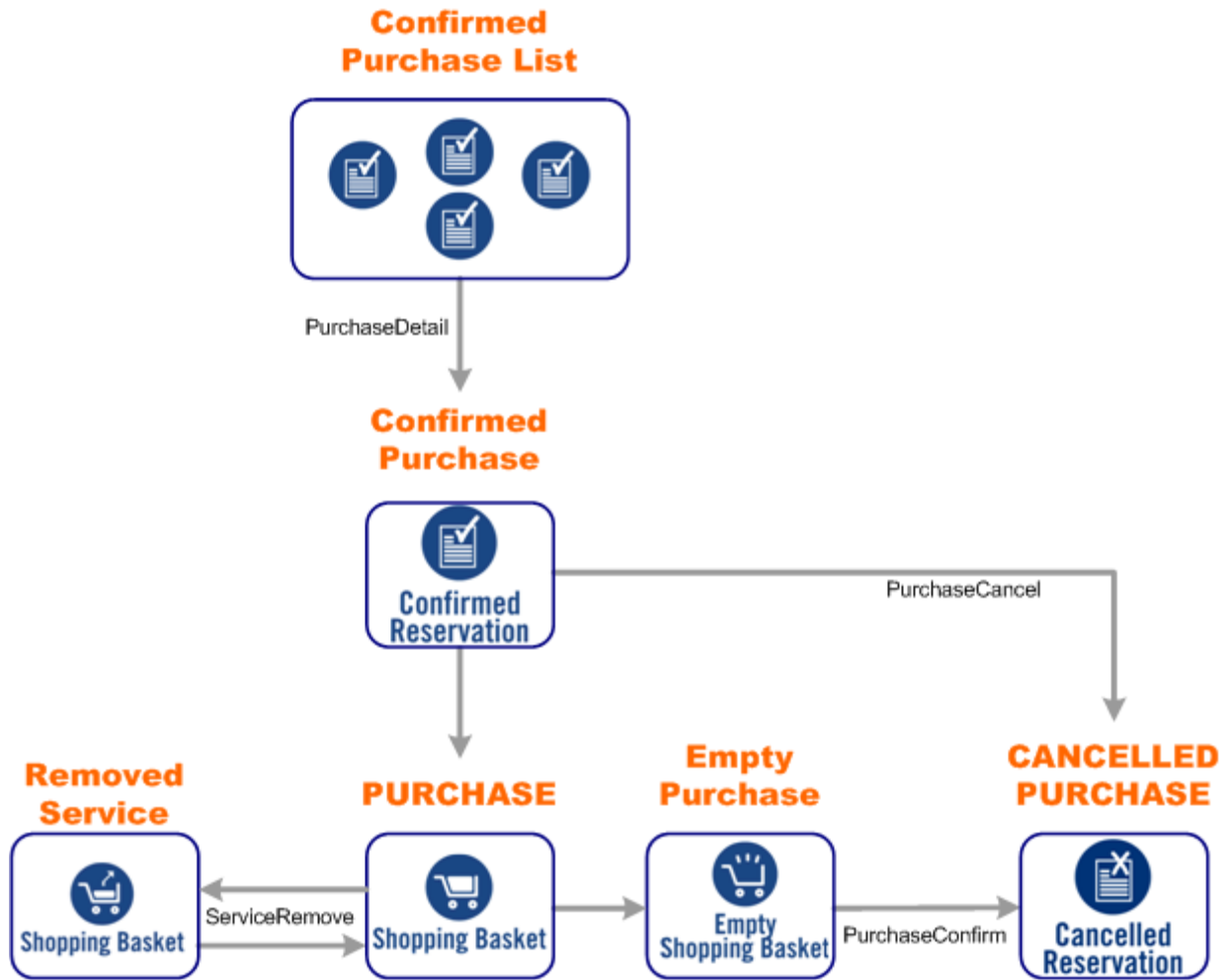
VII.2 Purchase modification (add or remove services) workflow



VII.2.1 DESCRIPTION

To modify a purchase, first of all, you should know which purchase you want to modify. Use `PurchaseList` and `PurchaseDetail` operations to get the desired purchase data. Once you have the purchase data, add new services individually using `ServiceAdd` operation subsequent to new service availability (`CarValuedAvail`, `HotelValuedAvail`, `TicketAvail`, `TicketValuation`, `TransferValuedAvail`). Remove an existing service at any time using `ServiceRemove` operation. Once you have the purchase containing all the desired services you can confirm the purchase using `PurchaseConfirm` operation. Please note that you can only add to a purchase valued services, the `TicketAvail` operation **does not** return valued ticket services. `TicketValuation` operation **must** be called before adding a ticket service to the purchase.

VII.3 Purchase cancellation workflow



VII.3.1 DESCRIPTION

To cancel a purchase, first of all, you should know which purchase you want to cancel. Use `PurchaseList` and `PurchaseDetail` operations to get the desired purchase data. Once you have the purchase data you want to cancel you have two options:

- Remove all the services individually using `ServiceRemove` operation. Confirm the modification using `PurchaseConfirm` operation.
- Cancel the whole purchase using `PurchaseCancel` operation.

VIII. Common Operations

This section show the common operations in use for all services (hotel, transfers, tickets and cars).

VIII.1 ServiceAdd

Use this operation to:

- Add a new service to a purchase:
 - If the Purchase has the `BOOKING` status you can add new services or modify existing services. Note that every service has a modification policy list defining all allowed modifications to this service. You can find this information at `Purchase/ServiceList/Service/ModificationPolicyList` element. You can modify a service only when it has a `MODIFICATION` value at `ModificationPolicy` element. At the moment only some hotel services can be modified. The allowed modifications for a 'modifiable' hotel service are: board modification, add, modify or remove a room (including paxes) and change dates.
 - To add the first service to a purchase use this operation without specifying a `purchaseToken`. This will create a new purchase and will generate a brand new `purchaseToken`.
- Add a new room to a confirmed hotel service:
 - Set the `ServiceAddRQ/Service/@SPUI` attribute with the `SPUI` of the purchase.
 - Set the `ServiceAddRQ/Service/AvailableRoom` with the new selected room.
- Remove only a room from a confirmed hotel service:
 - Set the `ServiceAddRQ/Service/@SPUI` attribute with the `SPUI` of the purchase.
 - Set the `ServiceAddRQ/Service/AvailableRoom` element you want to remove.
 - Set the `ServiceAddRQ/Service/AvailableRoom/HotelRoom/@cancelled` attribute to `Y`.
 - Note that removing the last room from a hotel service is equivalent to calling the `ServiceRemove` operation, it will remove the service from the purchase. If the service is the only service contained in the purchase it is equivalent to removing the last service of the purchase and you will get a purchase containing no services but with the cancellation fees (when applicable).
- Modify a room from a confirmed hotel service:
 - Set the `ServiceAddRQ/Service/@SPUI` attribute with the `SPUI` of the purchase.
 - Set the `ServiceAddRQ/Service/AvailableRoom` element you want to modify.
 - Set the `ServiceAddRQ/Service/AvailableRoom/HotelRoom/@SHRUI` obtained in the availability.
 - Set the `ServiceAddRQ/Service/AvailableRoom/HotelRoom/@modifiesSHRUI` attribute with the `SHRUI` you want to modify.

ServiceAdd

Details:	xsddocs/ServiceAddRQ.html	xsddocs/ServiceAddRS.html
Schema:	xml/ServiceAddRQ.xsd	xml/ServiceAddRS.xsd
Example:	<i>see xml/ServiceAddRQ_serviceName</i>	<i>see xml/ServiceAddRS_serviceName</i>

VIII.1.1 REQUEST BUSINESS RULES

Language	Mandatory
Service/ContractList/Contract	Mandatory. Only the first element will be taken in consideration

VIII.1.2 REQUEST OBSERVATIONS

At the end of the purchase booking process all customer name and last name is mandatory for all services. If you don't provide this information in this step you must provide it in the PurchaseConfirm request operation call.

VIII.1.3 REQUEST ACCEPTANCE TEST

The following elements and attributes are useless:

- ServiceAddRQ/ExtraParamList
- ServiceAddRQ/Service/Reference
- Only first ServiceAddRQ/Service/ContractList/Contract is taken in consideration
- ServiceAddRQ/Service/ContractList/Contract/IncomingOffice/Description
- ServiceAddRQ/Service/ContractList/Contract/IncomingOffice/ContactInfo
- ServiceAddRQ/Service/ContractList/Contract/IncomingOffice/FiscalNumber
- ServiceAddRQ/Service/ContractList/Contract/Sequence
- ServiceAddRQ/Service/ContractList/Contract/Classification
- ServiceAddRQ/Service/ContractList/Contract/CommentList
- ServiceAddRQ/Service/Supplier
- ServiceAddRQ/Service/CommentList
- ServiceAddRQ/Service/DateFrom/@time
- ServiceAddRQ/Service/Currency
- ServiceAddRQ/Service/TotalAmount
- ServiceAddRQ/Service/SupplementList
- ServiceAddRQ/Service/DiscountList
- ServiceAddRQ/Service/AdditionalCostList
- ServiceAddRQ/Service/ErrorList
- ServiceAddRQ/Service/ModificationPolicyList

VIII.1.4 RESPONSE BUSINESS RULES

If ServiceAddRS/Purchase element is present the following elements and attributes must be always present:

- ServiceAddRS/Purchase/@timeToExpiration
- ServiceAddRS/Purchase/Status
- ServiceAddRS/Purchase/Agency
- ServiceAddRS/Purchase/Language
- ServiceAddRS/Purchase/Currency
- ServiceAddRS/Purchase/TotalPrice

VIII.1.5 RESPONSE OBSERVATIONS

Note that if customer data is not specified in the service add request, the system will generate and return empty customer data with a customer id. You must use the customer ids to specify customer data in the purchase confirm request.

VIII.2 ServiceRemove

Remove a service from a purchase.

- When you remove the last service from a purchase that has not already been confirmed, you get a purchase containing no services. You can add new services to this purchase.
- When you remove the last service from a confirmed purchase, you get a purchase containing no services with the cancellation fees (when applicable). After that you must call the PurchaseConfirm operation to cancel the booking. This is equivalent to calling the PurchaseCancel operation.

ServiceRemove

Details:	xsddocs/ServiceRemoveRQ.html	xsddocs/ServiceRemoveRS.html
Schema:	xml/ServiceRemoveRQ.xsd	xml/ServiceRemoveRS.xsd
Example:	<i>see xml/ServiceRemoveRQ_serviceName</i>	<i>see xml/ServiceRemoveRS_serviceName</i>

VIII.2.1 REQUEST BUSINESS RULES

Language	Mandatory

VIII.2.2 REQUEST OBSERVATIONS

To get the purchase token for a purchase with the 'BOOKING' status please use the PurchaseDetail operation. Use the ServiceRemoveRQ/@purchaseToken attribute to select the purchase to remove the service. Use the ServiceRemoveRQ/@SPUI attribute to select the service to remove from the purchase.

VIII.2.3 REQUEST ACCEPTANCE TEST

The following elements and attributes are useless:

- ServiceRemoveRQ/ExtraParamList

VIII.2.4 RESPONSE BUSINESS RULES

If ServiceRemoveRS/Purchase element is present the following elements must be present:

- ServiceRemoveRS/Purchase/@timeToExpiration
- ServiceRemoveRS/Purchase/Status
- ServiceRemoveRS/Purchase/Agency
- ServiceRemoveRS/Purchase/Language
- ServiceRemoveRS/Purchase/Currency
- ServiceRemoveRS/Purchase/TotalPrice

VIII.2.5 RESPONSE OBSERVATIONS

VIII.3 PurchaseFlush

Empty a purchase. This operation is only valid when the purchase has a SHOPPING_CART status.

PurchaseFlush

Details:	xsddocs/PurchaseFlushRQ.html	xsddocs/PurchaseFlushRS.html
Schema:	xml/PurchaseFlushRQ.xsd	xml/PurchaseFlushRS.xsd
Example:	xml/PurchaseFlushRQ.xml	xml/PurchaseFlushRS.xml

VIII.3.1 REQUEST BUSINESS RULES

VIII.3.2 REQUEST OBSERVATIONS

Use PurchaseFlushRQ/@purchaseToken attribute to select the purchase to flush.

VIII.3.3 REQUEST ACCEPTANCE TEST

The following elements and attributes are useless:

- PurchaseFlushRQ/ExtraParamList

VIII.3.4 RESPONSE BUSINESS RULES

VIII.3.5 RESPONSE OBSERVATIONS

VIII.4 PurchaseConfirm

Confirm a purchase. If the purchase has already been confirmed, the purchase data is updated with the provided information.

PurchaseConfirm

Details:	xsddocs/PurchaseConfirmRQ.html	xsddocs/PurchaseConfirmRS.html
Schema:	xml/PurchaseConfirmRQ.xsd	xml/PurchaseConfirmRS.xsd
Example:	xml/PurchaseConfirmRQ.xml	xml/PurchaseConfirmRS.xml

VIII.4.1 REQUEST BUSINESS RULES

You can only provide AGENCY comments at PurchaseConfirmRQ/ConfirmationData/CommentList (Ex: telephone number, email ...).

VIII.4.2 REQUEST OBSERVATIONS

- Holder name and last name are mandatory for new purchase confirmation.
- Agency reference are mandatory for new purchase confirmation.
- Holder and agency reference data cannot be modified. So, you should not send elements Holder and AgencyReference to confirm a modification.
- You can only provide comments for hotel and transfer services.
- You cannot add new or modify comments to a previously confirmed service. To provide the customer data you must specify the same customer id returned in the ServiceAdd response. If you didn't provide the customer name and last name for all customer and services in the ServiceAdd request you must provide it now.
- Age is mandatory for children.

VIII.4.3 REQUEST ACCEPTANCE TEST

The following elements and attributes are useless:

- PurchaseConfirmRQ/ExtraParamList

VIII.4.4 RESPONSE BUSINESS RULES

If PurchaseConfirmRS/Purchase element is present the following elements and attributes must be always present:

- PurchaseConfirmRS/Purchase/@timeToExpiration
- PurchaseConfirmRS/Purchase/Reference
- PurchaseConfirmRS/Purchase/Reference/IncomingOffice
- PurchaseConfirmRS/Purchase/Status
- PurchaseConfirmRS/Purchase/Agency
- PurchaseConfirmRS/Purchase/Language
- PurchaseConfirmRS/Purchase/CreationDate
- PurchaseConfirmRS/Purchase/Holder
- PurchaseConfirmRS/Purchase/Currency
- PurchaseConfirmRS/Purchase/TotalPrice

VIII.4.5 RESPONSE OBSERVATIONS

Note that if you find a PurchaseConfirmRS/Purchase/ServiceList/Service/ErrorList/Error element it means that the confirmation of this service has not been

successful, this implies the purchase has not been confirmed. Note that two different services with the same hotel and dates, after perform the PurchaseConfirm operation will appear grouped in one single service.

VIII.5 PurchaseDetail

Get detailed information from a purchase.

PurchaseDetail

Details:	xsddocs/PurchaseDetailRQ.html	xsddocs/PurchaseDetailRS.html
Schema:	xml/PurchaseDetailRQ.xsd	xml/PurchaseDetailRS.xsd
Example:	xml/PurchaseDetailRQ.xml	xml/PurchaseDetailRS.xml

VIII.5.1 REQUEST BUSINESS RULES

Language	Mandatory
PurchaseReference/IncomingOffice/@code	Mandatory if PurchaseReference is present

VIII.5.2 REQUEST OBSERVATIONS

PurchaseDetailRQ/PurchaseReference element must be provided if the purchase status is BOOKING. PurchaseDetailRQ/PurchaseToken element must be provided if the purchase status is SHOPPING_CART.

VIII.5.3 REQUEST ACCEPTANCE TEST

The following elements and attributes are useless:

- PurchaseDetailRQ/ExtraParamList
- PurchaseDetailRQ/PurchaseReference/IncomingOffice/Description
- PurchaseDetailRQ/PurchaseReference/IncomingOffice/ContactInfo
- PurchaseDetailRQ/PurchaseReference/IncomingOffice/FiscalNumber

VIII.5.4 RESPONSE BUSINESS RULES

If PurchaseDetailRS/Purchase element is present the following elements and attributes must be always present:

- PurchaseDetailRS/Purchase/@timeToExpiration
- PurchaseDetailRS/Purchase/Status
- PurchaseDetailRS/Purchase/Agency
- PurchaseDetailRS/Purchase/Language
- PurchaseDetailRS/Purchase/Currency
- PurchaseDetailRS/Purchase/TotalPrice

VIII.5.5 RESPONSE OBSERVATIONS

VIII.6 PurchaseList

Get a list of purchases according to the requested parameters.

PurchaseList

Details:	xsddocs/PurchaseListRQ.html	xsddocs/PurchaseListRS.html
Schema:	xml/PurchaseListRQ.xsd	xml/PurchaseListRS.xsd
Example:	xml/PurchaseListRQ.xml	xml/PurchaseListRS.xml

VIII.6.1 REQUEST BUSINESS RULES

Holder/Name	Mandatory if Holder is provided
Holder/LastName	Mandatory if Holder is provided

VIII.6.2 REQUEST OBSERVATIONS

VIII.6.3 REQUEST ACCEPTANCE TEST

The following elements and attributes are useless:

- PurchaseFlushRQ/ExtraParamList
- PurchaseListRQ/IncomingOffice/Description
- PurchaseListRQ/IncomingOffice/ContactInfo
- PurchaseListRQ/IncomingOffice/FiscalNumber
- PurchaseListRQ/Destination/Name
- PurchaseListRQ/Destination/ZoneList
- PurchaseListRQ/Reference/IncomingOffice
- PurchaseListRQ/Holder/CustomerId
- PurchaseListRQ/Holder/Age
- PurchaseListRQ/Holder/AdditionalInfo
- PurchaseListRQ/Holder/Document
- PurchaseListRQ/Holder/BirthDate
- PurchaseListRQ/Holder/CountryCode

VIII.6.4 RESPONSE BUSINESS RULES

One of PurchaseListRS/Purchase or PurchaseListRS/ErrorList element must be present unless the PurchaseListRS/@totalItems attribute is 0. If PurchaseListRS/Purchase element is present the following elements and attributes must be present:

- PurchaseListRS/Purchase/@timeToExpiration
- PurchaseListRS/Purchase/Status
- PurchaseListRS/Purchase/Agency
- PurchaseListRS/Purchase/Language
- PurchaseListRS/Purchase/Currency
- PurchaseListRS/Purchase/TotalPrice

VIII.6.5 RESPONSE OBSERVATIONS

If one or more PurchaseListRS/Purchase elements are present PurchaseListRS/@totalItems attribute contains the total number of purchases returned in the response. If no PurchaseListRS/Purchase element is present and PurchaseListRS/@totalItems attribute is 0, no purchases were found matching the search criteria.

VIII.7 PurchaseCancel

Cancel a purchase or valuate a purchase cancellation. This operation is only valid when the purchase has a `BOOKING` status.

PurchaseCancel

Details:	xsddocs/PurchaseCancelRQ.html	xsddocs/PurchaseCancelRS.html
Schema:	xml/PurchaseCancelRQ.xsd	xml/PurchaseCancelRS.xsd
Example:	xml/PurchaseCancelRQ.xml	xml/PurchaseCancelRS.xml

VIII.7.1 REQUEST BUSINESS RULES

Language	Mandatory
PurchaseReference/IncomingOffice	Mandatory

VIII.7.2 REQUEST OBSERVATIONS

Set `PurchaseCancelRQ/@type` attribute value to `v` to get the cost of the purchase cancellation. Using this value does not cancel the purchase, only returns information about the cost of cancellation.

Set `PurchaseCancelRQ/@type` attribute value to `c` to cancel the purchase in the system, note that cancellation fees may apply.

VIII.7.3 REQUEST ACCEPTANCE TEST

The following elements and attributes are useless:

- `PurchaseCancelRQ/ExtraParamList`
- `PurchaseCancelRQ/PurchaseReference/IncomingOffice/Description`
- `PurchaseCancelRQ/PurchaseReference/IncomingOffice/ContactInfo`
- `PurchaseCancelRQ/PurchaseReference/IncomingOffice/FiscalNumber`

VIII.7.4 RESPONSE BUSINESS RULES

If `PurchaseCancelRS/Purchase` element is present the following elements and attributes must be always present:

- `PurchaseCancelRS/Currency`
- `PurchaseCancelRS/Amount`
- `PurchaseCancelRS/Purchase/@timeToExpiration`
- `PurchaseCancelRS/Purchase/Reference`
- `PurchaseCancelRS/Purchase/Reference/IncomingOffice`
- `PurchaseCancelRS/Purchase/Status`
- `PurchaseCancelRS/Purchase/Agency`
- `PurchaseCancelRS/Purchase/Language`
- `PurchaseCancelRS/Purchase/CreationDate`
- `PurchaseCancelRS/Purchase/Holder`
- `PurchaseCancelRS/Purchase/Currency`
- `PurchaseCancelRS/Purchase/TotalPrice`

VIII.7.5 RESPONSE OBSERVATIONS

If `PurchaseCancelRS/@type` attribute value is `v` the response shows the cost of the purchase cancellation. If you get this value the purchase has not been cancelled, only information about the cost of cancellation is provided.

If `PurchaseCancelRS/@type` attribute value is `c` the purchase cancellation has been done, note that cancellation fees may apply. Cancellation cost is shown `PurchaseCancelRS/@type` in element.

VIII.8 IncomingOfficeList

Get the information for each incoming office code provided. If no incoming office codes are provided all incoming office information will be returned.

IncomingOfficeList

Details:	xsddocs/IncomingOfficeListRQ.html	xsddocs/IncomingOfficeListRS.html
Schema:	xml/IncomingOfficeListRQ.xsd	xml/IncomingOfficeListRS.xsd
Example:	xml/IncomingOfficeListRQ.xml	xml/IncomingOfficeListRS.xml

VIII.8.1 REQUEST BUSINESS RULES

Language	Mandatory

VIII.8.2 REQUEST OBSERVATIONS

Provide a list of IncomingOfficeListRQ/IncomingOfficeCode elements to get only the selected incoming offices.

VIII.8.3 REQUEST ACCEPTANCE TEST

The following elements and attributes are useless:

- IncomingOfficeListRQ/ExtraParamList

VIII.8.4 RESPONSE BUSINESS RULES

One of IncomingOfficeListRS/IncomingOffice or IncomingOfficeListRS/ErrorList elements must be present in the response unless the @totalItems attribute is 0.

If IncomingOfficeListRS/IncomingOffice element is present the following elements and attributes must be always present:

- IncomingOfficeListRS/IncomingOffice/Description
- IncomingOfficeListRS/IncomingOffice/FiscalNumber

VIII.8.5 RESPONSE OBSERVATIONS

If no IncomingOfficeListRS/IncomingOffice element is present and /@totalItems attribute is 0, no incoming offices were found matching the search criteria.

If one or more IncomingOfficeListRS/IncomingOffice elements are present:

- /@totalItems attribute contains the total number of incoming offices returned in the response.

IX. FAQ

1. Which encoding type should I use in XML requests and responses?

The requests must be encoded in UTF-8. Responses are also encoded in UTF-8.

2. What is the availToken?

The availToken is a temporary availability identifier. Use this token to add a service to a purchase.

3. What is the SPUI?

The SPUI is the Service Purchase Unique Identifier. It identifies a service inside a purchase.

4. What is the purchaseToken?

The purchaseToken is a temporary purchase identifier. This identifier is needed to make operations with purchases.

The following rules are applied to the purchaseToken:

- Adding the first service to a purchase will generate a brand new purchaseToken.
- Getting the purchase detail from a confirmed purchase (using PurchaseDetail operation) will generate a brand new purchaseToken.
- PurchaseList operation will NOT generate usable purchase tokens.
- Confirming a purchase will invalidate its purchaseToken.
- After 30 minutes of a purchaseToken creation it will become invalid. Note that Purchase/@timeToExpiration attribute shows the purchaseToken's remaining life time in milliseconds. Adding or removing a service to or from a purchase when it is near to becoming invalid, extends the purchaseToken's life by five extra minutes.

5. How can I add the first service to a purchase?

To add the first service to a purchase use the ServiceAdd operation without specifying a purchaseToken. This will create a new purchase and will generate a brand new purchaseToken.

6. How can I get a purchaseToken to start working with a purchase?

If you haven't created a purchase, you need to add the first service to the purchase (see above). If you already have a confirmed purchase get its detail using PurchaseDetail operation. Note that you can get a list of confirmed purchases using the PurchaseList operation, this operation does NOT generate usable purchase tokens.

7. How long will a purchaseToken be valid without being confirmed?

The purchaseToken will be valid for 30 minutes subsequent to you adding the first service or getting it using PurchaseDetail operation. Note that Purchase/@timeToExpiration attribute shows the purchaseToken's remaining life time in milliseconds. Adding or removing a service to or from a purchase when it is near to becoming invalid, extends the purchaseToken's life by five extra minutes.

8. Can a service availability return different services with the same product code but different contracts?

Yes, a service availability can return different services with the same product code but different contract code. The system works with contracts, this means that a service availability can return the same product more than once but with different contracts, this usually happens when different prices are provided for the same product and different contracts.

9. How can I specify the customer data of a service?

Prior to adding customer data to a service it is necessary to understand what `CustomerId` tag means. Each tag representing a customer has a `CustomerId` tag, this id is generated and provided by our system. You can use this id to identify a customer inside a purchase.

The following rules are applied to specify the customer data of a service when you are using the `ServiceAdd` operation:

- The customer data is optional. If you send it, it will be stored in the system and returned in the response with the corresponding `CustomerId`. You can modify this data later when you call the `PurchaseConfirm` operation.
- In the case of children the age is mandatory.
- If no customer data is provided, empty customer data will be generated by the system and returned in the response with the corresponding `CustomerId` and empty data.
- Customer ids sent in the `ServiceAdd` operation will be ignored.
- If you are using the `ServiceAdd` operation to modify an existing purchase you cannot reuse the old customer data, you need to provide new data.
- To specify the customer data of a service when you are using the `PurchaseConfirm` operation:
 - You should provide the customer data you want to add or modify specifying the `CustomerId` provided in the `ServiceAdd` operation response.
 - At least the name of one adult customer must be provided for each service, thus you cannot confirm a service containing only children.

10. Which modifications are allowed for a confirmed purchase?

Only the Hotel service allows modifications for a confirmed purchase.

11. How can I add a new service to a confirmed purchase?

To add a new service to a confirmed purchase follow the steps below:

- Make the desired service availability. Note that you can only add to a purchase valued services, the `TicketAvail` operation does not return valued ticket services. `TicketValuation` operation MUST be called before adding a ticket service to the purchase.
- Call the `ServiceAdd` operation with the following considerations:
 - Set the `ServiceAddRQ/@purchaseToken` attribute with the purchase token of the purchase you want to add the service.
 - Set the `ServiceAddRQ/Service` element with the selected service data.
- Call `PurchaseConfirm` operation to commit the changes to the system.

Note that you need the purchase detailed information to get a valid purchase token. (see `PurchaseDetail` operation).

12. How can I remove a service from a confirmed purchase?

To remove a service from a confirmed purchase follow the steps below:

- Call the `ServiceRemove` operation with the following considerations:
 - Set the `ServiceRemoveRQ/@purchaseToken` attribute with the purchase token of the purchase you want to remove from the service.
 - Set the `ServiceRemoveRQ/@SPUI` attribute with the SPUI of the service you want to remove.
- Call `PurchaseConfirm` operation to commit the changes to the system.

Note that you need the purchase detailed information to get a valid purchase token. (see `PurchaseDetail` operation).

13. What is the system behaviour when I remove the last service of a purchase?

You can find two possible situations:

- When you remove the last service from a purchase that has not already been confirmed, you get a purchase containing no services. You can add new services to this purchase.
- When you remove the last service from a confirmed purchase, you get a purchase containing no services but with the cancellation fees (when applicable). After that you MUST call the `PurchaseConfirm` operation to cancel the booking. This is equivalent to calling the `PurchaseCancel` operation.

14. What is an XSD ?

XML Schema, published as a W3C recommendation in May 2001, is one of several XML schema languages. It was the first separate schema language for XML to achieve Recommendation status by the W3C.

Like all XML schema languages, XML Schema can be used to express a schema: a set of rules to which an XML document must conform in order to be considered 'valid' according to that schema. However, unlike most other schema languages, XML Schema was also designed with the intent that determination of a document's validity would produce a collection of information adhering to specific data types. Such a post-validation info set can be useful in the development of XML document processing software, but the schema language's dependence on specific data types has provoked criticism.

An XML Schema instance is an XML Schema Definition (XSD) and typically has the filename extension ".xsd". The language itself is sometimes informally referenced as XSD. It has been suggested that WXS (for W3C XML Schema) is a more appropriate initialism though this acronym has not been in a widespread use and W3C working group rejected it. XSD is also an initialism for XML Schema Datatypes, the datatype portion of XML Schema.

15. Why is defined the type of elements like 'anyType' at wsdl's?

Hotelbeds xml specification is not a closed language. This means that objects can be modified including new elements. In order to avoid issues to clients' integrations, we define the type of requests and responses like 'anyType'. A proxy generator will create them as Object types and in that way new elements in specification will not cause validation errors.

16. Which currency is returned by the system?

Price of services can be returned in different currencies. There are three default currencies available:

GBP	United Kingdom Pound	UK
USD	US Dollar	America and Caribbean
EUR	Euro	Rest of countries

The currencies returned will depend as well on the bussiness model.

X. Troubleshooting

In this section you will find possible problems and solutions that may help you to make an integration with the interface.

X.1 I get the HTTP Status code 500 with the following error message: “IndexOutOfBoundsException: Index: 0, Size: 0” error.

Problem: When I send a soap request to the Web Services frontend I get an “IndexOutOfBoundsException: Index: 0, Size: 0” error.

Cause: This error is usually caused when the hotelbeds namespace is not provided in the soap request.

Solution: You need to include the hotelbeds namespace declaration:
`http://axis.frontend.hydra.hotelbeds.com`.

For instance, a correct hotel category list XML request for the Web services frontend is:

X.2 I get “Envelope is not a valid tag name.” error.

Problem: When I send any request to the HTTP frontend I get “Envelope is not a valid tag name” error.

Cause: This error is caused when you try to access the HTTP frontend using a SOAP request.

Solution: Remove the SOAP envelope and body from the XML request.
For instance, a correct hotel category list XML request for the HTTP frontend is:

X.3 I get the HTTP Status code 500 with the following error message: “Unknown operation”

Problem: When I send a XML request to the HTTP frontend I get the Unknown operation error.

Cause: This error may be caused by any combination of the following reasons:

1. You use the GET method.
2. You didn't provide the `xml_request` param name.
3. You didn't send a valid XML request.

Solution: Ensure you are using POST method to access the HTTP frontend, sending the XML request with a parameter named `xml_request` and is a valid XML operation.

X.4 I get the HTTP Status code 500 with the following error message: “Method ‘GET’ not allowed, please use ‘POST’ method”

Problem: When I send a XML request to the HTTP frontend I get the Method GET not allowed, please use POST method error.

Cause: This error is caused when you use the GET method instead of POST.

Solution: Use POST method to access the HTTP frontend.

XI. Conventions

XI.1 Language Codes

One more characteristic of the interface is the capability of returning all the information in the requested language. The valid languages codes are not available through interface operations and are shown in the table below.

Spanish	CAS
French	FRA
English	ENG
German	ALE
Portuguese	POR
Italian	ITA
Dutch	HOL

XI.2 Error Codes

In this appendix you will find the error codes returned by the system and their meanings. Error codes are dynamically generated by the system, thus a complete list of error codes cannot be provided. You can find three different error categories.

XI.2.1 APPLICATION ERRORS

Format: AXX-XX-XXX-XX where x can be any character between [A-Z] or [0-9]

Cause: These errors are caused by an internal application error. You can't do anything to fix them but send it to hotelbeds technical support. Although there is a set of application error codes that depends on the XML request you sent. The A12-02-XXX-XX errors are caused by a XML Schema restriction violation. A description message in the requested language will be provided to allow you to correct the error.

Example: A12-02-A3R-05

XI.2.2 BUSSINES LOGIC ERRORS

Format: BXX-XX-XXX-XX where x can be any character between [A-Z] or [0-9]

Cause: These errors are caused by a bussines logic restriction violation. A description message in the requested language will be provided to allow you to correct the error.

Example: B01-01-123-07

XI.2.3 COMMUNICATION ERRORS

Format: CXX-XX-XXX-XX where x can be any character between [A-Z] or [0-9]

Cause: These errors are caused by a grant error. They occur when you are accesing from an unknown ip, you are accessing to the wrong environment or your credential details are wrong.

Example: C01-01-007-11