

**Oracle® Hospitality Suite8**  
XML Interface Specifications  
Release 8.11.0

December 2017

---

Copyright © 2002, 2017, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

---

---

# Contents

<b>Preface</b> .....	<b>12</b>
Audience .....	12
Customer Support.....	12
Documentation.....	12
Revision History.....	12
<b>1 Introduction</b> .....	<b>17</b>
XML Interface and OTA XML Interface .....	17
XML Interface and X64 Processors .....	17
<b>2 System Overview</b> .....	<b>18</b>
Provided functions.....	18
Communication.....	18
Message format .....	19
Data format .....	19
Communication Packing.....	20
Using Translation Tables .....	21
General Notes .....	21
Setup .....	21
Response Codes.....	22
Conditions definition.....	22
Licensing .....	22
<b>3 Fields available for selection by request type</b> .....	<b>27</b>
Departments .....	27
Reservation .....	27
Profile.....	29
Posting .....	33
SystemDate .....	33
SystemInfo .....	33
AvailabilityForWeb .....	33
ChildrenCategories.....	34
CreditCardTypes.....	34
Login (UserLogin).....	34
NoteCategory .....	34
Note.....	35
General note request.....	35
HtmlInfo.....	35

Event Info Board .....	35
Activity .....	36
Task .....	36
Group Template .....	37
Group .....	37
BlockHeader .....	37
BlockDetail .....	38
Country .....	38
CurrencyList .....	38
ConferenceStatusList .....	38
ConferencePackageList .....	38
HotelAvailability .....	39
Overbooking .....	39
Market .....	39
MainMarket .....	39
RateList .....	39
ReportList .....	40
ReportParameterList .....	40
ReportSectionList .....	41
RoomType .....	41
StatisticAndFuture .....	42
CalendarEvent .....	43
VoucherArticles .....	43
Voucher .....	44
BillingInstruction .....	44
Preference .....	45
Package .....	45
TableReservation .....	46
TableAvailability .....	46
AccountManager .....	46
ConferenceManager .....	46
Channel .....	47
Source .....	47
ConferenceStatusList .....	47
FunctionType .....	47
SeatingArrangments .....	47
EventType .....	47
EventStatus .....	47
FunctionSpace .....	48

---

FunctionSpaceSeating.....	48
FunctionSpaceRate.....	48
FunctionSpaceAvailability.....	49
ConferencePackagesList.....	50
CateringItem.....	51
CateringItemRate.....	52
StandardMenu.....	52
StandardMenuRate.....	53
MenuItem.....	53
MenuItemRate.....	54
ConferenceBooking.....	55
BookingPackage.....	56
BookingEvent.....	57
EventFunctionSpace.....	59
EventCateringItem.....	59
EventStandardMenu.....	60
StandardMenuMenuItem.....	62
<b>4 Available search criteria by request type.....</b>	<b>63</b>
Departments.....	63
Reservation.....	63
Profile.....	64
SystemDate.....	64
SystemInfo.....	64
AvailabilityForWeb.....	64
Posting.....	65
ChildrenCategories.....	65
CreditCardTypes.....	65
HtmlInfo.....	65
SeatingArrangments.....	65
Event Info Board.....	65
Activity.....	66
Task.....	66
Group Template.....	66
Group.....	66
BlockHeader.....	66
BlockDetail.....	66
CurrencyList.....	67
HotelAvailability.....	67
Overbooking.....	67

Market.....	67
MainMarket .....	67
RateList.....	67
ReportList.....	67
ReportParameterList.....	67
ReportSectionList.....	68
StatisticAndFuture (planned functionality only).....	68
CalendarEvent.....	68
VoucherArticles.....	68
Voucher .....	68
BillingInstruction .....	69
Package.....	69
RoomType.....	69
TableReservation.....	69
TableAvailability.....	69
NoteCategory .....	70
Note.....	70
Conference and Catering: General available search criteria .....	70
FunctionType.....	71
ConfiguredStatusList.....	71
EventType .....	71
EventStatus .....	71
FunctionSpace.....	71
FunctionSpaceSeating.....	71
FunctionSpaceRate.....	71
FunctionSpaceAvailability.....	71
ConferencePackagesList.....	72
CateringItem.....	72
CateringItemRate .....	72
StandardMenu.....	72
MenuItem.....	72
ConferenceBooking.....	72
BookingPackage .....	72
BookingEvent .....	73
EventFunctionSpace .....	73
EventCateringItem.....	73
EventStandardMenu.....	73
StandardMenuMenuItem .....	73

**5 Fields available to update / insert ..... 74**

---

Reservation .....	74
Profile.....	76
GuestAnonymization .....	82
Voucher .....	82
Profile Merge .....	82
ProfileAndReservation.....	83
FinancialAccount .....	83
Activity .....	83
Task.....	84
Group Template .....	84
Group.....	85
CreditCardApproval (Insert only).....	85
HurdleRate.....	85
Overbooking .....	86
BAR .....	86
BillingInstruction .....	86
Table Reservation.....	87
Postings .....	88
Cancellation of reservations .....	88
Note.....	89
Conference and Catering General.....	89
ConferenceBooking.....	90
BookingPackage .....	91
BookingEvent .....	92
EventFunctionSpace .....	93
EventCateringItem.....	94
EventStandardMenu.....	94
StandardMenuMenuItem .....	95
Voucher Redemption.....	96
<b>6 Conference and Catering General.....</b>	<b>97</b>
ConferenceBooking.....	97
BookingPackage .....	98
BookingEvent .....	98
EventFunctionSpace .....	99
EventCateringItem.....	99
EventStandardMenu.....	100
StandardMenuMenuItem .....	100
<b>7 Schema FidelioXML.xsd.....</b>	<b>102</b>

element fidelio.....	103
element fidelio/request.....	108
element fidelio/response.....	109
element fidelio/response/queryResponse.....	111
element fidelio/response/queryResponse/rows.....	111
element fidelio/response/insertResponse.....	112
element fidelio/response/insertResponse/rows.....	113
element fidelio/response/insertResponse/rows/ProfileID.....	114
element fidelio/response/insertResponse/rows/FANumber.....	114
element fidelio/response/insertResponse/rows/GuestNum.....	114
element fidelio/response/insertResponse/rows/ProfileGlobalID.....	114
element fidelio/response/insertResponse/rows/LastUpdateTime.....	114
element fidelio/response/insertResponse/rows/Addresses.....	114
element fidelio/response/insertResponse/rows/Addresses/Address.....	115
element fidelio/response/insertResponse/rows/Communications.....	115
element fidelio/response/insertResponse/rows/Communications/Communication.....	115
element fidelio/response/insertResponse/rows/Notes.....	115
element fidelio/response/insertResponse/rows/Notes/Note.....	116
element fidelio/response/insertResponse/rows/TaskID.....	116
element fidelio/response/insertResponse/rows/ActivityID.....	116
element fidelio/response/insertResponse/rows/GroupID.....	116
element fidelio/response/insertResponse/rows/GroupTemplateID.....	116
element fidelio/response/updateResponse.....	117
element fidelio/response/updateResponse/rows.....	117
complexType AvailabilityForWebType.....	117
element AvailabilityForWebType/fields.....	118
complexType BlockDetailConditionType.....	118
complexType BlockHeaderConditionType.....	119
complexType ccm_book_actualsType.....	120
element ccm_book_actualsType/bookingID.....	120
element ccm_book_actualsType/event_id.....	121
element ccm_book_actualsType/postingDate.....	121
element ccm_book_actualsType/menu_id.....	121
element ccm_book_actualsType/menu_item_id.....	121
element ccm_book_actualsType/misc_item_id.....	121
element ccm_book_actualsType/function_space_id.....	122
element ccm_book_actualsType/ResourceName.....	122
element ccm_book_actualsType/DepartmentCode.....	122



---

element ccm_book_actualsType/count .....	122
element ccm_book_actualsType/total_amount .....	122
element ccm_book_actualsType/OverrideReason .....	123
element ccm_book_actualsType/UserName .....	123
element ccm_book_actualsType/Covers.....	123
complexType ChildrenCategoriesConditionType .....	123
complexType commandType.....	124
element commandType/post.....	125
element commandType/checkIn.....	125
element commandType/checkIn/RoomNumber .....	125
element commandType/checkOut.....	125
element commandType/checkOut/RoomNumber .....	125
element commandType/cancel .....	126
element commandType/cancel/GuestNum.....	126
element commandType/cancel/Reason .....	126
element commandType/cancel/DeleteTraces.....	126
element commandType/cancel/SendEmail .....	127
element commandType/resetProfilePassword .....	127
element commandType/resetProfilePassword/userName.....	127
complexType conditionsType.....	127
element conditionsType/condition.....	128
element conditionsType/link.....	128
complexType conditionType.....	128
complexType ConditionType_ConfiguredStatusList .....	129
complexType ConditionType_InfoBoard.....	129
complexType CreditCardTypesConditionType .....	130
complexType crm_postType .....	131
element crm_postType/GuestNum .....	132
element crm_postType/PostingDate .....	132
element crm_postType/RoomNum.....	132
element crm_postType/Quantity .....	132
element crm_postType/Price.....	133
element crm_postType/Comment .....	133
element crm_postType/CheckNum .....	133
element crm_postType/CheckDate .....	133
element crm_postType/Check.....	134
element crm_postType/Description .....	134
element crm_postType/Currency .....	134
element crm_postType/ProfileID.....	134

complexType CurrencyListConditionType .....	135
complexType DepartmentCodeConditionType .....	135
complexType fieldsType .....	135
element fieldsType/field .....	136
complexType fieldType .....	136
complexType HotelAvailabilityType .....	139
element HotelAvailabilityType/fields .....	140
complexType importType .....	140
element importType/insert .....	140
element importType/insert/rows .....	141
element importType/update .....	141
element importType/delete .....	141
element importType/delete/conditions .....	142
complexType insertType .....	142
element insertType/fields .....	143
complexType linkType .....	143
element linkType/condition .....	143
complexType PostngConditionType .....	144
complexType postResponseType .....	144
complexType postType .....	144
element postType/ccm_book_actuals .....	145
element postType/crm_posting .....	146
element postType/crm_posting/DepartmentCode .....	147
element postType/crm_posting/ArticleCode .....	147
complexType ProfileConditionType .....	147
complexType queryType .....	148
element queryType/conditions .....	149
element queryType/fields .....	150
element queryType/froms .....	150
element queryType/froms/from .....	150
element queryType/orders .....	150
element queryType/orders/order .....	151
element queryType/groups .....	151
element queryType/groups/group .....	151
complexType ReportType .....	152
element ReportType/ReportID .....	152
element ReportType/ReportType .....	153
element ReportType/Parameter .....	153
element ReportType/Destination .....	153

---

element ReportType/ExportFormat .....	153
element ReportType/Email .....	154
element ReportType/ConfirmationArchiveRequest .....	154
complexType requestType .....	155
element requestType/query .....	156
element requestType/update .....	157
element requestType/insert .....	158
element requestType/command .....	158
element requestType/AvailabilityForWeb .....	159
element requestType/ValidateCreditCard .....	160
element requestType/import .....	160
element requestType/HotelAvailability .....	160
element requestType/Report .....	161
complexType ReservationConditionType .....	161
complexType rowsType .....	162
element rowsType/row .....	162
complexType rowType .....	163
element rowType/fields .....	163
complexType SystemDateConditionType .....	163
complexType SystemInformationConditionType .....	164
complexType updateType .....	164
element updateType/conditions .....	165
element updateType/fields .....	165
complexType ValidateCreditCardType .....	165
element ValidateCreditCardType/CreditCardNo .....	166
element ValidateCreditCardType/ExpiryMonth .....	166
element ValidateCreditCardType/ExpiryYear .....	166
element ValidateCreditCardType/CardholderName .....	166
element ValidateCreditCardType/CreditCardCode .....	166
simpleType binaryOperation .....	167
simpleType conditionSimpleType .....	167

---

---

# Preface

The Suite8 XML Interface is a service allowing external applications to communicate with Suite8 based on a unified abstract interface. This interface detaches the concept from a physical database structure and provides additional business logic and security when accessing data. The XML interface cannot be used for any Central Reservation or Point of Sales development.

## Audience

This interface specification is intended for the third party software vendors who are developing a system which needs connectivity to Suite8.

This document is intended for distribution to all vendors and developers of hotel software systems. The data contained herein is public information and is published for public use. However, if you have received this copy from a company other than Oracle Hospitality you may want to contact us to ensure that you have the most recent version of this document.

## Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:

<https://support.oracle.com>

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received and any associated log files
- Screenshots of each step you take

## Documentation

Oracle Hospitality product documentation is available on the Oracle Help Center at

<http://docs.oracle.com/en/industries/hospitality/>

## Revision History

Date	Description of Change
26 June, 2004	<ul style="list-style-type: none"><li>• 1.0.0 - First Draft</li></ul>
01 November, 2004	<ul style="list-style-type: none"><li>• Myfidelio Homepage support. Creation of FA via XML Interface</li></ul>
04 February, 2005	<ul style="list-style-type: none"><li>• 1.0.2 - CCM Actual Booking posting</li></ul>
14 July, 2005	<ul style="list-style-type: none"><li>• 1.0.3 - Profile interface extensions, additional CCM functions</li></ul>

Date	Description of Change
07 March, 2006	<ul style="list-style-type: none"> <li>1.0.4 - Operations with group reservations</li> </ul>
21 September, 2006	<ul style="list-style-type: none"> <li>1.0.5 - Blocks. Credit card approvals</li> </ul>
10 October, 2006	<ul style="list-style-type: none"> <li>1.0.6 - Allows to specify profile for postings</li> </ul>
09 November, 2006	<ul style="list-style-type: none"> <li>1.0.7 - Hurdle Rates Interface</li> </ul>
22 March, 2007	<ul style="list-style-type: none"> <li>1.0.8 - Conference booking messages</li> </ul>
08 May, 2007	<ul style="list-style-type: none"> <li>1.0.9 - Extended AvailabilityForWeb</li> </ul>
11 May, 2007	<ul style="list-style-type: none"> <li>1.0.10 - HotelAvailability</li> </ul>
12 July, 2007	<ul style="list-style-type: none"> <li>1.0.12 - Additional Profile Fields</li> </ul>
19 July, 2007	<ul style="list-style-type: none"> <li>1.0.13 - HotelAvailability request extended</li> </ul>
30 August, 2007	<ul style="list-style-type: none"> <li>1.0.15 - Additional reservation search criteria</li> </ul>
10 October, 2007	<ul style="list-style-type: none"> <li>1.0.16 - Additional reservation search criteria</li> </ul>
25 October, 2007	<ul style="list-style-type: none"> <li>1.1.1 - Added version 1.1.0 of the request/response specification</li> </ul>
23 January, 2008	<ul style="list-style-type: none"> <li>1.1.2 - Added Reservation Details request mode</li> </ul>
20 March, 2008	<ul style="list-style-type: none"> <li>1.1.3 - Added support of reservation share and parties</li> </ul>
15 May, 2008	<ul style="list-style-type: none"> <li>1.1.4 - Added maximum field length for the update/insert messages</li> </ul>
16 June, 2008	<ul style="list-style-type: none"> <li>1.1.5 - Added Translation Tables (currently 8.8 only)</li> </ul>
25 June, 2008	<ul style="list-style-type: none"> <li>1.1.6 - Added Information about required licenses</li> </ul>
21 August, 2008	<ul style="list-style-type: none"> <li>1.1.7 - Added RateList message</li> </ul>
27 August, 2008	<ul style="list-style-type: none"> <li>1.1.8 - Added BAR message, explained communication packing</li> </ul>
19 September, 2008	<ul style="list-style-type: none"> <li>1.1.9 - Added messages to work with reports</li> </ul>
07 October, 2008	<ul style="list-style-type: none"> <li>1.1.10 - Added message to create profile and reservation within 1 message.</li> </ul>
14 October, 2008	<ul style="list-style-type: none"> <li>1.1.11 - Added additional fields to the ReportParameter request</li> </ul>

Date	Description of Change
17 October, 2008	<ul style="list-style-type: none"> <li>1.1.12 - Additional Profile and Reservation fields / conditions</li> </ul>
03 June, 2009	<ul style="list-style-type: none"> <li>1.1.13 - Reservation packages and Version info</li> </ul>
12 June ,2009	<ul style="list-style-type: none"> <li>1.1.14 - Reservation and Profile custom fields</li> </ul>
18 June, 2009	<ul style="list-style-type: none"> <li>1.1.15 - Added description of profile communications</li> </ul>
27 August, 2009	<ul style="list-style-type: none"> <li>1.1.16 - Added StatisticAndFuture (planned functionality only)</li> </ul>
31 August, 2009	<ul style="list-style-type: none"> <li>1.1.17 - Added support of Guest 3rd Party IDs to the reservation message</li> </ul>
17 September, 2009	<ul style="list-style-type: none"> <li>1.1.18 - Extended AvailabilityForWeb</li> </ul>
12 February, 2010	<ul style="list-style-type: none"> <li>1.1.19 - HotelSegmentCode for AvailabilityForWeb</li> </ul>
29 March, 2010	<ul style="list-style-type: none"> <li>1.1.20 - Reservation data changes during the stay</li> </ul>
October, 2010	<ul style="list-style-type: none"> <li>8.8.4.0 - Added example for Profile Merge</li> </ul>
February, 2011	<ul style="list-style-type: none"> <li>8.8.4.3 - Added information about XML Interface &amp; X64 processors</li> </ul>
May, 2011	<ul style="list-style-type: none"> <li>8.9.0.0 - Voucher Redemption Interface and Waitlist reservations</li> </ul>
June, 2011	<ul style="list-style-type: none"> <li>8.9.0.0 - Billing Instructions</li> </ul>
August, 2011	<ul style="list-style-type: none"> <li>8.9.0.0 - Package request</li> </ul>
September, 2011	<ul style="list-style-type: none"> <li>8.9.0.3 - Added 'Continuous Rooms'</li> </ul>
December, 2011	<ul style="list-style-type: none"> <li>Added Table Reservations</li> </ul>
June, 2012	<ul style="list-style-type: none"> <li>Added Table Reservation update</li> </ul>
June, 2012	<ul style="list-style-type: none"> <li>Encrypted password explained</li> </ul>
August, 2012	<ul style="list-style-type: none"> <li>TableAvailability</li> </ul>
August, 2013	<ul style="list-style-type: none"> <li>RoomType</li> </ul>
August, 2013	<ul style="list-style-type: none"> <li>Table reservation waitlist</li> </ul>

Date	Description of Change
May, 2014	<ul style="list-style-type: none"> <li>• 8.9.5.0 - Added NDA (non-disclosure agreement) information</li> </ul>
November, 2014	<ul style="list-style-type: none"> <li>• 8.9.5.3 - Added license request for rate list</li> </ul>
February, 2015	<ul style="list-style-type: none"> <li>• 8.9.6.1 - Added Conference and Catering requests</li> </ul>
May, 2015	<ul style="list-style-type: none"> <li>• 8.9.6.2 - Added the following: <ul style="list-style-type: none"> <li>○ StandardMenuItem;</li> <li>○ MenuItem;</li> <li>○ BookingEvent to Conference Booking;</li> <li>○ BookingEvent to BookingPackage;</li> <li>○ EventFunctionSpace, EventResource and EventStandardMenu subsets to BookingEvent;</li> <li>○ NoteCategory Request;</li> <li>○ Note Request;</li> <li>○ Note subset to Booking, Event and Resources;</li> <li>○ StandardMenu</li> <li>○ MenuItem subset to EventStandardMenu;</li> <li>○ Delete operations for Resources</li> </ul> </li> </ul>
May, 2015	<ul style="list-style-type: none"> <li>• 8.9.6.2 - Applied Oracle /Micros intermediate template, corrected missing fields, general proof read and correction of numbering and formatting</li> </ul>

Date	Description of Change
May, 2016	<ul style="list-style-type: none"> <li>• 8.10.0.0 - Added profileID and profileGlobalID attribute to LinkedProfile Added examples and some business rules to documentation</li> </ul>
June, 2017	<ul style="list-style-type: none"> <li>• 8.10.2.10 – Added BedGuestNum to AccompanyingGuests of ReservationRequest-response.</li> <li>• Added BedReservation to ReservationRequest-response Added example for CheckIn BedReservation using BedGuestNum</li> </ul>
December, 2017	<ul style="list-style-type: none"> <li>• Licensing: added license table</li> <li>• 8.11.0.0 – Added Guest Anonymization</li> </ul>



---

---

# 1 Introduction

The Suite8 XML Interface is a service allowing external applications to communicate with Suite8 based on a unified abstract interface. This interface detaches the concept from a physical database structure and provides additional business logic and security when accessing data. The XML interface cannot be used for any Central Reservation or Point of Sales development.

## XML Interface and OTA XML Interface

Should you require extra programming based on the XML Interface or OTA XML Interface then the local office needs to fill out the NDA (Non-disclosure agreement) Info sheet and each company needs to sign the Non-Disclosure Agreement. Please contact your local Oracle contact person to get more information about the topic.

## XML Interface and X64 Processors

When running XML Interface on X64 processors, the following script needs to run from the command prompt:

```
csript %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET  
W3SVC/AppPools/Enable32bitAppOnWin64 true
```

**Please note:** Crystal reports do not run on X64 processors.

---

---

## 2 System Overview

### Provided functions

The XML interface provides the following functionality:

- Department lookup
- Reservation lookup (Including Financial Accounts)
- Profile Lookup
- Profile linkage
- Postings lookup
- Children Categories lookup
- Credit card types lookup
- Credit Card validation request
- System Info lookup (system date and license info)
- Availability (for WEB) retrieval
- Reservation update
- Profile Update
- Reservation Creation
- Financial Account Creation
- Profile Creation ( profile merge could be used)
- Submitting postings
- Submitting CCM booking actual data
- Groups lookup
- Group templates lookup
- Groups update and creation
- Group templates update and creation
- Building of Suite8 HTML files
- Block Info lookup (headers and details)
- Conference statuses lookup
- Conference packages lookup
- Hotel availability retrieval
- Overbooking control
- Hurdle Rate control
- Reservation cancellation
- Rate lookup

### Communication

Any operation with the XML interface is performed over TCP/IP or HTTP/HTTPS communication protocols. When using HTTP/HTTPS communication, the message must be sent using the POST method.

When a message is transmitted via HTTP/HTTPS protocol, it is possible to submit and receive a response in a packed format. This can significantly decrease bandwidth requirements (sometimes the message size can be decreased by 80-90%). Please refer to the “Communication Packing”

---

section in this document for further details, however, configuration and installation issues are not handled in this document.

## Message format

The interface uses synchronous messaging type, all XML messages must conform to the schema described below. Each message must provide the mandatory attribute ID for the request element. The same ID will be returned as attribute for the response element.

Each message must provide the mandatory attribute Version for the Suite8 element.

Request/response behavior might be different depending on the value provided.

For further explanations please refer to the description of the particular request / response type.

The response contains the used version in the request message.

The message format supports multiple requests within the same message. In this case all the responses will be returned within one message. The responses could be distinguished by an optional attribute Name.

If a message contains multiple requests for data change and at least one of them fails, none of the change requests which were in this message will be processed.

If a message contains multiple requests for data retrieval and at least one of them fails, the response code will show the error cause and no other response data will be provided.

Each message contains username and password which will be provided to a vendor when installing the system. In addition to the username and password, the request can contain the optional attribute "PasswordEncrypted". If the value in this attribute is "true", the transmitted password should be encrypted. Password encryption can be performed by using the program **FidelioV8DBPasswordEncrypt.exe**.

Different types of request could be disabled or forbidden based on the user rights granted to the specified user and the properties licenses and policies.

The interface supports multiple vendors simultaneously; the vendors should use different credentials in order to restrict the available functionality.

If the Suite8 license **CRS XML Interface One/Two Way** is used, the message header should also contain the mandatory attribute **Vendor**. If the **Standard CRS XML Interface One/Two Way** license, the string has to be "Suite8".

## Data format

In general, fields are either numeric (decimal digits '0' - '9'), monetary (this includes the decimal numeric characters, plus '-', and '.' as necessary), or alpha (all alphabetic letters).

Some schema elements contain the XML type definitions. In this event the schema definition should be considered. In other events, the data could be passed as a generic field Type or field element where no restriction is specified in the schema.

The following field types are supported:

- Strings (xs:string) Alpha characters, including all characters from 'A' - 'Z', 'a' - 'z', any common punctuation characters such as periods ('.'), commas (','), and dashes ('-'), and any characters from the extended ASCII character sets necessary to support local alphabets.
- Number (xs: integer) - Numeric characters, includes '0' - '9', the minus sign ('-') these fields always reflect integer values (no decimal positions). Leading zeros are ignored.
- Monetary characters (xs: double) – including all numeric characters and period (".") used as decimal indicators where necessary. The PMS can handle monetary fields with an implied decimal point.
- Alphanumeric characters - all characters included above as Alpha or Numeric characters.
- Date - numeric characters, formatted as DD.MM.YYYY
- Time - numeric characters, formatted as HH:MM:SS

- DateTime – DD.MM.YYYY HH:MM:SS.MM UTC±MM (time with UTC shift info)

The same rules apply for the data returned by the interface.

## Communication Packing

In order to work with packed messages there are a couple of necessary steps to be taken:

1. Notify the XML Interface that submitted data is packed.  
This is done by setting a ContentType to "V8/ZIP"
2. Notify the XML Interface that the client can receive a packed data.  
This is done by setting a custom HTML header "V8-supported" to "Zip/Msg"
3. If the client notifies an interface that it can receive a packed data, it should validate a content type of the response. If it is set to "V8/ZIP", it means that a response contains packed data.

**Note:** Submitting packed data and receiving packed data are independent and the user can decide to use only one of them.

For example when the request message is small, but the client expects to get a lot of data as a response, it can decide to submit a message as XML, but notify the XML interface to pack a response.

The following is a C# example of the procedure which submits a packed message and expects to get a packed response.

```
private XmlDocument submitRequest(string requestStr, string URL)
{
    req = WebRequest.Create(URL);
    req.Method = "POST";
    //Notify XML Interface that client can receive a packed data
    req.Headers["V8-supported"] = "Zip/Msg";
    //Notify XML Interface that submitted data is packed.
    req.ContentType = "V8/ZIP";
    StreamWriter sw = new StreamWriter(req.GetRequestStream());

    requestStr = doZip(requestStr);

    sw.Write(requestStr);
    sw.Close();

    WebResponse respStream = req.GetResponse();
    Stream resp = respStream.GetResponseStream();
    //Check if a response contains packed data.
    if (respStream.Headers["Content-type"].IndexOf("V8/ZIP") != -1)
    {
        resp = UnZip(resp);
    }

    XmlDocument doc = new XmlDocument();
    doc.Load(resp);
    return resp;
}
```

The packed data is a BASE64 encoded ZIP.

Submitted ZIP files should contain only one file with a filename "Msg".

---

The following is a C# example which prepares a message to be submitted:

```
private string doZip(string str)
{
    MemoryStream zipOut = new MemoryStream();
    string result;
    using (ZipOutputStream s = new ZipOutputStream(zipOut))
    {
        s.SetLevel(9);
        byte[] buffer = new byte[4096];

        ZipEntry entry = new ZipEntry("Msg");
        entry.DateTime = DateTime.Now;
        s.PutNextEntry(entry);

        System.Text.UTF8Encoding encoding=
            new System.Text.UTF8Encoding();
        byte[] b = encoding.GetBytes(str);
        s.Write(b, 0, b.Length);

        s.Finish();
        zipOut.Position = 0;
        result = Convert.ToBase64String(zipOut.ToArray());
    }
    return result;
}
```

## Using Translation Tables

For third party systems, it is possible to use translation tables for matching data with the Suite 8 setup and the third party system. The setup of the translation table is performed in the Suite 8 configuration. (Please refer to the Suite 8 Setup Manual).

In the XML interface messages it is possible to specify an optional attribute **TranslationTable** for the elements such as query, insert, update, command. In this case all configuration data in the request and response will be translated according to the defined setup.

The XML should in this case contain the following: "TranslationTable="PD", indicated in the following example:

```
<insert ReqType="Reservation" TranslationTable="PD">
```

'PD' is the name of the configured XML Interface Translation definition.

## General Notes

Please note that Suite8 is a very flexible system. Most of the data and codes are subject to the properties installation setup. The vendors using Suite 8 XML Interface have to be aware that the same type of data could have different codes depending on the setup. For example room types, department codes, currency codes, market codes, source codes etc. In some cases the vendors may want to setup their own conversion tables.

## Setup

- 1 For configuring Suite 8 XML Interface to use TCP/IP Sockets, open **Fidelio Server** from the Suite8 Program directory, select **Properties**, select the service **Socket Server**, click **Setup** button and enter under **Interpreter GUID** the following: **FidelioXMLInterface.DataHandler**

- 
- 2 If Suite 8 XML Interface should work via HTTP using **Microsoft Internet Information Server (IIS)**, open the program **FidelioWebAPPSetup** (usually in the Suite8 program directory) and select **Allowed Web Applications** from the **Options** Menu, click **New** and enter **FidelioXMLInterface.DataHandler** under **Object Name**, and make sure that **Allow** is selected.

## Response Codes

- OK ->Command or request successful
- CO->Posting denied because overwrite CreditLimit is not allowed
- IA->Guest related requests Invalid account
- NG->Guest not found
- GA->Profile search is ambiguous
- NP->Posting denied because NoPost is set
- UR->Not possible to process this request, this request cannot be carried out, no retry
- IR->Invalid request or unsupported request type
- IC->Invalid credentials. Wrong username or password
- AD->Access denied. Not enough user rights or trying to make batch update when it is not allowed by configuration or feature is not allowed by the license
- GF->Globalize command failed. The profile is not qualified to be globalized (not enough data)

If the response code is not OK, please check the attribute Message in the element, additional information of the cause of the error is provided under this attribute.

## Conditions definition

Following condition operators are available:

- lt = less
- le = less or equal
- gt = greater
- ge = greater or equal
- eq = equal
- neq = not equal
- like = Oracle like condition

## Licensing

- The availability of areas and messages of the xml interface is license controlled.
- Please find following matrix which shows which licenses must be active. Per area minimum one of the licenses must be active. For some areas an additional license is needed: Conference and Catering, Voucher Management, Table reservations, CCM Board Info.

**Table 1 – License table**

Area	XML IFC Basic	XML IFC Advanced	XML IFC Vari Bon Vari Table Jet Web	Web Connect	Suite 8 Central	XML IFC Hurdle Rates	Table Reserv.	Suite 8 Central Cross Reservations Availability	CRS XML One Way / Two Way	CCM	CCM Board Info	Voucher Mgt
Departments	x	x										
Reservations	x	x	x	x	x	x						
Profile	x	x	x	x	x							
Posting	x	x	x		x							
System Date	x	x										
System Info	x	x										
Availability for Web		x		x				x	x			
Children categories	x	x										
Credit Card Types	x	x										
Login (Guest Login)		x		x								
Note Category	x	x		x						x		
Note	x	x		x						x		
Html Info	x	x		x	x							

Area	XML IFC Basic	XML IFC Advanced	XML IFC Vari Bon Vari Table Jet Web	Web Connect	Suite 8 Central	XML IFC Hurdle Rates	Table Reserv.	Suite 8 Central Cross Reservations Availability	CRS XML One Way / Two Way	CCM	CCM Board Info	Voucher Mgt
Event Info Board	x	x									xx	
Activity	x	x		x								
Tasks	x	x		x								
Group Template	x	x										
Group	x	x										
Block Header		x				x		x				
Block Detail		x				x		x				
Country	x	x										
Currency List	x	x										
Conference Status List	x			x						xx		
Conference Package List	x			x						xx		
Hotel Availability		x				x		x				
Overbooking		x				x						
Market	x	x										



Area	XML IFC Basic	XML IFC Advanced	XML IFC Vari Bon Vari Table Jet Web	Web Connect	Suite 8 Central	XML IFC Hurdle Rates	Table Reserv.	Suite 8 Central Cross Reservations Availability	CRS XML One Way / Two Way	CCM	CCM Board Info	Voucher Mgt
Main Market	x	x										
RateList	x	x										
ReportList	x	x						x				
Report Parameter List	x	x						x				
Report Selection List	x	x						x				
Room Type	x	x										
StatisticAndFure		x				x						
Calendar Event	x	x		x								
Voucher Articles	x	x		x								xx
Voucher	x	x		x								xx
Billing Instructions	x	x	x		x							
Preference	x	x	x		x							
Package		x		x	x							
Table Reservations		x		x			xx					

Area	XML IFC Basic	XML IFC Advanced	XML IFC Vari Bon Vari Table Jet Web	Web Connect	Suite 8 Central	XML IFC Hurdle Rates	Table Reserv.	Suite 8 Central Cross Reservations Availability	CRS XML One Way / Two Way	CCM	CCM Board Info	Voucher Mgt
Table Availability		x		x			xx					
Conference & Catering	x	x		x						xx		

xx= Extra license which is required.

---

---

## 3 Fields available for selection by request type

### Departments

- a. DepartmentCode
- b. DepartmentName
- c. DepartmentShortDesc
- d. DepartmentType (1 = debit, 2 = tax, 3 = credit)
- e. DepartmentArrangement

### Reservation

- a. GuestFirstname
- b. GuestName
- c. Name2
- d. RoomType
- e. RoomNum
- f. GuestArrival (Arrival date)
- g. GuestDeparture (Departure date)
- h. GuestNum (Reservation Number)
- i. ReservationState - request version 1.1.0 and higher

The result can have the following definitions:

0= Reservation - future reservation or reservation which is checked in but not yet checked out

1 = Checked In (reservations checked in on the actual date (system date))

2 = Checked Out

3 = Canceled

4 = No Show

5 = Offer

- j. ReservationStatus

The result can have the following definitions:

0=tentative

1=definitive

3=not affecting availability

- k. ReservationComment1
- l. ReservationComment2
- m. CheckinDateTime
- n. CheckoutDateTime
- o. ReservationType - FA for Financial account and RES for reservation
- p. NoAvailReason (1 = Waitlist 2 = Offer 3 = Canceled 4 = No Show)
- q. ProfileID
- r. PaymentMethod – the department code associated with the payment
- s. CrsSystem - ID of the CRS system the reservation originates from
- t. CrsResNumber - Unique ID coming from the CRS System, linked to the CrsSystem in order to support multiple CRS systems
- u. Notes
- v. NoOfRooms
- w. NoOfAdults

- 
- x. RateCode
  - y. RateValue
  - z. ForeignRateValue - Rate in foreign currency
  - aa. CurrencyCode (Please check this in the Suite8 Configuration of the property, the Currency code can be different than the ISO Code)
  - bb. MarketCode
  - cc. CompanyID
  - dd. TravelAgentID
  - ee. SourceID
  - ff. BookerID
  - gg. CompanyName
  - hh. TravelAgentName
  - ii. SourceName
  - jj. BookerName
  - kk. HasShare – the value is 1 if the reservation has related share reservations, 0 if there is no share
  - ll. ReservationAttribute - Could be used in conjunction with the optional attribute "attr" which contains the required attribute type  
For Example <field name="ReservationAttribute" attr="ABC"/>  
If the attribute is not specified, all available attributes are included in the response
  - mm. Child - Could be used in conjunction with the optional attribute "attr" containing the required child type  
For Example <field name="Child" attr="JUN"/> for junior  
If the attribute is not specified, all available attributes are included in the response
  - nn. SourceCode
  - oo. ChannelCode
  - pp. AccompanyingGuests. If this field is in the reservation request, the response contains multiple field elements of "AccompanyingGuest" with information about the accompanying guests. Each element has a node value with the profile ID and the attributes "name1", "name2", "name3" and "BedGuestNum". If additional profile information is required, an explicit profile search request should be sent. Please note that multiple profiles could be retrieved using one request message.
  - qq. GroupName
  - rr. GroupID
  - ss. Guarantee
  - tt. TotalStay
  - uu. ForeignTotalStay
  - vv. BlockCode
  - ww. NetRoomRevenue – calculated based on the Revenue Forecast and the actual postings
  - xx. ReservationDetails – if this field and the detail date field exist in the request, the result contains a separate record for each day of state, allowing retrieving reservation details which might change from day to day
  - yy. DetailDate – when using ReservationDetails mode, the dates for which this detail is valid are indicated
  - zz. PartyID – ID of the party, if the reservation participates in a party reservation
  - aaa. PartyName – the name of the party, if the reservation participates in a party reservation
  - bbb. PartyReservation - If this field is in the reservation request, the response contains multiple field elements of "PartyReservation" listing the IDs of all other reservations participating in the party
  - ccc. PartyMember - If this field is present in the reservation request, the response will contain multiple field elements "PartyMember" with information of all persons attached to the

- other reservations participating in the party. Each element has as a node value the profile ID and the attributes "resID" (ID of the reservation), "name1", "name2" and "name3" – with the guest details.
- ddd. Share - If this field is in the reservation request, the response contains multiple field elements of "Share" with information of all reservations shared with this one. Each element has the node value reservation ID and the attributes "profileID" (ID of the main reservation profile), "name1", "name2" and "name3" – with the guest details.
  - eee. ResStatusPriorCXL – the Reservation status before it was cancelled (0 – tentative, 1 – definitive)
  - fff. OptionDate
  - ggg. CustomField - If this field is present in the reservation request, the response will contain multiple field elements for each defined custom field on reservations (Please note, custom field details are not supported). Each field contains an attribute "attr" with a field name. The data format depends on the defined field format. Please note: for all Date fields, DateTime format is used.
  - hhh. Guest3rdPartyID - If this exists in the request, the response contains multiple Guest3rdPartyID for every 3rd party profile ID. The attribute "attr" contains a type from the CRS system. Available only with "Third Party Suite8 Central Interface" or "CRS XML One Way" license.
  - iii. ReservationPreferences – If this field is in the request, the response will contain for each defined reservation preference an element for "ReservationPreference". Each field contains an attribute "attr" with the preference short description. The node value includes a long description. For example: <field name="ReservationPreference" attr="NSM">Non Smoking</field>
  - jjj. Policy – the short description of the Policy attached to the reservation
  - kkk. BedReservation – If this field is in the request, the response will contain the information if the reservation is valid for a room or for beds.  
The result can have the following definitions:  
0=RoomTypeReservation  
1=BedTypeReservation

The reservation search returns reservation records as well as records for financial accounts according to the selected conditions.

If a profile was already anonymized, the fields **privacyMessage** and **privacyRemoved** are included in the response.

For example:

```
<field name="privacyMessage">Profile information no longer available</field>
<field name="privacyRemoved">Yes</field>
```

## Profile

- a. ProfileID
- b. GuestFirstname
- c. GuestName
- d. ProfileType (1 - Company 2 - Individual 3 – Employee)
- e. ProfileCategory (1=Company 2=Travel Agent 3=Source 4=Individual 5=Vendors 6=Meeting Planner)
- f. Title

- g. AddressGreeting
- h. LetterGreeting
- i. Birthday
- j. City
- k. Street1
- l. Street2,
- m. Street3
- n. ZIP
- o. State
- p. CountryISO2
- q. CountryISO3
- r. Addresses – The response may contain multiple elements. Returns all addresses.
  - a. Attr. Address Type
  - b. addData. Address Id.
  - c. 1 primary 0 non primary.

In order to query the values of fields like Street, city, a combination of fieldname Addresses and the fieldname(City,Street1 etc..) needs to be used. In response each field name will be grouped by means of the attribute addData which contains the address id.

Please note that for those field names (City, Street1 etc.) also the value of the primary address is returned.

Example:

```

<field name="Addresses"/>

returns

<fields>
<field name="PrimaryAddr" attr="BIL" addData="3449">1</field>
<field name="PrimaryAddr" attr="ADR" addData="3450">0</field>
</fields>

<field name="Addresses"/>
<field name="Street1"/>
<field name="City"/>

returns

<fields>
<field name="PrimaryAddr" attr="BIL" addData="3449">111</field>
<field name="Street1" attr="BIL" addData="3449">Main Street</field>
<field name="City" attr="BIL" addData="3449">Anytown</field>
<field name="PrimaryAddr" attr="ADR" addData="3450">0</field>
<field name="Street1" attr="ADR" addData="3450">Main Street</field>
<field name="City" attr="ADR" addData="3450">Anytown</field>
<field name="City" attr="ADR" addData="3450">Anytown</field>
<field name="Street1">Main Street</field>
</fields>

```

- s. Telephone – Returns value of primary telephone
- t. Email – Returns value primary email.
- u. Fax – Returns value primary fax.
- v. Communication – Returns value of primary Home Page.
- w. Communications – The response may contain multiple elements. The field name Communications needs to be used in combination with the field name of the communication role. ( Email-Telephone-Fax-Communication)
  - a. attr. Communication type code. Eg 'EBD' email business direct
  - b. addData. Communication ID.

- c. Primary: defines if this communication is primary for this category.

Example:

Query all email, telephone and home page communications:

```
<field name="Communications"/>
<field name="Email"/>
<field name="Telephone"/>
<field name="Communication"/>

returns

<fields>
<field name="Email">name@example.com</field>
<field name="Telephone">1234567890</field>
<field name="Communication" attr="WEB" addData="2288"
Primary="1">www.example.com</field>
<field name="Email" attr="EBD" addData="2283"
Primary="0">name@example.com</field>
<field name="Telephone" attr="PBM" addData="2284"
Primary="1">1234567890</field>
<field name="Email" attr="EBD" addData="2285"
Primary="0">name@example.com</field>
<field name="Email" attr="EPR" addData="2286"
Primary="1">name@example.com</field>
</fields>
```

Query value of primary email

```
<field name="Email"/>

<field name="Email">name@example.com</field>
```

For further details on communication types, please refer to the Suite8 Setup manual.

- x. Gender (M – male, F – Female)
- y. DeleteCode (0 –profile is active 1 – profile is not active)
- z. NationalityISO2
- aa. NationalityISO3
- bb. Language
- cc. VipCode
- dd. Notes
- ee. NotesByType.Only used for hotel connector. Please use NotesList
- ff. NotesList: The response may contain multiple elements.
  - a. attr: Note category
  - b. addData: 0 not guest viewable. 1 Guest Viewable
  - c. noteID: Unique id.

Example:

```
<field name="Notes" attr="PRF" addData="0" noteID="1476" >NewGuest
note</field>
```

- gg. ProfileGlobalID
- hh. CentralSyncTime
- ii. CreditStatus
- jj. CreditReason
- kk. CreditStatusChangeTime

- 
- ll. LinkedProfile. If this field is included in the request, the response will contain a list of the linked profiles. The response may contain multiple elements.
    - a. Attr: Linked profile category.
    - b. addData : Profile type
    - c. profID (obsolete)
    - d. profileID
    - e. profGlobalID(obsolete)
    - f. profileGlobalID
    - g. mailing
    - h. primary
    - i. name2
    - j. name3

Example:

```
<field name="LinkedProfile" attr="SPO" addData="2"
profleID="30406" profleGlobalID="" mailing="0" primary="0"
name2="" name3="Test">Test</field>
```
  - mm. BirthPlace
  - nn. Department
  - oo. Position
  - pp. ProfileExternalID
  - qq. Membership – the list of attached membership cards
    - a. Membership Card
    - b. LevelNumber
    - c. Type
    - d. Role
    - e. StartDate
    - f. ExpiryDate
    - g. SignupDate
    - h. Inactive
    - i. InactiveReason
    - j. TaxOffice
  - rr. Guest3rdPartyID - If this exists in the request, the response contains for each 3<sup>rd</sup> partyprofile ID, the element “Guest3rdPartyID “. The attribute “attr” contains the type of the CRS system. Available only with the licenses “Third Party Suite8 Central Interface” or “CRS XML One Way”.
  - ss. ProfileAttribute - If this exists in the request, the response contains for each available attribute, the element “ProfileAttribute”. Each element contains the attribute “attr” with the profile attribute type. For example <field name="ProfileAttribute" attr="INF">EA</field>
  - tt. LastUpdateTime – Time stamp when the profile was updated
  - uu. CreationTime - Time stamp when the profile was created
  - vv. CustomField - If this field is present in the request, the response contains for each defined custom field on profiles, the element “CustomField”, each field contains the attribute “attr” with the field name. The data format depends on the defined field format, all date fields use DateTime format.
  - ww. ProfilePreferences – If this field is present in the request, the response will contain multiple elements “ProfilePreference” for every defined profile preference. Every field contains an attribute “attr” with a preference short description. The node value includes a long description. For example:
 

```
<field name="ProfilePreference" attr="NSM">Non Smoking</field>
```
  - xx. NoMailing - returned values 1/0
  - yy. NoEMailing - returned values 1/0



- zz. GuestComment – If this field is in the request, the response contains for each guest comment the element “GuestComment”.  
Each element contains the attribute “attr” with the comment type, the attribute “addData” with the comment ID (the comment ID can be used for update) and the attribute “Date” with the comment date. For Example  

```
<field name="GuestComment" addData="1216" attr="COM"
Date="20.11.2014">This is my very important comment</field>
```
- aaa. IncludeNotCommitted - If this field with condition operation="eq">1< is in the request, the response contains information about the anonymization status. For Example  

```
<field name="privacyMessage">Profile information no longer
available</field>
<field name="privacyRemoved">Yes</field>
```

## Posting

- a. PostingID
- b. ProfileID
- c. GuestNum
- d. DepartmentCode
- e. DepartmentType (1 = debit, 2 = tax, 3 = credit)
- f. PostingPrice
- g. PostingQuantity
- h. PostingDescription
- i. PostingDate
- j. CurrencyCode (Please note, the Currency Code can be different form the ISO Code, please refer to the Suite8 Setup of the property)
- k. PostingProfileOrigID
- l. PostingGuestNumOrig
- m. PostingComment
- n. PostingCheckNr
- o. PostingForeignPrice

## SystemDate

- a. SystemDate

## SystemInfo

- a. LicenseName
- b. LicenseNoOfRooms
- c. Version – Program version

## AvailabilityForWeb

- a. RateCode
- b. RoomType
- c. RoomTypeDesc
- d. RoomTypeLongDesc
- e. NoOfRooms
- f. Date
- g. Price
- h. Availability
- i. RateInfo
- j. CurrencyCode

- 
- k. CurrencyISO3
  - l. BaseCurrencyPrice
  - m. DepartmentCode
  - n. RateWebInfo
  - o. PackagePrice – indicates the price of included packages
  - p. AdultPrice – if “PricePerPerson” was specified in the request, the price per person
  - q. Child – if “PricePerPerson” was specified in the request, this field contains the attribute “attr” indicating the child age category.
  - r. ContinuousRooms – checks the room availability for the entire stay. If there is a request for continuous room assignment, the response contains 1 in the “ContinuousRooms” field, if not the result is 0.

## ChildrenCategories

- a. ChildrenCategoriesCode
- b. ChildrenCategoriesName
- c. ChildrenCategoriesFromAge
- d. ChildrenCategoriesToAge

## CreditCardTypes

- a. CreditCardCode
- b. CreditCardName
- c.

## Login (UserLogin)

The Request requires at least one of the following licenses:

- a. ProfileID
- b. GuestName

## NoteCategory

The Note Category request retrieves the configured note categories. This information is needed when using the Note insert request where the fields **NoteCategoryTableLinkedTo** and **NoteCategoryCode** or **NoteCategoryID** have to be supplied. Please refer to the Note request.

- a. NoteCategoryID
- b. NoteCategoryShortDescription  
The note category code
- c. NoteCategoryLongDescription
- d. NoteCategoryTableLinkedTo

The object types for which the notes are used:

- a. Reservations  
NoteCategoryTableLinkedTo used for Reservations request object type
- b. Profile  
NoteCategoryTableLinkedTo used for Profile request object type
- c. ConferenceBooking  
NoteCategoryTableLinkedTo used for ConferenceBooking request object type
- d. BookingEvent  
NoteCategoryTableLinkedTo used for BookingEvent request object type

- 
- e. EventResource  
EventResource is general for Conference and Catering resources. In the note insert request, one should use:
    - a. EventCateringItem for booked catering item notes
    - b. EventStandardMenu for notes to be linked to booked StandardMenus
    - c. EventFunctionSpace for notes to be linked to booked function spaces
  - f. BlockHeader  
NoteCategoryTableLinkedTo used for BlockHeader request object type  
TableReservation

## Note

The Request requires at least one of the following licenses:

### General note request.

The request enables to get notes from any object type, such as ConferenceBooking, BookingEvent, EventFunctionSpace, EventCateringItem, EventStandardMenu

- a. NoteID
- b. NoteTableLinkedTo  
Object type to which the notes are linked. Possible values are:
  - a. ConferenceBooking
  - b. BookingEvent
  - c. EventStandardMenu
  - d. EventCateringItem
  - e. EventFunctionSpace
  - f. Reservation
  - g. Profile
  - h. BlockHeader
  - i. TableReservation
- c. NoteTableID  
The Id of the object for which the notes are requested
- d. NoteDescription
- e. NoteCategoryCode
- f. NoteCategoryID
- g. NoteGuestViewable  
Defines if the note can be viewed by the customer (1) or for internal use (0)

## HtmlInfo

- a. HtmlName
- b. HtmlId
- c. HtmlContent

## Event Info Board

- a. Eventid
- b. EventName
- c. EventStartDate
- d. EventStartTime
- e. EventEndDate
- f. EventEndTime
- g. EventInfoBoard
- h. EventProfileName

- 
- i. EventRoomName
  - j. EventRoomId
  - k. EventPax
  - l. EventStatus
  - m. TimeStamp

## Activity

- a. ActivityID
- b. ActivityType
- c. ActivityTypeID
- d. ActivityDescription
- e. ActivityStartDate
- f. ActivityStartTime
- g. ActivityEndDate
- h. ActivityEndTime
- i. ActivityAllDay
- j. ActivityShowas
- k. ActivityReminder
- l. ActivityPriority
- m. ActivityCompletedPercentage
- n. ActivityCompletedDate
- o. ActivityNotes
- p. ActivityNotesID
- q. ActivityCompanyName
- r. ActivityContactName
- s. ActivityMangerName
- t. ActivityManagerID
- u. ActivityCompanyID
- v. ActivityContactID
- w. ActivityStatus
- x. MaintenanceManagerID
- y. MaintenanceEstimatedDurationMin

## Task

- a. TaskID
- b. TaskType
- c. TaskTypeID
- d. TaskDescription
- e. TaskStartDate
- f. TaskStartTime
- g. TaskEndDate
- h. TaskEndTime
- i. TaskAllDay
- j. TaskShowas
- k. TaskReminder
- l. TaskPriority
- m. TaskCompletedPercentage
- n. TaskCompletedDate
- o. TaskNotes
- p. TaskNotesID

- 
- q. TaskCompanyName
  - r. TaskContactName
  - s. TaskMangerName
  - t. TaskManagerID
  - u. TaskCompanyID
  - v. TaskContactID

## Group Template

- a. GroupTemplateID
- b. GroupTemplateName
- c. GroupTemplateContactID
- d. GroupTemplateComment
- e. GroupTemplateInactive
- f. CountryISO2
- g. CountryISO3
- h. Language
- i. State

## Group

- a. GroupID
- b. GroupName
- c. GroupContactID
- d. GroupFrom
- e. GroupTo
- f. GroupTemplate
- g. GroupTemplateID
- h. GroupInactive
- i. GroupCompanyName
- j. GroupCompanyID
- k. GroupTravelAgentName
- l. GroupTravelAgentID
- m. GroupSourceName
- n. GroupSourceID
- o. GroupIndividualName
- p. GroupIndividualID
- q. GroupVendorName
- r. GroupVendorID
- s. GroupMeetingPlannerName
- t. GroupMeetingPlannerID
- u. GroupComment
- v. CountryISO2
- w. CountryISO3
- x. Language
- y. State

## BlockHeader

- a. BlockHeaderID
- b. BlockHeaderCode
- c. BlockHeaderName
- d. BlockHeaderFrom

- 
- e. BlockHeaderTo
  - f. BlockHeaderStatusCode
  - g. BlockHeaderStatus
  - h. BlockHeaderReservationStatus - 0=tentative 1=definitive 3=not affecting availability
  - i. BlockHeaderRate
  - j. BlockHeaderElastic
  - k. BlockHeaderMarket
  - l. BlockHeaderSource
  - m. BlockHeaderChanel
  - n. BlockHeaderPackage – to retrieve a list of attached packages

## **BlockDetail**

- a. BlockDetailID
- b. BlockDetailDate
- c. BlockDetailRoomType
- d. BlockDetailCutOffDate
- e. BlockDetailInitNoOfRooms
- f. BlockDetailActualNoOfRooms
- g. BlockDetailPickedUp
- h. BlockDetailCancelled
- i. BlockDetailRateValue (for forecast only)
- j. BlockDetailNetRoomRevenue (based on forecast only)

## **Country**

- a. CountryISO2
- b. CountryISO3
- c. CountryShortdesc
- d. CountryName
- e. CountryNationality

## **CurrencyList**

- a. CurrencyShortdesc
- b. CurrencyName
- c. CurrencyISO3
- d. CurrencyDigits
- e. CurrencyDepartment

## **ConferenceStatusList**

- a. ConferenceStatusId
- b. ConferenceStatusShortdesc
- c. ConferenceStatusLongdesc
- d. ConferenceStatusDisplorder
- e. ConferenceStatusWebname

## **ConferencePackageList**

- a. ConferencePackageID
- b. ConferencePackageName
- c. ConferencePackagePrice
- d. ConferencePackageValidFrom

- e. ConferencePackageValidTo

## HotelAvailability

The Request requires at least one of the following licenses:

- XML Interface Enhanced Edition
- Suite8 Central Cross Reservations and Availability
- Hurdle Rates XML Interface
  - a. Date
  - b. RoomType
  - c. AvlDefinitive – definite Availability
  - d. AvlInclTentative – Availability including tentative
  - e. Overbooking
  - f. OccDefinitive – definite Occupancy
  - g. OccTentative – tentative Occupancy
  - h. OOO – Out of order rooms
  - i. OOS – Out of service rooms
  - j. Capacity - number of existing rooms per room type. The number might differ per dates, depending on the expiration date defined on the room configuration in Suite8.
  - k. AvlDefinitiveWithBookingPosition
  - l. AvlInclTentativeWithBookingPosition
  - m. OccDefinitiveWithBookingPosition
  - n. OccTentativeWithBookingPosition
  - o. NoOfPersonStd - standard number of persons (adult + child) allowed for this room type
  - p. NoOfPersonMax – maximum number of persons (adult + child) allowed for this room type

## Overbooking

- a. RoomType
- b. OverbookingDate
- c. OverbookingType - 1 = House Overbooking 2 = Room Type Overbooking
- d. OverbookingNoOfRooms - Number of Rooms for this overbooking

## Market

- a. MarketID
- b. MarketShortdesc
- c. MarketName
- d. MarketMainMarket – Short Description

## MainMarket

MainMarketID

- a. MainMarketShortdesc
- b. MainMarketName

## RateList

- a. RateCode
- b. RateName
- c. RateDepartment
- d. RateWeekendDepartment
- e. RateWebInfo

- 
- f. RateInfo
  - g. RateFolioText
  - h. RateStartSell
  - i. RateEndSell
  - j. RateMinLOS
  - k. RateMaxLOS
  - l. RatePublic
  - m. RateWebEnabled
  - n. RateBAR

## ReportList

- a. ReportID
- b. ReportSectionID
- c. ReportName
- d. ReportSectionName
- e. ReportSecurityLevel
- f. ReportCopies
- g. ReportSecondaryCopies
- h. ReportTitle
- i. ReportExternalID
- j. ReportType - the report type:
  - 0 =Crystal Report
  - 1 =Internal Editor RTF
  - 2 =OLAP Cube
  - 3 =HTML Report

## ReportParameterList

- a. ReportID
- b. ReportSectionID
- c. ReportName
- d. ReportParameterName
- e. ReportParameterDescription
- f. ReportParameterType – the report parameter
  - 0 = string
  - 1 = integer
  - 2 = date ( DD.MM.YYYY)
  - 3 = time ( HH:MM:SS)
  - 4 = date-time (DD.MM.YYYY HH:MM:SS.mmm UTC±MM)
  - 5 = Lookup
  - 6 = Multiselect lookup
  - 7 = Boolean
- g. ReportParameterLookupQuery
- h. ReportParameterLookupField
- i. ReportParameterLookupShowField
- j. ReportParameterLookupShowField2
- k. ReportParameterMandatory
- l. ReportParameterDisplayOrder
- m. ReportParameterInvisible
- n. ReportParameterShowSystemParam



- o. ReportParameterDefault – If the date or time format is used, this field contains the value entered in Suite8, such as TODAY, BEGINNINGOFTHISWEEK ). Please refer to the Suite8 Report Manual for further information.
- p. ReportParameterDefaultValue – contains a calculated value based on the definition (such as today's date or the date when the next week begins)

## ReportSectionList

- a. ReportSectionID
- b. ReportSectionName
- c. ReportSectionParentID - (refers to the tree structure)
- d. ReportSectionReportType - the following values are available:
  - 0 =Crystal Report
  - 1 =Internal Editor RTF
  - 2 =OLAP Cube
  - 3 =HTML Report
- e. ReportSectionPrintJob
- f. ReportSectionRole – defines the report section role and can be:
  - 1=Confirmation Letter
  - 2=Folio Layout
  - 3=Message Print
  - 4=Shift report
  - 5=Conference Confirmation
  - 6=Registration Card
  - 7=Currency Exchange
  - 8=AR Reminders
  - 9=Paid Outs
  - 10=Activities
- g. ReportSectionSecurityLevel

## RoomType

- a. RoomTypeShortDesc
- b. RoomTypeLongDesc
- c. RoomTypeDefaultPAX
- d. RoomTypeYieldable
- e. RoomTypeNoOfBeds
- f. RoomTypeExtraBeds
- g. RoomTypeTreshold
- h. RoomTypeWebInfo
- i. RoomTypeSuite
- j. RoomTypeMaxPAX
- k. RoomTypeMaxToSell
- l. RoomTypeMaxToSell
- m. RoomTypeDefaultRate
- n. RoomTypeLongWebInfo
- o. RoomTypeShortMobileWebInfo
- p. RoomTypeLongMobileWebInfo
- q. RoomTypeImage- (can be multiple please refer to the Suite8 Setup Manual for further information about Room Type images)

---

## StatisticAndFuture

- a. Date
- b. HotelName
- c. GuestNum – the reservation ID
- d. ActArrivalTime
- e. ActDepartureTime
- f. ExpArrivalTime – DateTime format
- g. ExpDepartureTime – DateTime format
- h. CreationTime – DateTime format, indicates the time when the reservation was created.
- i. ResStatus - 0=tentative 1=definitive 3=not effecting availability
- j. NoAvailReason - 1 = Waitlist 2 = Offer 3 = Canceled 4 = No Show
- k. ProfileID
- l. GuestName
- m. CompanyID
- n. CompanyName
- o. TravelAgentID
- p. TravelAgentName
- q. SourceID
- r. SourceName
- s. IndividualID
- t. IndividualName
- u. VendorsID
- v. VendorsName
- w. MeetingPlannerID
- x. MeetingPlannerName
- y. GroupID
- z. GroupName
- aa. BlockHeaderID
- bb. BlockHeaderCode
- cc. RoomNum
- dd. RoomType
- ee. RateCode
- ff. MarketCode
- gg. SourceCode
- hh. ChanelCode
- ii. NoOfAdults
- jj. ShareNum
- kk. TotalNoOfChildren
- ll. GrossRoomRev
- mm. NetRoomRev
- nn. GrossFBRev
- oo. NetFBRev
- pp. GrossMinibarRev
- qq. NetMinibarRev
- rr. GrossTeleRev
- ss. NetTeleRev
- tt. GrossMiscRev
- uu. NetMiscRev
- vv. NetRev
- ww. GrossRev
- xx. IsArrival – 1 if this is the arrival

- 
- yy. IsDeparture – 1 if this the departure
  - zz. NoOfRooms
  - aaa.CountryISO2
  - bbb. CountryISO3

## CalendarEvent

- a. CalendarEventLongDesc
- b. CalendarEventNotes
- c. CalendarEventFromDate
- d. CalendarEventToDate
- e. CalendarEventTypeShortDesc
- f. CalendarEventTypeLongDesc
- g. CalendarEventType – the type of the event:
  - 0 = miscellaneous event
  - 1 = availability event
  - 2 = public holiday event
  - 3= conference booking
- h. CalendarEventTypeShow – indicates where the Event Type is displayed in Suite8:
  - 0-show all;
  - 1-show diary;
  - 2-show F4;
  - 3-show CTRL-D;
  - 4-show diary+F4;
  - 5-show diary+CTRL-D;
  - 6-show F4+CTRL-D
- i. CalendarEventTypeWebEnabled
- j. CalendarEventWebEnabled
- k. CalendarEventShortWebInfo
- l. CalendarEventLongWebInfo
- m. CalendarEventShortMobileWebInfo
- n. CalendarEventLongMobileWebInfo
- o. CalendarEventImage

## VoucherArticles

- a. VoucherArticleID
- b. VoucherArticleName
- c. VoucherArticleText
- d. VoucherArticleDescription
- e. VoucherArticleWebName
- f. VoucherArticleWebShortDescription
- g. VoucherArticleWebLongDescription
- h. VoucherArticlePaymentDepartment
- i. VoucherArticlePrice
- j. VoucherArticleDefaultDuration
- k. VoucherArticleType 0 - Product voucher; 1 - Value Voucher; 2 - Arrangement voucher
- l. VoucherArticleImageID
- m. VoucherArticleImageTime
- n. VoucherArticleWebEnabled
- o. VoucherReservationTemplate
- p. VoucherWebCustomizable

---

## Voucher

- a. VoucherID
- b. VoucherArticleID
- c. VoucherProfileID
- d. VoucherNumber
- e. VoucherDescription
- f. VoucherText
- g. VoucherValidFrom
- h. VoucherValidUntil
- i. VoucherAmount
- j. VoucherAmountRedeemed
- k. VoucherFA
- l. VoucherPrinted
- m. VoucherFullyPaid
- n. VoucherType - 0 - Product voucher; 1 - Value Voucher; 2 - Arrangement voucher
- o. VoucherReservationID
- p. VoucherFinalized
- q. VoucherDedication
- r. VoucherPickupReservationID
- s. VoucherConsumingGuestID
- t. VoucherInternalInfo
- u. VoucherOrderReason
- v. VoucherOnHoldReason
- w. VoucherToBePaid
- x. VoucherIssueDate
- y. VoucherClearingDate
- z. VoucherProfileName
- aa. VoucherStatus – the following statuses are available:
  - 1=Valid
  - 2=Not yet valid
  - 3=Expired
  - 4=On Hold
  - 5= Inactive
  - 6=Redeemed
  - 7=Not yet paid

## BillingInstruction

- a. BillingInstructionID
- b. BillingInstructionFromDate
- c. BillingInstructionToDate
- d. BillingInstructionReservationID – the reservation id for the linked billing instruction items
- e. BillingInstructionToReservationID – the reservation the billing instruction routes to
- f. BillingInstructionFAID – the financial account, the billing instruction belongs to
- g. BillingInstructionToFAID - the financial account, the billing instruction is routed to
- h. BillingInstructionWindow
- i. BillingInstructionProfileID
- j. BillingInstructionBookingID - the booking the billing instruction is linked to
- k. BillingInstructionBlockID – the block the billing instruction belongs to
- l. BillingInstructionNote

- 
- m. BillingInstructionAuto – indicates if the billing instruction is created automatically by the reservation
  - n. BillingInstructionVideoCheckout
  - o. BillingInstructionBillView
  - p. BillingInstructionDepartment – for each included department the element is present. Each element contains the attribute “attr” with the following values:
    - 1 = include
    - 2 = exclude
    - 3 = all postings included (for Master Share)For example:  
<field name="BillingInstructionDepartment" attr="1">260</field>
  - q. BillingInstructionGroup – the element is present multiple times for each included group. Each element contains the attribute “attr” with the following values:
    - 1 = include
    - 2 = exclude
    - 3 = all postings included (for Master Share)For example:  
<field name="BillingInstructionGroup" attr="1">AL</field>

## Preference

- a. PreferenceID
- b. PreferenceShortdesc
- c. PreferenceName

## Package

- a. PackageCode
- b. PackageName
- c. PackageWebDescription
- d. PackageLongWebDescription
- e. PackageWebMobileDescription
- f. PackageLongWebMobileDescription
- g. PackageFrequencyCode
- h. PackageFrequencyName
- i. PackageFrequencyWebName
- j. PackageFrequencyWebMobileName
- k. PackageCurrency
- l. PackageStart
- m. PackageEnd
- n. PackagePrice
- o. PackagePriceType – the package price type:
  - 0=child
  - 1=adult
  - 2=flat
- p. PackageIncluded
- q. PackageSinglePerRes
- r. PackageShowInRes – indicates if the package is displayed in reservations:
  - 1=true
  - 0=not shown
- s. PackagePercentage – indicates if the package is a percentage package:
  - 1= true
- t. PackageDisplOrder

- 
- u. PackagePriceChildCategory – the child category for all records using “PackagePriceType = 0”
  - v. PackageImage – the linked image

## TableReservation

- a. TableReservationID
- b. TableReservationDate
- c. TableReservationStartTime
- d. TableReservationEndTime
- e. TableReservationAdults
- f. TableReservationChildren
- g. TableReservationRestaurant
- h. TableReservationRestaurantID
- i. TableReservationAllTables –comma separated list of all tables
- j. TableReservationGuests – semicolon separated list of all guests
- k. TableReservationProfileID
- l. GuestFirstname
- m. GuestName
- n. VipCode

If more than one profile is attached, the response may contain multiple elements for ‘GuestFirstname’, ‘GuestName’ and ‘VipCode’. All these elements have the attribute ‘attr’ containing the ID of the corresponding profile.

## TableAvailability

The following search criteria are available:

- a. RestaurantOccupiedTables – How many tables are occupied
- b. RestaurantFreeTables – How many tables are free
- c. RestaurantOccupiedPersons – Reservation for this amount of persons already exist
- d. RestaurantAvailStdPersons – Availability based on the standard number of persons
- e. RestaurantAvailMaxPersons – Availability based on the maximum number of persons
- f. RestaurantWebDescription – the short Web description of the restaurant
- g. RestaurantLongWebDescription – the Long Web description of the restaurant
- h. RestaurantWebMobileDescription – the Short mobile Web description of the restaurant
- i. RestaurantLongWebMobileDescription – the Long mobile Web description of the restaurant
- j. RestaurantID
- k. RestaurantShortDescription – the Short description of the restaurant
- l. RestaurantDefDuration – the configured default reservation duration
- m. RestaurantImage – (can be multiple) the information about the restaurant images

## AccountManager

- a. AccountManagerID
- b. AccountManagerShortDescription
- c. AccountManagerLongDescription

## ConferenceManager

- a. ConferenceManagerID
- b. ConferenceManagerShortDescription

- 
- c. ConferenceManagerLongDescription

## Channel

- a. ChannelID
- b. ChannelCode
- c. ChannelName

## Source

- a. SourceID
- b. SourceCode
- c. SourceName

## ConferenceStatusList

- a. StatusID
- b. StatusShortDesc
- c. StatusLongDesc
- d. StatusWebEnabled
- e. StatusDisplayOrder
- f. StatusWebName

## FunctionType

- a. FunctionTypeID
- b. FunctionTypeShortDescription
- c. FunctionTypeLongDescription
- d. FunctionTypeShortWebInfo
- e. FunctionTypeLongWebInfo
- f. FunctionTypeWebEnabled

## SeatingArrangements

- a. SeatingID
- b. SeatingShortDesc
- c. SeatingLongDesc
- d. SeatingDisplayOrder
- e. SeatingWebEnabled

## EventType

- a. EventTypeID
- b. EventTypeShortDescription
- c. EventTypeLongDescription
- d. EventTypeShortWebInfo
- e. EventTypeLongWebInfo
- f. EventTypeWebEnabled

## EventStatus

- a. EventStatusID
- b. EventStatusShortDescription
- c. EventStatusLongDescription
- d. EventStatusShortWebInfo

- 
- e. EventStatusLongWebInfo
  - f. EventStatusWebEnabled

## FunctionSpace

- a. FunctionSpaceID
- b. FunctionSpaceShortDescription
- c. FunctionSpaceLongDescription
- d. FunctionSpaceShortWebInfo
- e. FunctionSpaceLongWebInfo
- f. FunctionSpaceWebEnabled  
Is function space is web enabled. 0 false, 1 trueg.
- g. FunctionSpaceTypeShortDescription
- h. FunctionSpaceTypeLongDescription
- i. FunctionSpaceShareable  
Is function space is web shareable. 0 false, 1 true
- j. FunctionSpaceCombined  
Is function space is a combination room. 0 false, 1 true
- k. FunctionSpaceMinCapacity
- l. FunctionSpaceMaxCapacity
- m. FunctionSpaceCostPrice

## FunctionSpaceSeating

- a. FunctionSpaceSeatingFunctionSpaceID
- b. FunctionSpaceSeatingID
- c. FunctionSpaceSeatingShortDescription
- d. FunctionSpaceSeatingLongDescription
- e. FunctionSpaceSeatingShortWebInfo
- f. FunctionSpaceSeatingLongWebInfo
- g. FunctionSpaceSeatingWebEnabled  
Is seating web enabled. 0 false, 1 true
- h. FunctionSpaceSeatingMinCapacity
- i. FunctionSpaceSeatingMaxCapacity
- j. FunctionSpaceSeatingSetupTime
- k. FunctionSpaceSeatingSetdownTime
- l. FunctionSpaceSeatingDefault.  
Default function space seating for this function space. 0 false, 1 true

## FunctionSpaceRate

- a. FunctionSpaceRateFunctionSpaceID
- b. FunctionSpaceRateID
- c. FunctionSpaceRateShortDescription
- d. FunctionSpaceRateLongDescription
- e. FunctionSpaceRateShortWebInfo
- f. FunctionSpaceRateLongWebInfo
- g. FunctionSpaceRateWebEnabled  
defines if the rate is web enabled. 0 false, 1 true
- h. FunctionSpaceRatePerHour  
defines if the rate is per hour. 0 false, 1 true
- i. FunctionSpaceRateValue



- j. FunctionSpaceRateDefault  
defines if the is the default rate. 0 false, 1 true.
- k. FunctionSpaceRateValidFrom
- l. FunctionSpaceRateValidTo

## FunctionSpaceAvailability

- a. FunctionSpaceID
- b. FunctionSpaceShortDescription
- c. FunctionSpaceLongDescription
- d. FunctionSpaceWebEnabled  
defines if the function space is web enabled. 0 false, 1 true
- e. FunctionSpaceShortWebInfo
- f. FunctionSpaceLongWebInfo
- g. FunctionSpaceTypeShortDescription
- h. FunctionSpaceTypeLongDescription
- i. FunctionSpaceShareable  
defines if the function space is shareable:  
0= false  
1= true
- j. FunctionSpaceCombined  
defines if the function space is a combination room:  
0= false  
1= true
- k. FunctionSpaceMinCapacity
- l. FunctionSpaceMaxCapacity
- m. FunctionSpaceCostPrice
- n. FunctionSpaceRate.  
The Function space rates list. The response may contain multiple elements.The rates are being displayed as attributes.  
See also FunctionSpaceRate Request
  - a. ID  
The Id of the rate configuration record
  - b. FunctionSpaceID
  - c. ShortDescription
  - d. LongDescription
  - e. WebEnabled
  - f. ShortWebInfo
  - g. LongWebInfo
  - h. PerHour
  - i. Value
  - j. Default
  - k. ValidFrom
  - l. ValidTo  
For Example:  

```
<field name="FunctionSpaceRate" ID="100"
FunctionSpaceID="1008" ShortDescription="FUL"
LongDescription="full day" WebEnabled="1" ShortWebInfo=""
LongWebInfo="" PerHour="1" Value="1200" Default="1"
ValidFrom="20.02.2012" ValidTo="18.02.2021" CostPrice="0"/>
```
- o. FunctionSpaceSeating  
the Function space seating list. The response may contain multiple elements. Seatings are

---

displayed as attributes

See also FunctionSpaceSeating Request

- a. ID  
the Id of the seating configuration record
- b. FunctionSpaceID
- c. ShortDescription
- d. LongDescription
- e. WebEnabled
- f. ShortWebInfo
- g. LongWebInfo
- h. MinCapacity
- i. MaxCapacity
- j. SetupTime
- k. SetdownTime
- l. Default

For Example:

```
<field name="FunctionSpaceSeating" ID="1127"  
FunctionSpaceID="1012" ShortDescription="TOP"  
LongDescription="Top Tables" WebEnabled="1"  
ShortWebInfo="Short web info " LongWebInfo=""  
MinCapacity="10" MaxCapacity="300" SetupTime="0"  
SetdownTime="0" Default="1"/>
```

## ConferencePackagesList

- a. ConferencePackageID
- b. ConferencePackageShortDescription
- c. ConferencePackageShortWebInfo
- d. ConferencePackageLongWebInfo
- e. ConferencePackageWebEnabled  
Defines if the ConferencePackage is web enabled. 0 false, 1 true.
- f. ConferencePackageValidFrom
- g. ConferencePackageValidTo
- h. ConferencePackageRateValue
- i. ConferencePackageEvents  
the List of the linked events to the package. The response may contain multiple elements.  
Events are displayed as attributes
  - a. ID  
the Id of the event configuration record
  - b. ConferencePackageID
  - c. TypeID  
the Id of the event type.
  - d. TypeShortDescription  
The Event type short description
  - e. StartTime
  - f. EndTime
  - g. FromDay  
Starting day = 1
  - h. ToDay  
the Length of the package. 1= one day, 2= 2 days
  - i. WebEnabled  
For Example:

```

<field name="ConferencePackageEvents" ID="1008"
ConferencePackageID="1006" TypeID="117"
TypeShortDescription="MET" StartTime="09:00:00"
EndTime="18:00:00" FromDay="1" ToDay="1" WebEnabled="1"/>
<field name="ConferencePackageEvents" ID="1009"
ConferencePackageID="1006" TypeID="114"
TypeShortDescription="CBM" StartTime="10:00:00"
EndTime="10:30:00" FromDay="1" ToDay="1" WebEnabled="1"/>

```

## CateringItem

- a. CateringItemID
- b. CateringItemLongDescription  
Catering Item Name
- c. CateringItemDescription  
Additional description.
- d. CateringItemShortWebInfo
- e. CateringItemLongWebInfo
- f. CateringItemWebEnabled  
defines if the catering item is web enabled. 0 false, 1 true.
- g. CateringItemDescription
- h. CateringItemExternal
- i. CateringItemQuantity
- j. CateringItemArticleNumber
- k. CateringItemLeadDays
- l. CateringItemShowFO  
defines if the item is used in PMS. 0 false, 1 true.
- m. CateringItemShowSC  
defines if the item is used in CCM. 0 false, 1 true
- n. CateringItemTrackInventory  
defines if the Inventory is tracked on the catering item. 0 false, 1 true.
- o. CateringItemSetupTime
- p. CateringItemSetdownTime
- q. CateringItemCostPrice
- r. CateringItemRate  
the Catering item rate list. The response may contain multiple elements. Rates are displayed as attributes

See also CateringItemRate request

- a. ID  
the Id of the rate configuration record.
- b. CateringItemID
- c. ShortDescription
- d. LongDescription
- e. WebEnabled
- f. ShortWebInfo
- g. LongWebInfo
- h. PerHour
- i. Value
- j. Default
- k. ValidFrom
- l. ValidTo

For Example:

```
<field name="CateringItemRate" ID="108" CateringItemID="1027"
```

```

ShortDescription="HOU" LongDescription="" WebEnabled="1"
ShortWebInfo="" LongWebInfo="" PerHour="1" Value="30"
Default="0" ValidFrom="20.02.2012" ValidTo="18.02.2021"/>
<field name="CateringItemRate" ID="101" CateringItemID="1027"
ShortDescription="HAL" LongDescription="" WebEnabled="1"
ShortWebInfo="" LongWebInfo="" PerHour="0" Value="100"
Default="0" ValidFrom="20.02.2012" ValidTo="18.02.2021"/>

```

## CateringItemRate

- a. CateringItemRateID
- b. CateringItemRateCateringItemID
- c. CateringItemRateShortDescription
- d. CateringItemRateLongDescription
- e. CateringItemRateShortWebInfo
- f. CateringItemRateLongWebInfo
- g. CateringItemRateWebEnabled  
Is rate web enabled. 0 false, 1 true
- h. CateringItemRatePerHour  
Is rate an hourly rate. 0 false, 1 true.
- i. CateringItemRateDefault  
Is this rate default. 0 false, 1 true.
- j. CateringItemRateValue
- k. CateringItemRateValidFrom
- l. CateringItemRateValidTo

## StandardMenu

- a. StandardMenuID
- b. StandardMenuLongDescription  
the name of standard menu
- c. StandardMenuDescription  
the additional description.
- d. StandardMenuShortWebInfo
- e. StandardMenuLongWebInfo
- f. StandardMenuWebEnabled  
defines if the standard menu is web enabled. 0 false, 1 true
- g. StandardMenuMinimumAmount
- h. StandardMenuMaximumAmount
- i. StandardMenuServedPerPerson  
defines if this menu is served per person. 0 false, 1 true
- j. StandardMenuArticleNumber
- k. StandardMenuChargedOnConsumption  
defines if this menu is charged under consumption. 0 false, 1 true.
- l. StandardMenuCostPrice
- m. StandardMenuRate  
the Menu Rate list of the standard menu. The response may contain multiple elements.  
Menu rates are displayed as attributes  
See also StandardMenuRate request
  - a. ID
  - b. StandardMenuID
  - c. ShortDescription
  - d. LongDescription

- e. ShortWebInfo
- f. LongWebInfo
- g. WebEnabled
- h. ValidFrom
- i. ValidTo
- j. Value
- k. CostPrice
- l. Default

For Example:

```
<field name="StandardMenuRate" ID="1002"
StandardMenuID="1006" ShortDescription="BAR"
LongDescription="Bar" ShortWebInfo="short web for rate for
bar" LongWebInfo="Long short web for rate for bar"
WebEnabled="1" ValidFrom="30.04.2013" ValidTo="27.09.2023"
Value="35" CostPrice="15" Default="1"/>
<field name="StandardMenuRate" ID="0" StandardMenuID="1006"
ShortDescription="" LongDescription="&lt;General Rate&gt;"
ShortWebInfo="Rate short web info" LongWebInfo=""
WebEnabled="1" ValidFrom="30.04.2013" ValidTo="29.04.2023"
Value="30" CostPrice="15" Default="0"/>
```

## StandardMenuRate

- a. StandardMenuRateID
- b. StandardMenuRateStandardMenuID
- c. StandardMenuRateShortDescription  
Outlet name.
- d. StandardMenuRateLongDescription  
Returns the general rate or the outlet name if this rate is used for an outlet.
- e. StandardMenuRateShortWebInfo
- f. StandardMenuRateLongWebInfo
- g. StandardMenuRateWebEnabled  
defines if the standard menu rate is web enabled. 0 false, 1 true
- h. StandardMenuRateDefault  
defines if this rate is the default. 0 false, 1 true.
- i. StandardMenuRateValidFrom
- j. StandardMenuRateValidTo
- k. StandardMenuRateValue
- l. StandardMenuRateCostPrice

## MenuItem

- a. MenuItemID
- b. MenuItemLongDescription
- c. MenuItemShortWebInfo
- d. MenuItemLongWebInfo
- e. MenuItemWebEnabled  
defines if the menu item is web enabled. 0 false, 1 true
- f. MenuItemMinimumAmount
- g. MenuItemMaximumAmount
- h. MenuItemOrigin
- i. MenuItemArticleNumber
- j. MenuItemCostPrice
- k. MenuItemPortion

- l. MenuItemLeadDays
- m. MenuItemAvailableFrom
- n. MenuItemAvailableTo
- o. MenuItemInclMenu  
defines if the menu item is included when adding it to a standard menu  
0 false, 1 true
- p. MenuItemLongDescription2
- q. MenuItemLongDescription3
- r. MenuItemLongDescription4
- s. MenuItemOrigin2
- t. MenuItemOrigin3
- u. MenuItemVintage
- v. MenuItemDescription
- w. MenuItemRate  
the Menu item rates list. The response may contain multiple elements. The rates linked to the menu item are being displayed as attributes.

See also Menu ItemRate Request

- a. ID
- b. MenuItemID
- c. ShortDescription
- d. LongDescription  
Returns the general rate or the outlet name if the rate is used for an outlet
- e. ShortWebInfo
- f. LongWebInfo
- g. WebEnabled
- h. Value
- i. ValidFrom
- j. ValidTo
- k. Default
- l. CostPrice

For Example:

```
<field name="MenuItemRate" ID="1189" MenuItemID="1062"
ShortWebInfo="" LongWebInfo="" WebEnabled="1" Value="4"
ValidFrom="20.05.2012" ValidTo="19.05.2023" Default="1"
CostPrice="0"/>
<field name="MenuItemRate" ID="1230" MenuItemID="1062"
ShortWebInfo="Short rate info for web" LongWebInfo=""
WebEnabled="1" Value="5" ValidFrom="20.05.2012"
ValidTo="19.05.2023" Default="0" CostPrice="0"/>
```

## MenuItemRate

- a. MenuItemRateID
- b. MenuItemRateMenuItemID
- c. MenuItemRateShortDescription  
the Outlet Name
- d. MenuItemRateLongDescription  
Returns the general rate or the outlet name if this rate is used for an outlet
- e. MenuItemRate ShortWebInfo
- f. MenuItemRate LongWebInfo
- g. MenuItemRate WebEnabled
- h. MenuItemRateValue
- i. MenuItemRate RateValidFrom

- j. MenuItemRate RateValidTo
- k. MenuItemRateDefault
- l. MenuItemRateCostPrice

## ConferenceBooking

- a. Booking\_CompanyID
- b. Booking\_ContactID
- c. Booking\_ArrivalDate
- d. Booking\_DepartureDate
- e. Booking\_Name
- f. Booking\_ConferenceStatus
- g. Booking\_RoomStatus
- h. Booking\_RoomStatusID
- i. Booking\_ConferenceOnly  
defines if the booking is for conference only, 0 false, 1 true
- j. Booking\_RoomsOnly  
defines if the booking is for rooms only, 0 false, 1 true
- k. Booking\_ConferenceAndRooms  
defines if the booking is for both: conference and guest rooms, 0 false, 1 true
- l. Booking\_Notes  
returns the first note entered on the booking – not longer used, please refer to the **Note request**
- m. Booking\_NoOfAttendes
- n. Booking\_MarketCode
- o. Booking\_SourceCode
- p. Booking\_NoteCategory
- q. Booking\_ID
- r. Booking\_ContactOnSite
- s. Booking\_AccountManagerID
- t. Booking\_AccountManagerCode
- u. Booking\_MarketCodeID
- v. Booking\_SourceCodeID
- w. Booking\_ChannelCodeID
- x. Booking\_ChannelCode
- y. Booking\_ConferenceStatusID
- z. Booking\_ConferenceManagerID
- aa. Booking\_ConferenceManagerCode
- bb. Booking\_DecisionDate
- cc. Booking\_FollowupDate
- dd. Booking\_CutoffDate
- ee. Booking\_Infoboard
- ff. Booking\_FunctionTypeID
- gg. Booking\_FunctionTypeCode
- hh. Booking\_CurrencyID
- ii. Booking\_CurrencyCode
- jj. BookingEvent  
the BookingEvent list. The response may contain multiple elements. The data of the BookingEvents is displayed as attributes “attr”.  
See also BookingEvent
  - a. BookingID
  - b. ID

- c. StatusID
- d. StatusCode
- e.TypeID
- f. TypeCode
- g. EventName
- h. NumberOfAttendees
- i. StartTime
- j. EndTime
- k. Infoboard
- l. BookingPackageID

For Example:

```
<field name="BookingEvent" BookingID="1411" ID="1797"
StatusID="4" StatusCode="DEF"TypeID="117" TypeCode="MET"
EventName="Meeting" NumberOfAttendees="50"
StartTime="29.02.2016 14:30:00" EndTime="29.02.2016 17:00:00"
Infoboard=" Sales Event" BookingPackageID="0"/>
<field name="BookingEvent" BookingID="1411" ID="1799"
StatusID="4" StatusCode="DEF"TypeID="106" TypeCode="REC"
EventName="Reception" NumberOfAttendees="2"
StartTime="29.02.2016 09:00:00" EndTime="01.03.2016 22:00:00"
Infoboard=" " BookingPackageID="0"/>
<field name="BookingEvent" BookingID="1411" ID="1798"
StatusID="4" StatusCode="DEF"TypeID="117" TypeCode="MET"
EventName="Meeting" NumberOfAttendees="50"
StartTime="02.03.2016 09:00:00" EndTime="02.03.2016 12:00:00"
Infoboard="Sales Event" BookingPackageID="0"/>
```

kk. Note

the Note list. The response may contain multiple elements. The data of the notes is displayed as attributes "attr". Please refer to the **Note Request**

- a. ID
- b. CategoryID
- c. CategoryCode
- d. Description
- e. Guest Viewable
- f. TableLinkedTo

the object (table, the note is linked to)

For example:

```
<field name="Note" ID="1395" CategoryID="5"
CategoryCode="BOO" Description="This is a booking note"
TableID="1425" TableLinkedTo="ConferenceBooking"
GuestViewable="1"/>
```

## BookingPackage

Packages linked to a booking

- a. BookingPackageID
- b. BookingPackageConfigID
- c. BookingPackageConfigCode
- d. BookingPackageStartTime
- e. BookingPackageEndTime
- f. BookingPackageNumberOfAttendees
- g. BookingPackageRateValue
- h. BookingEvent

The response may contain multiple elements. The data of the linked events is displayed as attributes "attr".



- a. BookingID
- b. ID
- c. StatusID
- d. StatusCode
- e.TypeID
- f. TypeCode
- g. EventName
- h. NumberOfAttendees
- i. StartTime
- j. EndTime
- k. InfoBoard
- l. BookingPackageID

For example:

```
<field name="BookingEvent" BookingID="1425" ID="1823"
StatusID="4" StatusCode="DEF"TypeID="117" TypeCode="MET"
EventName="Meeting" NumberOfAttendees="20"
StartTime="22.02.2016 09:00:00" EndTime="22.02.2016 17:00:00"
Infoboard="" BookingPackageID="1287"/>
```

## BookingEvent

Events linked to a booking

- a. BookingEventID
- b. BookingID
- c. BookingEventStatusID
- d. BookingEventStatusCode
- e. BookingEventTypeID
- f. BookingEventTypeCode
- g. BookingEventName
- h. BookingEventNumberOfAttendees
- i. BookingEventStartTime
- j. BookingEventEndTime
- k. BookingEventInfoboard
- l. BookingEventBookingPackageID
- m. EventFunctionSpace

The EventFunctionSpace listThe response may contain multiple elements. The data of linked function spaces is displayed as attributes "attr". Please refer also to

### EventFunctionSpace

- a. ID
- b. EventID
- c. ConfigID
- d. ConfigCode
- e. ConfigSeatingID
- f. ConfigSeatingCode
- g. ConfigRateID
- h. ConfigRateCode
- i. SpaceStartTime
- j. SpaceEndTime
- k. RateValue

For Example:

```
<field name="EventFunctionSpace" ID="1759" EventID="1787"
ConfigID="1008" ConfigCode="Ball I" ConfigSeatingID="100"
ConfigSeatingCode="CLA" ConfigRateID="100"
```

---

```
ConfigRateCode="FUL" StartTime="28.02.2015 09:00:00"
EndTime="28.02.2015 12:00:00" RateValue="1200"/>
```

n. **EventCateringItem**

The EventCateringItem list. The response may contain multiple elements. The data of the EventCateringItem is displayed as attributes "attr". Please refer also to **EventCateringItem**

- a. ID
- b. EventID
- c. Name
- d. Description
- e. ConfigID
- f. ConfigName
- g. ConfigRateID
- h. ConfigRateCode
- i. StartTime
- j. EndTime
- k. Quantity
- l. RateValue

For Example:

```
<field name="EventCateringItem" ID="3140" EventID="1787"
ConfigID="1008" ConfigName="Overhead Projector"
CateringItemName="Overhead Projector" Description=""
ConfigRateID="0" ConfigRateRateCode="" StartTime="28.02.2015
09:00:00" EndTime="28.02.2015 12:00:00" Quantity="1"
RateValue="0"/>
```

o. **EventStandardMenu**

The EventStandardMenu list. The response may contain multiple elements. The data of the EventStandardMenu is displayed as attributes "attr". Please refer also to

**EventStandardMenu**

- a. ID
- b. EventID
- c. Name
- d. Description
- e. ConfigID
- f. ConfigName
- g. ConfigRateID
- h. ConfigRateCode
- i. StartTime
- j. EndTime
- k. MenuQuantity
- l. RateValue

For Example:

```
<field name="EventStandardMenu" ID="1699" EventID="1787"
ConfigID="1006" ConfigName="Menu I" Description=""
StandardMenuName="Menu I" ConfigRateID="1002"
ConfigRateCode="" StartTime="28.02.2015 09:00:00"
EndTime="28.02.2015 12:00:00" Quantity="20" RateValue="35"/>
```

p. **Note**

the Note list. The response may contain multiple elements. The data of the notes is displayed as attributes "attr".

- a. ID
- b. CategoryID
- c. CategoryCode
- d. Description

- e. TableID  
the Id of the record in the table to which the note is linked. In this example it is the booked event id
- f. TableLinkedTo  
the object to which the note is linked
- g. GuestViewable  
For Example:  

```
<field name="Note" ID="1399" CategoryID="8"
CategoryCode="EVE" Description="This note is about the event"
TableID="1823" TableLinkedTo="BookingEvent"
GuestViewable="1" />
```

## EventFunctionSpace

Booked function spaces linked to an event

- a. EventFunctionSpaceID –required only for update requests in the condition
- b. EventFunctionSpaceEventID\*
- c. EventFunctionSpaceConfigID or
- d. EventFunctionSpaceConfigCode\* - either one is mandatory, the ID or the Code
- e. EventFunctionSpaceConfigSeatingID or
- f. EventFunctionSpaceConfigSeatingCode - either one is possible, the ID or the Code
- g. EventFunctionSpaceConfigRateID or
- h. EventFunctionSpaceConfigRateCode\*- either one is mandatory, the ID or the Code
- i. EventFunctionSpaceStartTime – should be in the range of the event start and end time
- j. EventFunctionSpaceEndTime– should be in the range of the event start and end time
- k. EventFunctionSpaceRateValue – only required if different than the configured rate value, if not transmitted or blank the value of the configured Function Space Rate Value is taken
- l. Note  
the Note list. The response may contain multiple elements. The data of the notes is displayed as attributes “attr”.
  - a. ID
  - b. CategoryID
  - c. CategoryCode
  - d. Description
  - e. GuestViewable

For example:  

```
<field name="Note" CategoryID="10" CategoryCode="REN"
Description="This is a resource note for function space"
GuestViewable="1" TableLinkedTo="EventFunctionSpace" />
```

## EventCateringItem

Booked catering items linked to an event

- a. EventCateringItemID – only required for update in the condition of the update request
- b. EventCateringItemEventID\*
- c. EventCateringItemConfigID or
- d. EventCateringItemConfigCode\* -either one ID or Code is mandatory
- e. EventCateringItemName – maximum 60 characters, only required if different than the configured Catering Item Name, if this field is not transmitted or with blank value, the configured Catering Item Name is used
- f. EventCateringItemDescription
- g. EventCateringItemConfigRateID or

- h. EventCateringItemConfigRateCode\* - either one ID or Code is mandatory
- m. EventCateringItemStartTime- should be in the range of the event start and end time
- n. EventCateringItemEndTime- should be in the range of the event start and end time
- i. EventCateringItemQuantity\*
- j. EventCateringItemRateValue - only required if different than the configured Catering Item Rate Value, if this field is not transmitted or with blank value the configured Catering Item Rate Value is used
- k. Note -the Note list, the response may contain multiple elements. The data of the notes is displayed as attributes "attr".
  - a. ID- only required for update
  - b. CategoryID
  - c. CategoryCode
  - d. Description
  - e. TableID –the ID of the record in the table to which the note is linked to, in this case it is the catering item id
  - f. TableLinkedTo – the table to which the note is linked to
  - g. GuestViewable

For Example:

```
<field name="Note" ID="1400" CategoryID="10"
CategoryCode="REN" Description="This note is attached to the
function space" TableID="1787"
TableLinkedTo="EventFunctionSpace" GuestViewable="1"/>
```

## EventStandardMenu

Booked Standard Menus linked to an event

- a. EventStandardMenuID
- b. EventStandardMenuEventID
- c. EventStandardMenuName
- d. EventStandardMenuDescription
- e. EventStandardMenuConfigID
- f. EventStandardMenuConfigName
- g. EventStandardMenuConfigRateID
- h. EventStandardMenuConfigRateCode
- i. EventStandardMenuStartTime
- j. EventStandardMenuEndTime
- k. EventStandardMenuQuantity
- l. EventStandardMenuRateValue
- m. StandardMenuMenuItem -The menu items of the standard menu, the response may contain multiple elements. The data of the StandardMenuMenuItem is displayed as attributes "attr". Please refer also to **StandardMenuMenuItem**.
  - a. ID
  - b. StandardMenuID
  - c. EventID
  - d. ConfigID
  - e. ConfigName
  - f. LongDescription
  - g. LongDescription2
  - h. LongDescription3
  - i. LongDescription4
  - j. Description
  - k. Origin

- l. Origin2
- m. Origin3
- n. Vintage
- o. ConfigRateID
- p. ConfigRateCode
- q. Quantity
- r. RateValue
- s. Included
- t. Course
- u. CourseOrder
- v. Portion
- w. FoodOrBeverage
- x. CostPrice

For Example:

```
<field name="StandardMenuMenuItem" ID="3491"
StandardMenuID="1700" EventID="1790" ConfigID="1005"
ConfigName="Parfait of coffee with oranges"
LongDescription="Parfait of coffee with oranges"
LongDescription2="" LongDescription3="" LongDescription4=""
Description="" Origin="" Origin2="" Origin3="" Vintage=""
ConfigRateID="0" ConfigRateCode="" Quantity="30"
RateValue="10" Included="1" Course="3" CourseOrder="0"
Portion="" FoodOrBeverage="1" CostPrice="5"/>
<field name="StandardMenuMenuItem" ID="3492"
StandardMenuID="1700" EventID="1790" ConfigID="1007"
ConfigName="’Caesar Salad’ with Dijon-mustard
dressing" LongDescription="’Caesar Salad’ with
Dijon-mustard dressing" LongDescription2="and roasted chicken
breast" LongDescription3="Third menu item name"
LongDescription4=" menu item name 4" Description="This is the
description&#xD;&#xA;" Origin="" Origin2="" Origin3=""
Vintage="" ConfigRateID="0" ConfigRateCode="" Quantity="30"
RateValue="10" Included="1" Course="1" CourseOrder="0"
Portion="" FoodOrBeverage="1" CostPrice="5"/>
<field name="StandardMenuMenuItem" ID="3493"
StandardMenuID="1700" EventID="1790" ConfigID="1008"
ConfigName="Red Snapper with red curry and limes"
LongDescription="Red Snapper with red curry and limes"
LongDescription2="fennel and rice" LongDescription3=""
LongDescription4="" Description="" Origin="" Origin2=""
Origin3="" Vintage="" ConfigRateID="0" ConfigRateCode=""
Quantity="30" RateValue="10" Included="1" Course="2"
CourseOrder="0" Portion="" FoodOrBeverage="1" CostPrice="5"/>
<field name="StandardMenuMenuItem" ID="3495"
StandardMenuID="1700" EventID="1790" ConfigID="1137"
ConfigName="Chateau Lafleur ’78"
LongDescription="Chateau Lafleur ’78"
LongDescription2="" LongDescription3="" LongDescription4=""
Description="" Origin="" Origin2="" Origin3="Bordeaux"
Vintage="1978" ConfigRateID="0" ConfigRateCode=""
Quantity="30" RateValue="10" Included="0" Course="0"
CourseOrder="0" Portion="" FoodOrBeverage="0" CostPrice="5"/>
```

- n. Note
  - the Note list. The response may contain multiple elements. The data of the notes is displayed as attributes “attr”.
  - a. ID
  - b. CategoryID

- 
- c. CategoryCode
  - d. Description
  - e. TableID  
the Id of the record in the table to which the note is linked. In this case the booked standard menu id
  - f. TableLinkedTo  
the resource to which the note is linked. In this case the booked standard menu ID
  - g. GuestViewable  
For Example:  

```
<field name="Note" ID="1402" CategoryID="10"
CategoryCode="REN" Description="This note is attached to the
booked Standard Menu" TableID="1704"
TableLinkedTo="EventStandardMenu" GuestViewable="1"/>
```

## StandardMenuMenuItem

Booked Menu items linked to a standard menu

- a. StandardMenuMenuItemID
- b. StandardMenuMenuItemStandardMenuID
- c. StandardMenuMenuItemEventID
- d. StandardMenuMenuItemConfigID
- e. StandardMenuMenuItemConfigName
- f. StandardMenuMenuItemLongDescription
- g. StandardMenuMenuItemLongDescription2
- h. StandardMenuMenuItemLongDescription3
- i. StandardMenuMenuItemLongDescription4
- j. StandardMenuMenuItemDescription
- k. StandardMenuMenuItemOrigin
- l. StandardMenuMenuItemOrigin2
- m. StandardMenuMenuItemOrigin3
- n. StandardMenuMenuItemVintage
- o. StandardMenuMenuItemConfigRateID
- p. StandardMenuMenuItemConfigRateCode
- q. StandardMenuMenuItemQuantity
- r. StandardMenuMenuItemRateValue
- s. StandardMenuMenuItemIncluded
- t. StandardMenuMenuItemCourse
- u. StandardMenuMenuItemPortion
- v. StandardMenuMenuItemFoodOrBeverage
- w. StandardMenuMenuItemCostPrice

---

---

## 4 Available search criteria by request type

### Departments

- a. DepartmentCode
- b. DepartmentName
- c. DepartmentShortDesc

### Reservation

- a. GuestFirstname
- b. GuestName
- c. RoomType
- d. RoomNum
- e. GuestArrival (Arrival date)
- f. GuestDeparture (Departure date)
- g. GuestNum (Reservation Number)
- h. ProfileID
- i. ReservationState - request version 1.1.0 and higher  
Allowed values are:
  - 0 – Reservation - returns all future reservations and reservations which are checked in but not yet checked out
  - 1 - Checked In (checked in today reservations)
  - 2 - Checked Out (checked out today reservations)
  - 3 – Canceled
  - 4 - No Show
  - 5 - Offer
- j. ReservationStatus  
The behavior is different and depends on the request version.  
For request version prior 1.1.0 allowed values are:
  - 0 – Reservation - returns all future reservations and reservations which are checked in but not yet checked out
  - 1 - Checked In (checked in today reservations)
  - 2 - Checked Out (checked out today reservations)
  - 3 – Canceled
  - 4 - No ShowFor request version 1.1.0 and higher allowed values are:
  - 0 -tentative
  - 1 -definitive
  - 3 -not affecting availability
- k. CrsSystem
- l. CrsResNumber
- m. CompanyID
- n. TravelAgentID
- o. SourceID
- p. BookerID
- q. AccompanyingGuest – Same as ProfileID, but search also performed in the profiles linked to reservation

- r. AccompanyingGuestLastName Last name of the accompanying guest
- s. AccompanyingGuestFirstName First name of the accompanying guest
- t. PartyID – ID of the party the reservation is attached to
- u. Share – ShareID of the shared reservations
- v. LastUpdateTime
- w. CreationTime
- x. Guest3rdPartyID and Guest3rdPartyType – must be used together. Available only with “Third Party Hotel Connector” or “CRS XML One Way” license.

## Profile

- a. ProfileID
- b. GuestFirstname
- c. GuestName
- d. Name2
- e. ProfileType(1 - Company 2 - Individual 3 – Employee)
- f. ProfileCategory  
(1=Company 2=Travel Agent 3=Source 4=Individual 5=Vendors 6=Meeting Planner)
- g. LoginName (Requires Suite8 webConnect or Suite8 Homepage CCM or CRS XML ONE/TWO WAY license)
- h. LoginPassword (Requires Suite8 webConnect or Suite8 Homepage CCM or CRS XML ONE/TWO WAY license). This is a Base64 encoded SHA1 hash of the real password
- i. ProfileGlobalID
- j. CentralSyncTime
- k. City
- l. ZIP
- m. CorporateID
- n. EMail
- o. ProfileExternalID
- p. Membership – membership number
- q. MembershipType
- r. Guest3rdPartyID and Guest3rdPartyType – must be used together. Available only with “Third Party Suite8 Central” or “CRS XML One Way” license.
- s. LastUpdateTime
- t. CreationTime

## SystemDate

All conditions and fields list are ignored

## SystemInfo

All conditions and fields list are ignored

## AvailabilityForWeb

- a. GuestArrival (Arrival date)
- b. GuestDeparture(Departure date)
- c. Child – Number of children by category. Has to be used with conjunction with attribute "attr" which contains the required child type.  
For Example <field name="Child" attr="JUN">2</field> means 2 juniors.
- d. NoOfAdults
- e. NoOfRooms



- 
- f. RateCode
  - g. RoomType
  - h. WebOnly if the value is 1, only rates opened for web will be returned. Default value is 1
  - i. MultipleRates – Needed only in conjunction with WebOnly = 1. If MultipleRates = 0, only the lowest rate will be returned. Default value is 0.
  - j. GetRateWebInfo (if specified, RateWebInfo will be returned instead of RateInfo)
  - k. ProfileID – Limits rates to profile linked rates. In case linked rates exist, WebOnly flag will be ignored.
  - l. AvailabilityLimit – validates rates against defined availability limit. If AvailabilityLimit field contains value MFHP, the availability limit defined for Suite 8 webConnect is taken, otherwise the limit with the selected name will be used
  - m. PricePerPerson – if specified, price break down per adult / child age category will be returned. Please note that package price will be returned without these details
  - n. HotelSegmentCode - if specified, only room types which belong to the defined Segment Code, will be returned
  - o. WebSalesCategory - if specified Web Sales category info for every rate will be returned

## Posting

- a. PostingID
- b. ProfileID
- c. GuestNum
- d. DepartmentCode
- e. PostingDescription
- f. PostingDate
- g. PostingProfileOrigID
- h. PostingGuestNumOrig
- i. PostingComment
- j. PostingCheckNr

## ChildrenCategories

- a. ChildrenCategoriesCode

## CreditCardTypes

- a. CreditCardCode

## HtmlInfo

- a. HtmlName
- b. HtmlParameter

All parameters are mandatory

## SeatingArrangments

- a. SeatingID
- b. SeatingShortDesc

## Event Info Board

- a. Eventid
- b. EventStartDate
- c. EventEndDate

- 
- d. EventStatusIsDEF
  - e. EventStatusIsDEFForTEN
  - f. EventRoomID

## Activity

- a. ActivityStartDate
- b. ActivityEndDate
- c. ActivityTypeID
- d. ActivityManagerID
- e. MaintenanceManagerID
- f. MaintenanceEstimatedDurationMin

## Task

- a. TaskStartDate
- b. TaskEndDate
- c. TaskTypeID
- d. TaskManagerID

## Group Template

- a. GroupTemplateID
- b. GroupTemplateName
- c. GroupTemplateContactID
- d. GroupTemplateInactive

## Group

- a. GroupID
- b. GroupName
- c. GroupContactID
- d. GroupFrom
- e. GroupTo
- f. GroupTemplateID
- g. GroupInactive

## BlockHeader

- a. BlockHeaderID
- b. BlockHeaderCode
- c. BlockHeaderName
- d. BlockHeaderFrom
- e. BlockHeaderTo
- f. BlockHeaderStatus
- g. BlockHeaderOpenForPickup

## BlockDetail

- a. BlockHeaderID
- b. BlockDetailStartDate
- c. BlockDetailEndDate

---

## CurrencyList

- a. CurrencyShortdesc
- b. CurrencyISO3
- c. CurrencyDepartment

## HotelAvailability

The request should be performed using HotelAvailability type (see example)

- a. StartDate
- b. EndDate
- c. CalculateFO
- d. Overbooking
- e. CalculateDefinitive
- f. CalculateTentative
- g. CalculateOOO – Result will include Out of Orders and Out of service rooms figures
- h. BookingPosition – Result will include additional fields for Availability and Occupancy inclusive booking positions.

## Overbooking

- a. OverbookingDate
- b. OverbookingType - 1 = House Overbooking      2 = Room Type Overbooking

## Market

- a. MarketID
- b. MarketShortdesc
- c. MarketMainMarket – Short Description

## MainMarket

- a. MainMarketID
- b. MainMarketShortdesc

## RateList

- a. RateCode
- b. RateStartSell
- c. RateEndSell
- d. RatePublic
- e. RateWebEnabled

## ReportList

- a. ReportID
- b. ReportSectionID
- c. ReportName
- d. ReportExternalID

## ReportParameterList

- a. ReportID
- b. ReportSectionID
- c. ReportParameterName

---

## ReportSectionList

- a. ReportSectionID
- b. ReportSectionParentID
- c. ReportSectionReportType

## StatisticAndFuture (planned functionality only)

- a. Date
- b. GuestNum – reservation ID
- c. ExpArrivalTime
- d. ExpDepartureTime
- e. ResStatus - 0=tentative 1=definitive 3=not effecting availability
- f. NoAvailReason - 1 = Waitlist 2 = Offer 3 = Canceled 4 = No Show
- g. ProfileID
- h. CompanyID
- i. TravelAgentID
- j. SourceID
- k. IndividualID
- l. VendorID
- m. MeetingPlannerID
- n. GroupID
- o. BlockHeaderID
- p. RoomType

## CalendarEvent

- a. CalendarEventFromDate
- b. CalendarEventToDate
- c. CalendarEventType - 0 = miscellaneous event 1 = availability event 2 = public holiday event 3= conference booking
- d. CalendarEventTypeShow - 0-show all;1-show diary; 2-show F4; 3-show CTRL-D; 4-show diary+F4; 5-show diary+CTRL-D; 6-show F4+CTRL-D
- e. CalendarEventTypeWebEnabled
- f. CalendarEventWebEnabled

## VoucherArticles

- a. VoucherArticleID
- b. VoucherArticleName
- c. VoucherArticleWebEnabled

## Voucher

- a. VoucherID
- b. VoucherArticleID
- c. VoucherProfileID
- d. VoucherNumber
- e. VoucherType
- f. VoucherPickupReservationID
- g. VoucherFinalized

---

## BillingInstruction

- a. BillingInstructionID
- b. BillingInstructionReservationID – The BI item belongs to this Reservation
- c. BillingInstructionToReservationID - The BI routes to this reservation
- d. BillingInstructionFAID - The BI item belongs to this Financial Account
- e. BillingInstructionToFAID - The BI routes to this Financial Account
- f. BillingInstructionProfileID
- g. BillingInstructionBookingID - The BI item belongs to this Booking
- h. BillingInstructionBlockID - The BI item belongs to this Block

## Package

- a. StartDate
- b. EndDate
- c. PackageCode
- d. PackageIncluded
- e. PackageShowInRes
- f. ExcludeRateCode – all packages attached to the specified rate will be excluded
- g. ExcludeBlockCode – all packages attached to the specified block will be excluded
- h. WebOnly

## RoomType

- a. RoomTypeShortDesc

## TableReservation

- a. TableReservationID
- b. TableReservationDate
- c. TableReservationStartTime – Data format HH:MM:SS
- d. TableReservationEndTime – Data format HH:MM:SS
- e. TableReservationRestaurantID
- f. ProfileID
- g. TableReservationGuestLastName
- h. TableReservationGuestFirstName
- i. Communication – any email / phone etc. Always searches for the substring. Not case sensitive
- j. RoomNum – Room number of the attached PMS reservation
- k. SeatingAt – Data format HH:MM:SS. To find out which reservation is actual on this time. Should work together with condition

## TableAvailability

- a. StartDate - mandatory
- b. StartTime - mandatory
- c. EndTime
- d. Restaurant – Reatuarant ID
- e. Persons
- f. Child - Number of children by category. Has to be used with conjunction with mandatory attribute "attr" which contains the required child type  
For Example <field name="Child" attr="JUN">2</field> means 2 juniors.
- g. WebOnly – By default only Web Enabled restaurants are returned. If WebEnabled is 0, all restaurants will be returned

- h. AlternativeSuggestions – if multiple suggestions are required, this parameter can be set. So requester doesn't send a request for 5 time frames, but instead asking the interface to return suggestions. The Interface looks to the restaurant config and returns either intervals with 30 min difference or web enabled intervals

## NoteCategory

All fields as described in **Fields available for selection by request type**, in the section NoteCategory can be used as optional conditions.

## Note

All fields as described in **Fields available for selection by request type**, in the section Note can be used as optional conditions.

## Conference and Catering: General available search criteria.

The fields mentioned under the chapter: **Fields available for selection by request type under Conference and Catering** can be used as optional conditions

All optional and mandatory parameters have to be defined as conditions in the conditions node.

For Example:

```
<fidelio Version="1.0.0">
  <request ID="25" UserName="XMLInterface" Password="REMOVED" PasswordEncrypted="true">
    <query ReqType="FunctionSpaceAvailability" TranslationTable="XML">
      <conditions>
        <condition name="FromTime" operation="eq">16.02.2015 12:27:00</condition>
        <link operator="AND">
          <condition>
            <condition name="ToTime" operation="eq">16.02.2015 12:27:09</condition>
            <link operator="AND">
              <condition>
                <condition name="Capacity" operation="eq">10</condition>
                <link operator="AND">
                  <condition>
                    <condition name="WebOnly" operation="eq">0</condition>
                    <link operator="AND">
                      <condition>
                        <condition name="EventTypesCodes" operation="eq">MET,CBA</condition>
                        <link operator="AND">
                          <condition>
                            <condition name="PackagesCodes" operation="eq">CP3,CP2</condition>
                            <link operator="AND">
                              <condition>
                                <condition name="HotelSegmentCodes" operation="eq">CC1</condition>
                                <link operator="AND">
                                  <condition>
                                    <condition name="FunctionSpaceShareable"
operation="eq">1</condition>
                                  </condition>
                                </link>
                              </condition>
                            </link>
                          </condition>
                        </link>
                      </condition>
                    </link>
                  </condition>
                </link>
              </condition>
            </link>
          </condition>
        </link>
      </conditions>
      <fields>
        <field name="FunctionSpaceID"/>
        <field name="FunctionSpaceShortDescription"/>
        <field name="FunctionSpaceLongDescription"/>
        <field name="FunctionSpaceWebEnabled"/>
        <field name="FunctionSpaceShortWebInfo"/>
        <field name="FunctionSpaceLongWebInfo"/>
      </fields>
    </query>
  </request>
</fidelio>
```

```
<field name="FunctionSpaceTypeShortDescription" />
<field name="FunctionSpaceTypeLongDescription" />
<field name="FunctionSpaceShareable" />
<field name="FunctionSpaceCombined" />
<field name="FunctionSpaceMinCapacity" />
<field name="FunctionSpaceMaxCapacity" />
<field name="FunctionSpaceCostPrice" />
<field name="FunctionSpaceRate" />
<field name="FunctionSpaceSeating" />
</fields>
</query>
</request>
</fidelio>
```

## FunctionType

The optional parameter condition **WebOnly** can be used.

The default value is 1, if this is used, then function types marked as open for web are returned, if 0 is used all function types are returned

## ConfiguredStatusList

The optional parameter condition **WebOnly** can be used.

The default value for this condition is 1, if 1 is used, the status lists marked as open for web are returned, if 0 is used all status lists are returned

## EventType

The optional parameter condition **WebOnly** can be used. The default value for this condition is 1 and returns event types marked as open for web, 0 returns all

## EventStatus

The optional parameter condition **WebOnly** can be used. The default value for this condition is 1 and returns event statuses marked as open for web, 0 returns all

## FunctionSpace

The optional parameter condition **WebOnly** can be used. The default value for this condition is 1 and returns Functionspaces marked as open for web, 0 returns all

## FunctionSpaceSeating

The optional parameter condition **WebOnly** can be used. The default value for this condition is 1 and returns all function space seatings marked as open for web, 0 returns all

## FunctionSpaceRate

The optional parameter condition **WebOnly** can be used. The default value for this condition is 1 and returns function space rates marked as open for web, if the value is set to 0, all function space rates are returned

## FunctionSpaceAvailability

The following parameter conditions are mandatory

- a. FromTime
- b. ToTime
- c. Capacity

The following parameter conditions are optional:

- 
- a. **WebOnly**  
the default value for this condition is 1, then all capacities for function spaces, marked as open for web are returned, if the condition is set to 0, all capacity are returned. The Capacity is taking effect on the FunctionSpaceRate and the FunctionSpaceSeating.1 returns only rates and seatings open for the web, 0 returns all
  - b. **EventTypesCodes**. Comma separated list of even types
  - c. **SeatingsCodes**. Comma separated list of seatings
  - d. **PackagesCodes**. Comma separated list of seating.
  - e. **HotelSegmentCodes**. Comma separated list of hotel segments

## ConferencePackagesList

The following parameter conditions are optional:

- a. **WebOnly**  
1 is the default and returns conference packages marked as open for web. 0 returns all all conference packages. ConferencePackageEvents are affected as well with this parameter, if the condition is 1 then all events linked to the package and with an event type marked as open for web, are returned. If the condition is 0 all events linked to the package are returned

## CateringItem

The optional paramter condition **WebOnly** can be used.

The default value for the condition is 1 and returns catering items marked as open for web, if it is set to 0 all catering items are returned. If the condition is used for fCateringItem, it affects the field CateringItemRate as well.

## CateringItemRate

The optional parameter condition **WebOnly** can be used.

The default value for the condition is 1 and returns catering item rates marked as open for web, 0 returns all catering item rates

## StandardMenu

The optional condition **WebOnly** can be used.

The default value for the condition is 1 and returns standard menus marked as open for web, 0 returns all standard menus

## MenuItem

The optional parameter condition **WebOnly** can be used.

The default value for the condition is 1 and returns all menu items marked as open for web, 0 returns all menu items

## ConferenceBooking

The fields described in the chapter **Fields available for selection by request type** under **Conference and Catering** can be used as optional conditions.

## BookingPackage

The fields described in the chapter **Fields available for selection by request type**, in the section **Conference and Catering** can be used additionally as optional conditions.



---

## **BookingEvent**

The fields described in the chapter **Fields available for selection by request type**, in the section Conference and Catering can be used additionally as optional conditions.

## **EventFunctionSpace**

The fields described in the chapter **Fields available for selection by request type**, in the section Conference and Catering can be used additionally as optional conditions.

## **EventCateringItem**

The fields described in the chapter **Fields available for selection by request type**, in the section Conference and Catering can be used additionally as optional conditions.

## **EventStandardMenu**

The fields described in the chapter **Fields available for selection by request type**, in the section Conference and Catering can be used additionally as optional conditions.

## **StandardMenuMenuItem**

The fields described in the chapter **Fields available for selection by request type**, in the section Conference and Catering can be used additionally as optional conditions.

---

---

## 5 Fields available to update / insert

The search conditions can include the same fields as for the regular search type. The following is the list of fields allowed to be changed:

### Reservation

The Request requires at least one of the following licenses:

- XML Interface Standard Edition
- Suite8 webConnect
- XML Interface Enhanced Edition
- Suite8 Central
- XML Interface for Varibon/VariTable/JetWeb
- CRS XML ONE/TWO WAY

On reservations, some data, such as the rate, the room type the market code can change. To submit this type of data, each reservation field element can contain optional attributes, for example "startDate" and "endDate" indicating a date range for when the submitted data is valid. If the 'startDate' is omitted, the arrival date is taken. If the 'endDate' is omitted, the departure date is taken:

- a. RoomType (max 5 char)
- b. RoomNum (max 10 char)
- c. GuestArrival (Arrival date)
- d. GuestDeparture (Departure date)
- e. ReservationComment1 (max 100 char)
- f. ReservationComment2 (max 100 char)
- g. ProfileID
- h. PaymentMethod – the department code which is associated with the payment
- i. CrsSystem - Only for Insert- it is the ID of the CRS system where the reservation originates from (max 30 char)
- j. CrsResNumber - Only for Insert – This is the unique ID coming from the CRS System it is related to the field CrsSystem, to support multiple CRS systems. (max 30 char)
- k. NoOfRooms
- l. NoOfAdults
- m. RateCode
- n. RateValue
- o. ForeignRateValue
- p. CurrencyCode (This field can be different from the ISO Standard. Please refer to the Suite8 Configuration of the property)
- q. MarketCode
- r. CompanyID
- s. TravelAgentID
- t. SourceID
- u. BookerID
- v. ReservationAttribute – this must be used in conjunction with a mandatory attribute "attr" containing the required attribute type.  
For Example `<field name="ReservationAttribute" attr="ABC"/>`
- w. Child – this must be used in conjunction with a mandatory attribute "attr" containing the required child type.  
For Example `<field name="Child" attr="JUN"/>` for junior

- 
- x. SourceCode
  - y. ChannelCode
  - z. Guarantee
  - aa. GroupID
  - bb. CreditCardNo – this is mandatory if there is any other credit card detail sent, such as number, CreditCardCode
  - cc. ExpiryMonth – this is mandatory if there is any other credit card detail sent
  - dd. ExpiryYear - this is mandatory if there is any other credit card detail sent
  - ee. CardholderName (max 30 char) – this is mandatory if there is any other credit card detail sent
  - ff. CreditCardCode – this is mandatory if there is any other credit card detail sent, such as credit card number, expiration date
  - gg. ApprovalCode
  - hh. ApprovalAmount
  - ii. BlockCode. When attaching a block, the interface can be instructed to use the Market, Source and Channel code defined on the block. If this functionality is used, the element contains the attributes “UseMarket”, “UseSource” and “UseChannel” with the value “1”
  - jj. Notes - the Attribute “attr” must contain the note category (length is not limited)
  - kk. OptionDate
  - ll. PackageCode.  
Following attributes are supported:
    - Included (allowed values are 0 and 1)
    - Price (if manual price is required)
 When sending an update message and the attribute: ‘addData’ contains the string “DELETE”, the package is detached. For Example:  

```
<field name="PackageCode" Price="9.99" Included="1">ZB</field>
```
  - mm. CustomField – the details of custom fields are not supported currently. Each field must contain an attribute “attr” with a field name. The data format depends on the defined field format. All date fields must use the format: ‘dateTime’  
For example  

```
<field name="CustomField" attr="textfield">custom text</field>
<field name="CustomField" attr="doublefield">568.88</field>
<field name="CustomField" attr="integerfield">10</field>
<field name="CustomField" attr="datefield">25.05.2008 11:25:00.000
UTC-60</field>
```
  - nn. Waitlist. If the value 1 is passed, for example: 

```
<field name="Waitlist">1</field>
```

 – the reservation status is set to ‘WaitList’. Only 1 is allowed. It is not possible to move a reservation to affective availability.
  - oo. RateRoomType – room type used by rate calculation. I can also be specified per date
  - pp. AllowHouseOverbooking - Only for Insert. This forces a reservation creation in the event that the House is overbooked. The User right under Reservation for ‘House Overbooking’ is then required for the XML Interface user.
  - qq. AllowRoomOverbooking - Only for Insert. This forces a reservation creation even if the room type is overbooked. The User right under Reservation for ‘Overbooking’ is then required for the XML Interface user.
  - rr. SendConfirmation -  
Only for Insert. If this is present in the message, a confirmation letter will be sent. The Confirmation report is taken form the rate configuration on the XML Interface. Optional attributes of the message are:  
TO – Recipient address. If not present, the email of the main reservation profile will be taken  
CC – Carbon copy address

---

BCC – Blind carbon copy address

For example:

```
<field name="SendConfirmation" TO="name@example.com" CC=" "
BCC=" ">1</field>
```

- ss. AccompanyingGuest – allows adding or removing the accompanying guest to or from the reservation. The profile ID should be send as a node value. If required, the age of the guest can be sent as an attribute "attr".

```
<field name="AccompanyingGuest" attr="6">1556</field>
```

in case of update, if Attribute addData contains a string "DELETE", the profile will be unlinked:

```
<field name="AccompanyingGuest" addData="DELETE">1556</field>
```

- tt. ReservationPreference – The preference is specified as a mandatory attribute "attr". For example:

```
<field name="ReservationPreference" attr="NSM"></field>
```

In order to unlink the preference, the addtribute "addData" has to be set to "DELETE"

```
<field name="ProfilePreference" attr="NOS" addData="DELETE" />
```

- uu. Policy – The short description of the Policy. If this is empty, the Policy will be removed from the reservation

## Profile

The Request requires at least one of the following licenses:

- XML Interface Standard Edition
- Suite8 webConnect
- XML Interface Enhanced Edition
- Suite8 Central
- XML Interface for Varibon/VariTable/JetWeb
- CRS XML ONE/TWO WAY
  - a. GuestFirstname (max 35 char)
  - b. GuestName (max 35 char)
  - c. Name2 (max 35 char)
  - d. ProfileType Only for Insert (1 - Company 2 - Individual 3 – Employee)
  - e. ProfileCategory  
Only for Insert (1=Company,2=Travel Agent,3=Source,4=Individua,5=Vendors, 6=Meeting Planner)
  - f. Title
  - g. AddressGreeting (max 30 char)
  - h. LetterGreeting (max 30 char)
  - i. Birthday

The following fields for address updates insert or deletions are supported.

- j. City (max 30 char)
- k. Street1 (max 60 char)
- l. Street2 (max 60 char)
- m. Street3 (max 60 char)
- n. ZIP (max 15 char)
- o. State – In case of profile insert or if updating both: state and country, this field must be sent after Country filed in the XML element sequence.
- p. CountryISO2
- q. CountryISO3
- r. Addresses
- s. PrimaryAddr: is address primary

---

To delete a complete address use the field name Address.  
The attribute attr must contain "DELETE" and addDate the id of address.  
For Example

```
<field name="Address" attr="DELETE" addData="3434"></field>
```

To add an address, the field name PrimaryAddr must be used in combination with the name fields for Street1, City etc.. Use the attribute addDate to group the address together. -1 for first address to add, -2 for second address to add etc..  
Use the attribute attr for the address type.

For Example: adding 2 addresses to a profile.

```
<fields>
  <field name="PrimaryAddr" attr="ADR" addData="-1">111</field>
  <field name="Street1" attr="ADR" addData="-1">MainStreet</field>
  <field name="City" attr="ADR" addData="-1">Anytown</field>
  <field name="PrimaryAddr" attr="BIL" addData="-2">111</field>
  <field name="Street1" addData="-2">MainStreet</field>
  <field name="City" addData="-2">Anytown</field>
</fields>
```

Response

```
<response ID="15" Status="OK">
<updateResponse>
  <rows>
    <row>
      <fields>
        <field name="ProfileID">30621</field>
        <field name="Address" initID="-1"
newID="3456"/>
        <field name="Address" initID="-2"
newID="3457"/>
      </fields>
    </row>
  </rows>
</updateResponse>
</response>
```

To update an address use the attribute addDate to group by address id.  
For Example:

```
<fields>
  <field name="PrimaryAddr" attr="ADR"
addData="3456">0</field>
  <field name="Street1" attr="ADR" addData="3456">Update the
street2.</field>
  <field name="City" attr="ADR"
addData="3456">UpdateCity.</field>
</fields>
```

The following fields for communication are supported:

- t. Telephone (max 60 char)
- u. Email (max 60 char)
- v. Fax (max 60 char)

w. Communication (max 60 char)

All operations for updating or deleting communications require the communication ID. It can be retrieved using the query message. The ID and the ID of the new created communication are also returned within an insert / update message response.

For Example

```
<?xml version="1.0" encoding="utf-8" ?><fidelio Version="1.1.0">
<response ID="{0D30FACF-6454-462C-93CE-8E69C4BD0712}" status="OK">
<updateResponse>
<rows>
<row>
<fields><field name="ProfileID">1526</field><field
name="Communication" newID="1287"/></fields>
</row>
</rows>
</updateResponse>
</response>
</fidelio>
```

Communication fields support the following:

In order to delete the Communication, the attribute "attr" should contain the string "DELETE", the attribute "addData" must contain a communication ID

In order to update the Communication, the attribute "attr" should contain a communication type, the attribute "addData" must contain a communication ID, the attribute Primary may contain a Primary communication indicator.

For Example: Request to add, update and delete a communication.

```
<update ReqType="Profile" TranslationTable="XML">
  <conditions>
    <condition name="ProfileID"
operation="like">30621</condition>
  </conditions>
  <fields>
    <field name="Communication" attr="WEB"
Primary="1">www.example.com</field>  ADD
    <field name="Email"
addData="2283">name@example.com</field>  Update
    <field name="Telephone" attr="DELETE" addData="2290"/>
Delete
  </fields>
</update>
```

Response

```
<response ID="15" Status="OK">
  <updateResponse>
    <rows>
      <row>
        <fields>
          <field name="ProfileID">30621</field>
          <field name="Communication"
newID="2291"/>
        </fields>
      </row>
    </rows>
```

```
</updateResponse>
</response>
```

- x. Gender (M – male, F – Female)
- y. DeleteCode (0 –profile is active 1 – profile is not active)
- z. NationalityISO2
- aa. NationalityISO3
- bb. Language
- cc. LoginName - only for Insert (max 20 char) (Requires Suite8 webConnect or Suite8 CCM Homepage or CRS XML ONE/TWO WAY license)
- dd. LoginPassword (the length is not limited because the system calculates and stores only hash code, so the stored length is always the same) (Requires the Suite8 webConnect or CCM Homepage or CRS XML ONE/TWO WAY license)
- ee. CreditCardNo
- ff. ExpiryMonth
- gg. ExpiryYear
- hh. CardholderName (max 30 char)
- ii. CreditCardCode
- jj. TryToMerge Only for Insert  
Values 1 - true, 0 – false  
If the value is 1, the XML interface searches for existing profiles with similar data (according to the profile merge setup in Suite8)
- kk. TryToGlobalize Only for Insert  
(Requires Third Party Suite8 Central license)  
Values 1 - true, 0 – false  
If the value is 1, the XML interface searches existing profiles with similar data (according to the profile merge setup in Suite8). If a profile is found, then it is merged if not a new record is created. A GlobalID is assigned to the profile and returned in the response message
- ll. CreditStatus - empty=normal 'C'=cash only 'B'=blacklist 'A'=City ledger authorization
- mm. CreditReason (max 60 char)
- nn. CreditStatusChangeTime
- oo. Notes (length is not limited)  
Use the attribute noteID for the id of the note.The attribute “addData” is used to determine if the note is guest viewable or not. Use attribute Delete=“1” to delete a note record.

For Example: Request to add, update and delete notes.

```
<update ReqType="Profile" TranslationTable="XML">
<conditions>
  <condition name="ProfileID"
operation="like">30495</condition>
</conditions>
  <fields>
    <field name="Notes" attr="PRF">New note</field>
    <field name="Notes" attr="PRF" addData="0"
noteID="1508">Update Me</field>
    <field name="Notes" noteID="1500" Delete="1"/>
  </fields>
</update>
```

```
Response
<response ID="15" Status="OK">
<updateResponse>
```

```

        <rows>
          <row>
            <fields>
              <field name="ProfileID">30495</field>
              <field name="Note" newID="1513" />
            </fields>
          </row>
        </rows>
      </updateResponse>
    </response>

```

pp. BirthPlace (max 30 char)

qq. Department (max 60 char)

rr. Position (max 60 char)

ss. Membership.

The attribute "attr" always contains the card ID. If this is submitted, the card is added to the profile.

In order to delete the card attribute "addData" should contain the value DELETE.

Additional attributes which could be used dependant on the card type are:

Level, Type, StartDate, ExpiryDate, SignupDate, TaxOffice

For Example:

```

<field name="Membership" Type="CON" StartDate="01.01.2007"
TaxOffice="Anytown">123456</field>

```

tt. ProfileExternalID – If the field is specified in the insert message, and the same ProfileExternalID already exists in the DB and the option 'TryToMerge' field is used, the interface will merge the profiles. Otherwise the request will be ignored and the ID of the existing profile will be returned.

uu. ProfileAttribute – this must be used in conjunction with a mandatory attribute "attr" containing the required attribute type.

In order to delete the attribute, the element "addData" should contain the string "DELETE".

vv. ProfilePreference – The preference is specified as a mandatory attribute "attr". For example:

```

<field name=" ProfilePreference" attr="NOS" />

```

In order to unlink the preference, the attribute "addData" has to be set to "DELETE"

```

<field name=" ProfilePreference" attr="NOS" addData="DELETE" />

```

ww. Guest3rdPartyID – ID of the guest in the 3rd party system. This must be used in conjunction with the attribute "attr" containing the code of the CRS system. Available only with the licenses: "Third Party Suite8 Central" or "CRS XML One Way".

xx. PersonalDocument

The attribute "attr" always contains the document ID. If this is submitted, the document will be added to the profile.

The node value of the element must contain the document number with a maximum length of 30 characters

In order to delete the document the attribute "addData" should contain the value DELETE.

Additional supported attributes are:

- a. Type – Document type
- b. Name (maximum 30 characters)
- c. CountryISO2
- d. CountryISO3
- e. NationalityISO2
- f. NationalityISO3
- g. ValidFrom



- h. ValidTo
  - i. IssuedBy (maximum 30 characters)
  - j. PlaceOfIssue (maximum 30 characters)
  - k. IssuerCountryISO2
  - l. IssuerCountryISO3
  - m. IssuerState
  - n. FatherName (maximum 30 characters)
  - o. MotherName (maximum 30 characters)
  - p. RegCity (maximum 30 characters)
  - q. RegMunicipality (maximum 30 characters)
  - r. RegStreet (maximum 30 characters)
  - s. BornInISO2
  - t. BornInISO3
  - u. BornState
  - v. BirthPlace (maximum 30 characters)
- For Example : `<field name="PersonalDocument" attr="REI" Name="Smith" CountryISO2="DE" NationalityISO2="DE" ValidFrom="01.01.2004" ValidTo="01.01.2014" IssuedBy="Issuer">123456</field>`

ww. AddProfileLink.

The following attributes are required:

- a. ProfileID or ProfileGlobalID
- b. Attr – role
- c. addData – defines if this link is primary. Allowed values 1(true) or 0(false)

xx. DeleteProfileLink

The following attributes are required:

- a. ProfileID or ProfileGlobalID

yy. EditProfileLink

The following attributes are required:

- a. ProfileID or ProfileGlobalID
- b. Attr – role
- c. addData – defines if this link is primary. Allowed values are 1=true or 0=false

zz. CustomField –Each field must contain the attribute “attr” with the field name. The data format depends on the defined field format. All Date fields must use DateTime format

For Example

```
<field name="CustomField" attr="textfield">custom text</field>
<field name="CustomField" attr="doublefield">568.88</field>
<field name="CustomField" attr="integerfield">10</field>
<field name="CustomField" attr="datefield">25.05.2008
11:25:00.000 UTC-60</field>
```

aaa.NoMailing - acceptable values are 1= true/0=false

bbb. NoEMailing - acceptable values are 1= true/0=false

ccc. GuestComment – Updating / Inserting Guest Comments

If the attribute “addData” contains the comment ID, the comment is updated, if not a new comment is inserted. The ID of the inserted comment is returned in the response message. The attribute “attr” should contain the comment type, the attribute Date – the comment date.

For Example

```
<field name="GuestComment" addData="1216" attr="COM"
Date="20.11.2014">This is the comment</field>
```

In order to delete a comment, the attribute “addData” should contain the value DELETE and the “comment ID” in the “attr”

---

For Example

```
<field name="GuestComment" addData="DELETE"
attr="1216"></field>.
```

ddd. StorePrivateData – Updating / Inserting Anonymization Status  
acceptable values are 0 = Normal, 1 = Store private Data

## GuestAnonymization

- ProfileID – Updating, to be used to anonymize guest profiles

Fields in response are:

-AnonymizationStatus

0 = already anonymized

1 = flagged as permanent

2 = has Activity

3 = success

4 = other failure

5 = not an individual profile

- AnonymizationMessage

## Voucher

The Request requires at least one of the following licenses:

- XML Interface Standard Edition
- XML Interface Enhanced Edition
- Suite8 Central
- Voucher Management
  - a. VoucherArticleID – mandatory, for Insert only
  - b. VoucherProfileID - mandatory
  - c. VoucherDescription
  - d. VoucherText
  - e. VoucherValidFrom
  - f. VoucherValidUntil
  - g. VoucherDedication
  - h. VoucherOnHoldReason

Note: The voucher amount is taken based on the voucher type setup and cannot be changed.

## Profile Merge

Since Suite8 Version 8.8.3.2 Profile Merge is available for Suite8 XML Interface. The merge is based on the defined local merge rule in the Suite8 Configuration. The following is an example of a profile update message:

```
<?xml version="1.0" encoding="UTF-8"?>
<fidelio xmlns="V8XMLInterface"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="V8XMLInterface"
<request ID="55555" UserName="XMLInterface" Password="xxx">
<update ReqType="Profile">
<fields>
<field name="MergeTo">921056</field>
</fields>
<conditions>
<condition name="ProfileID" operation="eq"
xsi:type="ProfileConditionType">921051</condition>
</conditions>
```

---

```
</update>  
</request>  
</fidelio>
```

All profiles returned by these conditions (might be multiple) will be merged to the one specified as 'MergeTo'. If the field list contains the field 'MergeTo', all other fields are ignored.

## ProfileAndReservation

The Request requires at least one of the following licenses:

- XML Interface Standard Edition
- Suite8 webConnect
- XML Interface Enhanced Edition
- Suite8 Central
- XML Interface for Varibon/VariTable/JetWeb
- CRS XML ONE/TWO WAY

This request accepts all fields from the Reservation and the Profile insert messages. It creates both profile and reservation referencing to this profile. The Response message contains the ID of the created reservation as well as the profile ID. For more details please refer to the example of the insert Reservation and insert Profile messages available under XML Interface Specifications.

## FinancialAccount

The Request requires at least one of the following licenses:

- XML Interface Standard Edition
- Suite8 webConnect
- XML Interface Enhanced Edition
  - a. ProfileID
  - b. MarketCode
  - c. FANumber
  - d. FAGroup – optional parameter. It is ignored if FANumber is specified. If group configuration has AUTONUMBER flag, than new financial account with auto number will be created.
  - e. FADescription (maximum 30 characters)
  - f. ExpArrival
  - g. ExpDeparture
  - h. ActArrival
  - i. ActDeparture
  - j. CreditLimitType
  - k. CreditLimitAmount

## Activity

The Request requires at least one of the following licenses:

- XML Interface Standard Edition
- MyCCM Homepage
- Suite8 webConnect
- XML Interface Enhanced Edition
- Suite8 Central
  - a. ActivityCompanyID
  - b. ActivityCompanyGlobalID

- 
- c. ActivityContactID
  - d. ActivityContactGlobalID
  - e. ActivityManagerID
  - f. ActivityManagerGlobalID
  - g. ActivityTypeID
  - h. ActivityAllDayID
  - i. ActivityShowasID
  - j. ActivityReminderID
  - k. ActivityNeedReminder
  - l. ActivityPrivate
  - m. ActivityDescription (max 60 char)
  - n. ActivityStartDateTime
  - o. ActivityEndDateTime
  - p. ActivityCompletedDate
  - q. ActivityLinkedBookingId
  - r. ActivityLinkedCommunicationID
  - s. ActivityCheckAvailability
  - t. ActivityNotes (length is not limited)

## Task

The Request requires at least one of the following licenses:

- XML Interface Standard Edition
- MyCCM Homepage
- Suite8 webConnect
- XML Interface Enhanced Edition
- Suite8 Central
  - a. TaskCompanyID
  - b. TaskCompanyGlobalID
  - c. TaskContactID
  - d. TaskContactGlobalID
  - e. TaskManagerID
  - f. TaskManagerGlobalID
  - g. TaskTypeID
  - h. TaskDescription (maximum 60 characters)
  - i. TaskStartDate
  - j. TaskEndDate
  - k. TaskCompletedDate
  - l. TaskLinkedBookingID
  - m. TaskLinkedCommunicationID
  - n. TaskStatusID
  - o. TaskPriorityID
  - p. TaskCompletedPercentageID
  - q. TaskNotes (length is not limited)

## Group Template

The Request requires at least one of the following licenses:

- XML Interface Standard Edition
- XML Interface Enhanced Edition

- 
- Suite8 Central
    - a. GroupTemplateName (maximum 60 characters)
    - b. GroupTemplateContactID
    - c. GroupTemplateComment (not limited)
    - d. GroupTemplateInactive
    - e. CountryISO2
    - f. CountryISO3
    - g. Language
    - h. State

## Group

The Request requires at least one of the following licenses:

- XML Interface Standard Edition
- XML Interface Enhanced Edition
- Suite8 Central
  - a. GroupName (maximum 60 characters)
  - b. GroupContactID
  - c. GroupFrom
  - d. GroupTo
  - e. GroupTemplate
  - f. GroupTemplateID
  - g. GroupInactive
  - h. GroupComment (maximum 200 characters)
  - i. CountryISO2
  - j. CountryISO3
  - k. Language
  - l. State

## CreditCardApproval (Insert only)

The Request requires at least one of the following licenses:

- XML Interface Standard Edition
- XML Interface Enhanced Edition
  - a. GuestNum (Reservation number)
  - b. CreditCardNo
  - c. ExpiryMonth
  - d. ExpiryYear
  - e. CardholderName (max 30 char)
  - f. CreditCardCode
  - g. ApprovalCode
  - h. ApprovalAmount

## HurdleRate

The Request requires at least one of the following licenses:

- Hurdle Rates XML Interface

The available method for this request is: update only

The system updates existing records and inserts a new record if there is no record found for the combination of "Day", "Length of Stay" and "Room type":

- a. RoomType

- b. GuestArrival – Arrival Date
- c. HurdleLOS - the Length of Stay. If no explicit LOS is given, the reservation days with LOS = 0 are added up to a hurdle value
- d. HurdleRateAmount – the amount which should be exceeded for the given arrival date and Length of Stay to pass the hurdle
- e. HurdleDelta – the variation of the hurdle rate each time a reservation is made. If the Length of Stay is not “0”, this is not allowed
- f. HurdleClosed – the value 1 defines: Closed for this arrival day and this length of stay
- g. HurdleMaxRateAmount- Maximum amount of the hurdle rate

## Overbooking

The Request requires at least one of the following licenses:

- XML Interface Enhanced Edition
- Hurdle Rates XML Interface

The system will update existing records and insert a new record if no record is found for this combination of Day and Room type

- a. RoomType
- b. OverbookingDate
- c. OverbookingType - 1 = House Overbooking 2 = Room Type Overbooking
- d. OverbookingNoOfRooms - Number of Rooms for this overbooking

## BAR

The Request requires at least one of the following licenses:

- Hurdle Rates XML Interface

The available method for this request is: update only

The system will update existing records and insert a new one if the record for this combination of Day, Length of Stay and BAR code is not found.

- a. BarCode
- b. RoomType
- c. GuestArrival – Arrival Date
- d. BARLOS - Length of Stay for this BAR Code
- e. BarDeleteDay – If included in the request, all BAR records for selected day for all lengths of stay will be deleted

## BillingInstruction

The Request requires at least one of the following licenses:

- XML Interface Standard Edition
- XML Interface Enhanced Edition
- Suite8 Central
- XML Interface for Varibon/VariTable/JetWeb
  - a. BillingInstructionFromDate
  - b. BillingInstructionToDate
  - c. BillingInstructionReservationID – The BI item belongs to this Reservation
  - d. BillingInstructionToReservationID - The BI routes to this reservation
  - e. BillingInstructionFAID - The BI item belongs to this Financial Account
  - f. BillingInstructionToFAID - The BI routes to this Financial Account
  - g. BillingInstructionProfileID
  - h. BillingInstructionBookingID - The BI item belongs to this Booking

- 
- i. BillingInstructionBlockID - The BI item belongs to this Block
  - j. BillingInstructionNote
  - k. BillingInstructionVideoCheckout
  - l. BillingInstructionBillView
  - m. BillingInstructionDelete – if this field is present in the request, the billing instruction will be deleted
  - n. BillingInstructionDepartment – this element is in the request for each included department code and for each element a mandatory attribute “attr” is required with the following values: 1 = include, 2 = exclude, 3 = all postings included (used for Master Share)  
For example:  
`<field name="BillingInstructionDepartment" attr="1">260</field>`
  - o. BillingInstructionGroup – the element is repeated per included group. Each element requires the mandatory attribute “attr” with the following values: 1 = include, 2 = exclude, 3 = all postings included (used for Master Share)  
For example:  
`<field name="BillingInstructionGroup" attr="1">AL</field>`
  - p. BillingInstructionDelete – if this field is in the request, the billing instruction will be deleted. For example:  
`<update ReqType="BillingInstruction"><fields>  
<fieldname="BillingInstructionDelete">1</field></fields><conditions><condition name="BillingInstructionID" operation="eq">4630</condition>  
</conditions>  
</update>`

If at least one element “BillingInstructionDepartment” or “BillingInstructionGroup” exists in the update request, the values in the request overwrite existing lists (the values provided in the request will be deleted)

## Table Reservation

The Request requires at least one of the following licenses:

- Suite8 webConnect
- XML Interface Enhanced Edition
- Table Reservation license
  - a. RestaurantID
  - b. Date
  - c. StartTime
  - d. EndTime
  - e. NumberOfPerson
  - f. Description
  - g. ReservationID – links a new PMS reservation to the table reservation. There can be multiple elements in the same message.
  - h. ProfileID - links a profile to the table reservation. There can be multiple elements in the same message
  - i. TableNumber – adds a table to the table reservation. There can be multiple elements in the same message
  - j. TableList - comma separated list of all tables. If there are tables assigned to the reservation, they will be removed.
  - k. Sharable
  - l. Seated – flags the reservation as seated. The “seated” time is set to the current time.

- m. SeatedTime – flags the reservation as seated. The “seated time” is set as content of the field. Data format HH:MM:SS
- n. Left – flags the reservation as left. The “Left” time is set to the current time
- o. LeftTime – flags the reservation as left. The “Left” time is set as content of the field. Data format HH:MM:SS
- p. CrsSystem – the name of the submitting CRS system
- q. Child – the number of children by category. This has to be used in conjunction with the attribute "attr" containing the required child type
- r. Cancel – if this is submitted in the update request, the table reservation is cancelled
- s. WaitList – if the value is 1, the reservation is set to the status wait list. In the update message, if the value is set to 0, the reservation status is set to “Definite”
- t. WaitListIfFull – if this is submitted and the value is 1, then a definitive reservation is not possible, the reservation is moved to the status wait list. The Response message contains an element “WaitList” indicating the reservation status

## Postings

The Request requires at least one of the following licenses:

- XML Interface Standard Edition
- XML Interface Enhanced Edition
- XML Interface for Varibon/VariTable/JetWeb
- Hurdle Rates XML Interface

There are two ways to submit CRM postings:

1. Using reservation ID (field GuestNum)
2. Using Room number / Financial account number (using field RoomNum)

Please note that the fields GuestNum and RoomNum cannot be used together.

The system works as follows way:

- If the request contains the field GuestNum system submits the posting for the specified reservation
- If the request contains the field GuestNum system tries to find financial account with specified number. If the find is failed, the it is looking for the checked in reservation with the specified room number

The system also accepts only one the two parameters ‘DepartmentCode’ or ‘ArticleCode’. They can not be used together.

If more than one crm\_posting element is detected in the message, all crm\_posting elements are considered as belonging to the same posting (e.g. split details). The posting sum up amount will be equal to the sum of all details.

For the details on the acceptable fields, please refer to the schema (FidelioXML.xsd)

## Cancellation of reservations

The Request requires at least one of the following licenses:

- Suite8 webConnect
- XML Interface Enhanced Edition
- Suite8 Central Cross Reservations and Availability

For cancelling reservations, the command -> cancel is used.

- a. GuestNum– reservation id to cancel
- b. Reason– text description
- c. DeleteTraces– pass 1 for deleting traces attached to the reservation



- 
- d. SendEmail– pass 1 for sending cancelling via Email

## Note

- a. NoteTableLinkedTo\*  
The object type to which the notes are linked, the following values are accepted:
  - a. ConferenceBooking
  - b. BookingEvent
  - c. EventStandardMenu
  - d. EventCateringItem
  - e. EventFunctionSpace
  - f. Reservation
  - g. Profile
  - h. BlockHeader
  - i. TableReservation
- b. NoteTableID\*  
The table ID of the table which the note is linked
- c. NoteDescription\*
- d. NoteCategoryCode\*
- e. NoteCategoryID or NoteCategoryCode has to be passed
- f. NoteGuestViewable\*  
Defines if the note can be viewed by the customer, then the value is 1 or if it is used internally, then the value is 0

## Conference and Catering General

In the Conference and Catering Module of Suite8 the following request types are available for Insert, Update and Delete:

### Insert:

- Main XML request
- Attributes of a main XML request

### Update:

- Main XML request with condition and XML ID as parameter
- As attributes of a main XML using XML ID field

### Delete:

The following delete command can be used for some of the CCM Resources:

```
<fields><field name="Command">DELETE</field> </fields>
```

Delete operations are restricted to the following fields or their attributes used in a main Event of a Conference Booking:

- a. EventCateringItem
- b. EventFunctionSpace
- c. EventStandardMenu
- d. StandardMenuMenuItem

Delete operations are not allowed in CCM for Conference Bookings or Booking Events.

Cancellation of Conference Bookings or Events is driven by changing the status, such as a cancelled or a lost status, depending on the configured status of the property. For further details, please refer to the Conference and Catering Request and Response examples in this document.

The following example is an update of a main request using events attributes:

```
<fidelio Version="1.0.0">  
<request ID="15" UserName="s" Password="s" PasswordEncrypted="false">  
<update ReqType="ConferenceBooking" TranslationTable="XML">  
  <conditions>
```

```

        <condition name="Booking_ID" operation="eq">1428</condition>
    </conditions>
    <fields>
        <field name="Booking_Name"> I updated the booking
name</field>
<field name="BookingEvent" ID="1840" StatusID="4" StatusCode="DEF"
TypeID="117" TypeCode="MET" EventName="Meeting" NumberOfAttendees="6"
StartTime="22.02.2016 14:30:00" EndTime="22.02.2016 17:00:00"
Infoboard="I Updated you"/>
    </fields>
</update>
</request>
</fidelio>

```

## ConferenceBooking

- a. Booking\_ID  
Only necessary for update request in the condition, delete is not available for Conference Booking
- b. Booking\_CompanyID or
- c. Booking\_ContactID \* -either one is mandatory, the ID of the Company or the ID of the Contact, the ID of the Contact is mandatory if the booking is linked to guest rooms
- d. Booking\_ArrivalDate \*
- e. Booking\_DepartureDate \*
- f. Booking\_Name - maximum 60 characters
- g. Booking\_ConferenceStatus or
- h. Booking\_ConferenceStatusID \* -either one is mandatory if booking is linked to a conference
- i. Booking\_RoomStatus or
- j. Booking\_RoomStatusID \* - either one is mandatory room status code or ID if the booking is linked to guest rooms
- k. Booking\_ConferenceOnly - defines if the booking is for conference only, 0 false, 1 true
- l. Booking\_RoomsOnly- defines if the booking is for guest rooms only, 0 false, 1 true
- m. Booking\_ConferenceAndRooms – defines if the booking is for conference and guest rooms, 0 false, 1 true
- n. Booking\_Notes – No longer used, since 8.9.6.2, Notes are supplied as separate element
- o. Booking\_NoOfAttendes \* - if the booking has booking events, all attendees of the booking are updated when an update request is received
- p. Booking\_MarketCode or
- q. Booking\_MarketCodeID\* - either one, the code or ID is mandatory
- r. Booking\_SourceCode or
- s. Booking\_SourceCodeID\*- either one, the code or ID is mandatory
- t. Booking\_NoteCategory
- u. Booking\_ContactOnSite- maximum 60 characters
- v. Booking\_AccountManagerID
- w. Booking\_AccountManagerCode
- x. Booking\_ChannelCode or
- y. Booking\_ChannelCodeID - - either one, the code or ID can be passed
- z. Booking\_ConferenceManagerID
- aa. Booking\_ConferenceManagerCode
- bb. Booking\_DecisionDate
- cc. Booking\_FollowupDate
- dd. Booking\_CutoffDate
- ee. Booking\_Infoboard - maximum 60 characters

- ff. Booking\_FunctionTypeID
- gg. Booking\_FunctionTypeCode
- hh. Booking\_CurrencyID
- ii. Booking\_CurrencyCode
- jj. BookingEvent
  - a. StatusID
  - b. StatusCode
  - c.TypeID
  - d. TypeCode
  - e. EventName
  - f. NumberOfAttendees
  - g. StartTime
  - h. EndTime
  - i. Infoboard
  - j. BookingPackageID

For example:

```
<field name="BookingEvent" StatusID="4" StatusCode="DEF"
TypeID="117" TypeCode="MET" EventName="Meeting"
NumberOfAttendees="50" StartTime="29.02.2016 14:30:00"
EndTime="29.02.2016 17:00:00" Infoboard="Sales Event"
BookingPackageID="0"/><field name="BookingEvent" StatusID="4"
StatusCode="DEF" TypeID="106" TypeCode="REC"
EventName="Reception" NumberOfAttendees="2"
StartTime="29.02.2016 09:00:00" EndTime="01.03.2016 22:00:00"
Infoboard=" " BookingPackageID="0"/><field name="BookingEvent"
StatusID="4" StatusCode="DEF" TypeID="117" TypeCode="MET"
EventName="Meeting" NumberOfAttendees="50"
StartTime="02.03.2016 09:00:00" EndTime="02.03.2016 12:00:00"
Infoboard="Micros Sales Event" BookingPackageID="0"/>
```

- kk. Note

The request may contain multiple elements. The data is indicated in the attributes "attr" element.

- a. ID – only used for the update request
- b. CategoryID
- c. CategoryCode
- d. Description
- e. GuestViewable
- f. TableLinkedTo

Example:

```
<field name="Note" CategoryID="5" CategoryCode="BOO" Description="This is
a booking note" GuestViewable="1" TableLinkedTo="ConferenceBooking"/>
```

## BookingPackage

When inserting a package all configured events linked to the package are inserted automatically.

Please refer to ConferencePackageList -> ConferencePackageEvents.

- a. BookingPackageID- needs only to be passed in the condition for the update request
- b. BookingPackageBookingID\*
- c. BookingPackageName- only required if different than the name defined on the package configuration The same rule applies to the ConferencePackage Request
- d. BookingPackageConfigID
- e. BookingPackageConfigID or
- f. BookingPackageConfigCode\*- either ID of Code is mandatory and has to be provided
- g. BookingPackageStartTime\* - has to be in the range between booking arrival and booking departure date

- 
- h. BookingPackageEndTime\*- has to be in the range between booking arrival and booking departure date
  - i. BookingPackageNumberOfAttendees\*
  - j. BookingPackageRate – not possible to override this value
  - k. BookingEvent -the response may contain multiple elements.The data of the linked events is displayed as attributes “attr”
    - a. BookingID
    - b. ID – only needed for update request
    - c. StatusID
    - d. StatusCode
    - e. TypeID
    - f. TypeCode
    - g. EventName
    - h. NumberOfAttendees
    - i. StartTime
    - j. EndTime
    - k. InfoBoard
    - l. BookingPackageID

## BookingEvent

The booked events, the delete command can not be used

- a. BookingEventID- needs only to be provided in the condition of the update request
- b. BookingID \*
- c. BookingEventStatusID or
- d. BookingEventStatusCode \*- either one is mandatory, the ID or the Code
- e. BookingEventTypeID or
- f. BookingEventTypeCode - either one can be used ID or Code
- g. BookingEventName – maximum 60 characters, only required if different than the Event Name of the configured Event Type, if not transmitted or a blank value, then the EventTypeLongDescription is used
- h. BookingEventNumberOfAttendees\*
- i. BookingEventStartTime\* - the start and end time has to be in the range of the Booking Arrival and the Booking Departure Date
- j. BookingEventEndTime\* - the start and end time has to be in the range of the Booking Arrival and the Booking Departure Date
- k. BookingEventInfoboard – maximum 60 characters
- l. BookingEventBookingPackageID- the ID of the Booking Package the Event is linked to
- m. EventFunctionSpace - The request may contain multiple elements.The data is displayed as attributes “attr”
  - a. ID – only required for update
  - b. ConfigID
  - c. ConfigCode
  - d. ConfigSeatingID
  - e. ConfigSeatingCode
  - f. ConfigRateID
  - g. ConfigRateCode
  - h. SpaceStartTime
  - i. SpaceEndTime
  - j. RateValue
- n. EventStandardMenu - The request may contain multiple elements.The data is displayed as attributes “attr”

- 
- a. ID- only required for update
  - b. StandardMenuName
  - c. Description
  - d. ConfigID
  - e. ConfigName
  - f. ConfigRateID
  - g. ConfigRateCode
  - h. StartTime
  - i. EndTime
  - j. MenuQuantity
  - k. RateValue
- o. EventCateringItem - The request may contain multiple elements. The data is displayed as attributes "attr"
- a. ID - only required for update
  - b. Description
  - c. ConfigID
  - d. ConfigName
  - e. ConfigRateID
  - f. ConfigRateCode
  - g. CateringItemName
  - h. StartTime
  - i. EndTime
  - j. Quantity
  - k. RateValue
- p. Note – the note list, the request may contain multiple elements. The data of the notes is displayed as attributes "attr".
- a. ID- only required for update
  - b. CategoryID
  - c. CategoryCode
  - d. Description
  - e. GuestViewable
  - f. TableLinkedTo

## EventFunctionSpace

- a. EventFunctionSpaceID – required only in the update request condition
- b. EventFunctionSpaceEventID\* - mandatory
- c. EventFunctionSpaceConfigID - or
- d. EventFunctionSpaceConfigCode\*- either one is mandatory ID or Code
- e. EventFunctionSpaceConfigSeatingID or
- f. EventFunctionSpaceConfigSeatingCode\* - either one ID or Code is mandatory
- g. EventFunctionSpaceConfigRateID or
- h. EventFunctionSpaceConfigRateCode - either one ID or Code is mandatory
- i. EventFunctionSpaceStartTime – the start time has to be in the range between start and end time of the event
- j. EventFunctionSpaceEndTime - the start time has to be in the range between start and end time of the event
- k. EventFunctionSpaceRateValue – only required if different than the rate entered on the function space definition
- l. Note - the request may contain multiple elements. The data is displayed as attributes in the "attr" element.
  - a. ID – required only in the update request

- b. CategoryID
- c. CategoryCode
- d. Description
- e. GuestViewable
- f. TableLinkedTo

For example:<field name="Note" CategoryID="10" CategoryCode="REN"  
Description="This is a resource note for function space" GuestViewable="1"  
TableLinkedTo="EventFunctionSpace"/>

## EventCateringItem

- a. EventCateringItemID– required only in the update request condition
- b. EventCateringItemEventID\*
- c. EventCateringItemConfigID or
- d. EventCateringItemConfigCode\* - either ID or Code is mandatory
- e. EventCateringItemName – maximum of 60 characters; only required if different than the configured Catering Item LongDescription
- f. EventCateringItemConfigRateID or
- g. EventCateringItemConfigRateCode\* - either ID or Code is mandatory
- h. EventCateringItemStartTime\* - the start time has to be in the range between start and end time of the event
- i. EventCateringItemEndTime\* - the start time has to be in the range between start and end time of the event
- j. EventCateringItemQuantity\*
- k. EventCateringItemRateValue - only required if different than the configured rate
- l. Note - the request may contain multiple elements.The data is displayed as attributes in the “attr” element.
  - a. ID – required only in the update request
  - b. CategoryID
  - c. CategoryCode
  - d. Description
  - e. GuestViewable
  - f. TableLinkedTo

## EventStandardMenu

- a. EventStandardMenuID required only in the update request condition
- b. EventStandardMenuEventID\*
- c. EventStandardMenuConfigID or
- d. EventStandardMenuName\* - maximum of 60 characters; either one ID or Name is mandatory, if the name is blank or not transmitted, the Name of the configured standard menu is used
- e. EventStandardMenuConfigName
- f. EventStandardMenuConfigRateID or
- g. EventStandardMenuConfigRateCode \* -either one is mandatory either ID or Code
- h. EventStandardMenuStartTime\*
- i. EventStandardMenuEndTime\*
- j. EventStandardMenuQuantity\*
- k. EventStandardMenuRateValue – only required if different than the configured rate value
- l. StandardMenuMenuItem – the data of the elements is displayed as attributes “attr”
  - a. ID – required only for update requests
  - b. ConfigID

- c. ConfigName
  - d. LongDescription
  - e. LongDescription2
  - f. LongDescription3
  - g. LongDescription4
  - h. Description
  - i. Origin
  - j. Origin2
  - k. Origin3
  - l. Vintage
  - m. ConfigRateID
  - n. ConfigRateCode
  - o. Quantity
  - p. RateValue
  - q. Included
  - r. Course
  - s. CourseOrder
  - t. Portion
  - u. FoodOrBeverage
  - v. CostPrice
- m. Note - The request may contain multiple elements. The data is displayed as attributes in the "attr" element
- a. ID – only required for update
  - b. CategoryID
  - c. CategoryCode
  - d. Description
  - e. GuestViewable
  - f. TableLinkedTo

## StandardMenuMenuItem

Booked Menu items linked to a standard menu

- a. StandardMenuMenuItemID -only required for update in the condition
- b. StandardMenuMenuItemStandardMenuID or
- c. StandardMenuMenuItemConfigID or
- d. StandardMenuMenuItemConfigName - maximum 55 characters; either one is mandatory, the ID or Name, this is the link to the configured item and is required also for a custom menu item
- e. StandardMenuMenuItemLongDescription – maximum 55 characters; only required if different than the configured menu item long description
- f. StandardMenuMenuItemLongDescription2 - maximum 55 characters; only required if different than the configured menu item long description2
- g. StandardMenuMenuItemLongDescription3- maximum 55 characters;only required if different than the configured menu item long description3
- h. StandardMenuMenuItemLongDescription4 maximum 55 characters; only required if different than the configured menu item long description4
- i. StandardMenuMenuItemDescription – long field; only required if different than the configured menu item long description
- j. StandardMenuMenuItemOrigin - maximum 100 characters; only required if different than the configured menu item origin
- a. StandardMenuMenuItemOrigin2 - maximum 55 characters; only required if different than the configured menu item origin2

- 
- b. StandardMenuMenuItemOrigin3 - maximum 55 characters; only required if different than the configured menu item origin3
  - c. StandardMenuMenuItemVintage – maximum 4 characters; only required if different than the configured menu item
  - d. StandardMenuMenuItemConfigRateID or
  - e. StandardMenuMenuItemConfigRateCode\* - either one, ID or Code is mandatory
  - f. StandardMenuMenuItemQuantity\*
  - g. StandardMenuMenuItemRateValue –only required if different than the configured rate value

## Voucher Redemption

The Request requires at least one of the following licenses:

- XML Interface Standard Edition
- XML Interface Enhanced Edition
- The Request also requires the Voucher Management license.

For voucher redemption, the command -> RedeemVoucher is used.

Following fields should be passed:

- a. VoucherNumber – voucher to redeem
- b. ReservationID or
- c. FinancialAccountID\* - Either one: the reservation ID or the ID of the financial account is allowed and must be specified.
- d. Amount – optional. In case it is not specified, the whole available amount will be redeemed. Partial redemption will be performed only if voucher type is configured to allow this.
- e. ForceRedemption – optional. The flag is valid only for the Value vouchers. If value 1 is sent, the voucher will be redeemed even when there is no consumption. If the “Amount” element is specified in the request for the value voucher, it also works as “ForceRedemption” , then the specified amount is redeemed even if there is no consumptions

The response contains the amount which was redeemed.



---

---

## 6 Conference and Catering General

### ConferenceBooking

Depending on the internal business processes of the site, more fields can be mandatory. Please check with your Oracle Hospitality contact person.

- a. Booking\_CompanyID
- b. Booking\_ContactID – either Company or Contact Id is mandatory, the Id is mandatory if the booking is linked to guest rooms
- c. Booking\_ArrivalDate \*<sup>1</sup>
- d. Booking\_DepartureDate \*
- e. Booking\_Name  
Maximum 60 Characters.
- f. Booking\_ConferenceStatus  
Mandatory if booking contains conference. Supply Booking\_ConferenceStatus (Code) or Booking\_ConferenceStatusID.
- g. Booking\_ConferenceStatusID
- h. Booking\_RoomStatus  
Mandatory if booking contains guest rooms. Supply Booking\_RoomStatus (Code) or Booking\_RoomStatusID
- i. Booking\_RoomStatusID.
- j. Booking\_ConferenceOnly  
1 if booking is conference only. Else leave blank.
- k. Booking\_RoomsOnly  
1 if booking is guest rooms only. Else leave blank
- l. Booking\_ConferenceAndRooms  
1 if booking containing both conference and guest rooms. Else leave blank.
- m. Booking\_Notes
- n. Booking\_NoOfAttendes \*
- o. Booking\_MarketCode \*  
Supply Booking\_MarketCode or Booking\_MarketCodeID
- p. Booking\_MarketCodeID
- q. Booking\_SourceCode \*  
Supply Booking\_MarketCode or Booking\_MarketCodeID
- r. Booking\_SourceCodeID
- s. Booking\_NoteCategory
- t. Booking\_ContactOnSite  
Maximum 60 characters.
- u. Booking\_AccountManagerID
- v. Booking\_AccountManagerCode
- w. Booking\_ChannelCode  
Supply Booking\_ChannelCode or Booking\_ChannelCodeID
- x. Booking\_ChannelCodeID
- y. Booking\_ConferenceManagerID
- z. Booking\_ConferenceManagerCode
- aa. Booking\_DecisionDate
- bb. Booking\_FollowupDate
- cc. Booking\_CutoffDate

---

<sup>1</sup> Fields marked with \* are mandatory.

- dd. Booking\_Infoboard  
Maximum 60 characters
- ee. Booking\_FunctionTypeID
- ff. Booking\_FunctionTypeCode
- gg. Booking\_CurrencyID
- hh. Booking\_CurrencyCode

## BookingPackage

When inserting a package all configured events linked to the package will be inserted automatically. See ConferencePackageList->ConferencePackageEvents.

- a. BookingPackageBookingID\*
- b. BookingPackageName  
Supply the field only to override the Package Name from the package configuration.  
Supply the field only to override the Package Name from the Conference Package configuration.  
If field is not supplied or blank the ConferencePackageName is taken.  
See also ConferencePackage Request.
- c. BookingPackageConfigID
- d. BookingPackageConfigID\*  
Supply BookingPackageConfigID or BookingPackageConfigCode
- e. BookingPackageConfigCode
- f. BookingPackageStartTime\*  
Must fall within Booking\_ArrivalDate & Booking\_DepartureDate
- g. BookingPackageEndTime\*  
Must fall within Booking\_ArrivalDate & Booking\_DepartureDate
- h. BookingPackageNumberOfAttendees\*

Please note: The BookingPackageRateValue can not be overridden

## BookingEvent

Depending on the internal business processes of the site, more fields can be mandatory. Please check with your Oracle Hospitality contact person.

- a. BookingID \*
- b. BookingEventStatusID \*  
Supply BookingEventStatusID or BookingEventStatusCode
- c. BookingEventStatusCode \*
- d. BookingEventTypeID  
Supply BookingEventTypeID or BookingEventTypeCode
- e. BookingEventTypeCode
- f. BookingEventName  
Supply the field only to override the Event Name from the Event Type configuration.  
If the field is not supplied or blank the EventTypeLongDescription is taken  
see also EventType Request.  
Maximum 60 characters
- g. BookingEventNumberOfAttendees\*
- h. BookingEventStartTime\*  
Must fall within Booking\_ArrivalDate & Booking\_DepartureDate
- i. BookingEventEndTime\*  
Must fall within Booking\_ArrivalDate & Booking\_DepartureDate

- 
- j. BookingEventInfoBoard  
Maximum 60 characters
  - k. BookingEventBookingPackageID  
the Id of the BookingPackage to which the event is linked

## EventFunctionSpace

- a. EventFunctionSpaceEventID\*
- b. EventFunctionSpaceConfigID\*  
Supply the EventFunctionSpaceConfigID or the EventFunctionSpaceConfigCode
- c. EventFunctionSpaceConfigCode
- d. EventFunctionSpaceConfigSeatingID\*Supply the EventFunctionSpaceSeatingID or the EventFunctionSpaceConfigSeatingCode
- e. EventFunctionSpaceConfigSeatingCode
- f. EventFunctionSpaceConfigRateID\*  
Supply the EventFunctionSpaceRateID or the EventFunctionSpaceConfigRateCode
- g. EventFunctionSpaceConfigRateCode
- h. EventFunctionSpaceStartTime  
In most cases this is the same start and end time as the BookingEvent.
- i. EventFunctionSpaceEndTime  
Between start and end time of event, in most cases it is the same start and end time as the BookingEvent.
- j. EventFunctionSpaceRateValue  
Supply the field only to override the rate value from the rate configuration.  
If field is not supplied or blank the rate value of the rate configuration is taken. See also FunctionSpaceRate request

## EventCateringItem

- a. EventCateringItemEventID\*
- b. EventCateringItemConfigID\*  
Supply the EventCateringItemConfigID or EventCateringItemConfigCode
- c. EventCateringItemConfigCode
- d. EventCateringItemName  
Supply only the field to override the Catering Item Name from the Catering Item configuration.  
If the field is not supplied or blank the CateringItemLongDescription is taken.  
See also CateringItem Request.  
Maximum 60 Characters
- e. EventCateringItemConfigRateID\*  
Supply the EventCateringItemRateID or the EventCateringItemRateCode
- f. EventCateringItemConfigRateCode
- g. EventCateringItemStartTime\*  
between start and end time of event, in most cases it is the same start and end time as the BookingEvent
- h. EventCateringItemEndTime\*  
between start and end time of event, in most cases it is the same start and end time as the BookingEvent.
- i. EventCateringItemQuantity\*
- j. EventCateringItemRateValue  
Supply only the field to override the rate value from the rate configuration.

---

If the field is not supplied or blank the rate value of the rate configuration is taken. See also CateringItemRate request

## EventStandardMenu

- a. EventStandardMenuEventID\*
- b. EventStandardMenuConfigID\*  
Supply EventStandardMenuConfigID or EventStandardMenuConfigName
- c. EventStandardMenuName  
Supply only the field to override the Name from the standard menu configuration if field is not supplied or blank the StandarMenuLongDescription is taken. See also StandardMenu Request.  
Maximum 60 Characters
- d. EventStandardMenuConfigName
- e. EventStandardMenuConfigRateID\*  
Supply the EventStandardMenuConfigRateID or the EventStandardMenuConfigRateCode
- f. EventStandardMenuConfigRateCode
- g. EventStandardMenuStartTime\*
- h. EventStandardMenuEndTime\*
- i. EventStandardMenuQuantity\*
- j. EventStandardMenuRateValue  
Supply only the field to override the rate value from the rate configuration. If field is not supplied or blank the rate value of the rate configuration is taken. See also StandardMenuRate request

## StandardMenuMenuItem

Booked Menu items linked to a standard menu.

- a. StandardMenuMenuItemStandardMenuID\*
- b. StandardMenuMenuItemConfigID or StandardMenuMenuItemConfigName\* - Either one is mandatory: ID of Name – It is the link to the configured item. Needs to be passed also in the event of a custom menu item
- c. StandardMenuMenuItemLongDescription
- d. StandardMenuMenuItemLongDescription2
- e. StandardMenuMenuItemLongDescription3  
StandardMenuMenuItemLongDescription4  
all description fields support a maximum of 55 characters – pass only the field which should be used as description
- f. StandardMenuMenuItemDescription  
pass only the field which should be used as description. The same applies to MenuItem Request- Long Field
- g. StandardMenuMenuItemOrigin  
maximum 100 characters. Pass only the required field from the menu item configuration. The same applies to MenuItem Request.
- h. StandardMenuMenuItemOrigin2  
maximum 55 characters
- i. StandardMenuMenuItemOrigin3  
maximum 55 characters

- 
- j. StandardMenuMenuItemVintage  
maximum 4 characters. Used only if overriding the menu item configuration is required. The same applies to the MenuItem Request.
  - k. StandardMenuMenuItemConfigRateID or
  - l. StandardMenuMenuItemConfigRateCode\*- either one is mandatory ID or Code
  - m. StandardMenuMenuItemQuantity\*
  - n. StandardMenuMenuItemRateValue- If the field is not passed or blank, the rate value of the rate configuration is used. This applies also the the MenuItemRate request

# 7 Schema FidelioXML.xsd

Schema FidelioXML.xsd

schema location:	<b>FidelioXML.xsd</b>	
targetNamespace:	<b>V8XMLInterface</b>	
Elements	Complex types	Simple types
<b><u>fidelio</u></b>	<b><u>AvailabilityForWebType</u></b>	<b><u>binaryOperation</u></b>
	<b><u>BlockDetailConditionType</u></b>	<b><u>conditionSimpleType</u></b>
	<b><u>BlockHeaderConditionType</u></b>	
	<b><u>ccm book actualsType</u></b>	
	<b><u>ChildrenCategoriesConditionType</u></b>	
	<b><u>commandType</u></b>	
	<b><u>conditionsType</u></b>	
	<b><u>conditionType</u></b>	
	<b><u>ConditionType ConfiguredStatusList</u></b>	
	<a href="#">ConditionType_InfoBoard</a>	
	<b><u>CreditCardTypesConditionType</u></b>	
	<a href="#">crm_postType</a>	
	<b><u>CurrencyListConditionType</u></b>	
	<b><u>DepartmentCodeConditionType</u></b>	
	<b><u>fieldsType</u></b>	
	<b><u>fieldType</u></b>	
	<b><u>HotelAvailabilityType</u></b>	
	<b><u>importType</u></b>	
	<b><u>insertType</u></b>	
	<b><u>linkType</u></b>	
	<b><u>PostngConditionType</u></b>	
	<b><u>postResponseType</u></b>	
	<b><u>postType</u></b>	
	<b><u>ProfileConditionType</u></b>	
	<b><u>queryType</u></b>	
	<b><u>ReportType</u></b>	
	<b><u>requestType</u></b>	
	<b><u>ReservationConditionType</u></b>	
	<b><u>rowsType</u></b>	
	<b><u>rowType</u></b>	
	<b><u>SystemDateConditionType</u></b>	
	<b><u>SystemInformationConditionType</u></b>	
	<b><u>updateType</u></b>	
	<b><u>ValidateCreditCardType</u></b>	

**element fidelio**

<p>diagram</p>	<p>The root element of any XML accepted by V8 XML Interface</p> <p>Status codes are:          OKCommand or request successful          CPosting denied because overwrite CreditLimit is not allowed          IAGuest related requests Invalid account          NGGuest not found          NPPosting denied because NoPost is set          URUnprocessable request, this request cannot be carried out, no retry          IRIInvalid request or unsupported request type          ICInvalid credentials. Wrong username or password          ADAAccess denied. Not enough user rights or trying to make batch update when it is not allowed by configuration</p> <p>Message is optional attribute used when Status is not OK</p>												
<p>namespace</p>	<p>V8XMLInterface</p>												
<p>children</p>	<p><b><u>request response</u></b></p>												
<p>attributes</p>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation documentation</th> </tr> </thead> <tbody> <tr> <td>Version</td> <td>xs:NMTOKEN</td> <td>required</td> <td></td> <td></td> <td>Specifies the version of the interface. Could be changed and in the future accept more values</td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation documentation	Version	xs:NMTOKEN	required			Specifies the version of the interface. Could be changed and in the future accept more values
Name	Type	Use	Default	Fixed	Annotation documentation								
Version	xs:NMTOKEN	required			Specifies the version of the interface. Could be changed and in the future accept more values								
<p>annotation</p>	<p>documentation The root element of any XML accepted by V8 XML Interface</p>												
<p>source</p>	<pre> &lt;xs:element name="fidelio"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The root element of any XML accepted by V8 XML Interface&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:choice&gt;       &lt;xs:element name="request"&gt;         &lt;xs:complexType&gt;           &lt;xs:complexContent&gt;             &lt;xs:extension base="requestType"&gt;               &lt;xs:attribute name="ID" type="xs:string" use="required"/&gt;               &lt;xs:attribute name="UserName" type="xs:string" use="required"/&gt;             &lt;/xs:extension&gt;           &lt;/xs:complexContent&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;     &lt;/xs:choice&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>												

```

<xs:attribute name="Password" type="xs:string" use="required"/>
<xs:attribute name="Vendor" type="xs:string" use="optional">
  <xs:annotation>
    <xs:documentation>Required when one of the licenses
CRSXMLONEWAY_NCM, CRSXMLTWOWAY_NCM, CRSXMLONEWAY_FIDELIO8 or
ftCRSXMLTWOWAY_FIDELIO8 is used</xs:documentation>
  </xs:annotation>
</xs:attribute>
</xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:element>
<xs:element name="response">
  <xs:annotation>
    <xs:documentation>Status codes are:
OK      Command or request successful
CO      Posting denied because overwrite CreditLimit is not allowed
IA      Guest related requests Invalid account
NG      Guest not found
NP      Posting denied because NoPost is set
UR      Unprocessable request, this request cannot be carried out, no retry
IR      Invalid request or unsupported request type
IC      Invalid credentials. Wrong username or password
AD      Access denied. Not enough user rights or trying to make batch update when it is not allowed
be configuration

Message is optional attribute used when Status is not OK</xs:documentation>
  </xs:annotation>
<xs:complexType>
  <xs:complexContent>
    <xs:extension base="postResponseType">
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="queryResponse" minOccurs="0" maxOccurs="unbounded">
          <xs:annotation>
            <xs:documentation>Provided only when request contains the "query" section
          </xs:documentation>
        </xs:element>
      </xs:choice>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
  <xs:annotation>
    <xs:documentation>Attribute name exists only when was passed in the request</xs:documentation>
  </xs:annotation>
<xs:complexType>
  <xs:all>
    <xs:element name="rows" type="rowsType"/>
  </xs:all>
  <xs:attribute name="Name" type="xs:string" use="optional"/>
</xs:complexType>
</xs:element>
<xs:element name="insertResponse" minOccurs="0" maxOccurs="unbounded">

```



```

<xs:annotation>
  <xs:documentation>Provided when request contains the "insert" section and contains the IDs
of the created objects
Attribute name exists only when was passed in the request</xs:documentation>
</xs:annotation>
<xs:complexType>
  <xs:all>
    <xs:element name="rows">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ProfileID" type="xs:integer" minOccurs="0"/>
          <xs:element name="FANumber" minOccurs="0"/>
          <xs:element name="GuestNum" type="xs:integer" minOccurs="0"/>
          <xs:element name="ProfileGlobalID" type="xs:integer" minOccurs="0"/>
          <xs:element name="LastUpdateTime" type="xs:string" minOccurs="0"/>
          <xs:choice minOccurs="0">
            <xs:element name="Addresses">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Address" maxOccurs="unbounded">
                    <xs:complexType>
                      <xs:attribute name="initID" type="xs:integer" use="required"/>
                      <xs:attribute name="newID" type="xs:integer" use="required"/>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:choice>
          <xs:choice minOccurs="0">
            <xs:element name="Communications">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Communication" maxOccurs="unbounded">
                    <xs:complexType>
                      <xs:attribute name="newID" type="xs:integer" use="required"/>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:choice>
          <xs:choice minOccurs="0">
            <xs:element name="Notes">

```

```

<xs:complexType>
  <xs:sequence>
    <xs:element name="Note" maxOccurs="unbounded">
      <xs:complexType>
        <xs:attribute name="newID" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:choice>
<xs:choice minOccurs="0">
  <xs:element name="GuestComments">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="GuestComment" maxOccurs="unbounded">
          <xs:complexType>
            <xs:attribute name="newID" type="xs:integer" use="required"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="TaskID" type="xs:integer" minOccurs="0"/>
  <xs:element name="ActivityID" type="xs:integer" minOccurs="0"/>
  <xs:element name="GroupID" type="xs:integer" minOccurs="0"/>
  <xs:element name="GroupTemplateID" type="xs:integer" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:all>
  <xs:attribute name="Name" type="xs:string" use="optional"/>
</xs:complexType>
</xs:element>
<xs:element name="updateResponse" minOccurs="0" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>Provided when request contains the "update" section and contains
    requested data of the updated objects created objects
  </xs:documentation>
  Attribute name exists only when was passed in the request</xs:documentation>
  </xs:annotation>
</xs:complexType>
<xs:all>
  <xs:element name="rows" type="rowsType"/>

```

```

        </xs:all>
        <xs:attribute name="Name" type="xs:string" use="optional"/>
    </xs:complexType>
</xs:element>
</xs:choice>
<xs:attribute name="ID" type="xs:string" use="required"/>
<xs:attribute name="Status" use="required">
    <xs:simpleType>
        <xs:restriction base="xs:NMTOKENS">
            <xs:enumeration value="OK"/>
            <xs:annotation>
                <xs:documentation>Command or request successful</xs:documentation>
            </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="CO"/>
            <xs:enumeration value="IA"/>
            <xs:enumeration value="NG"/>
            <xs:enumeration value="NP"/>
            <xs:enumeration value="UR"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
<xs:attribute name="Message" type="xs:string" use="optional"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:element>
</xs:choice>
<xs:attribute name="Version" use="required">
    <xs:annotation>
        <xs:documentation>Specifies the version of the interface. Could be changed and in the future accept more values</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="1.0.0"/>
            <xs:enumeration value="1.0.1"/>
            <xs:enumeration value="1.0.2"/>
            <xs:enumeration value="1.0.3"/>
            <xs:enumeration value="1.0.4"/>
            <xs:enumeration value="1.0.5"/>
            <xs:enumeration value="1.0.6"/>
            <xs:enumeration value="1.0.7"/>
            <xs:enumeration value="1.1.0"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>

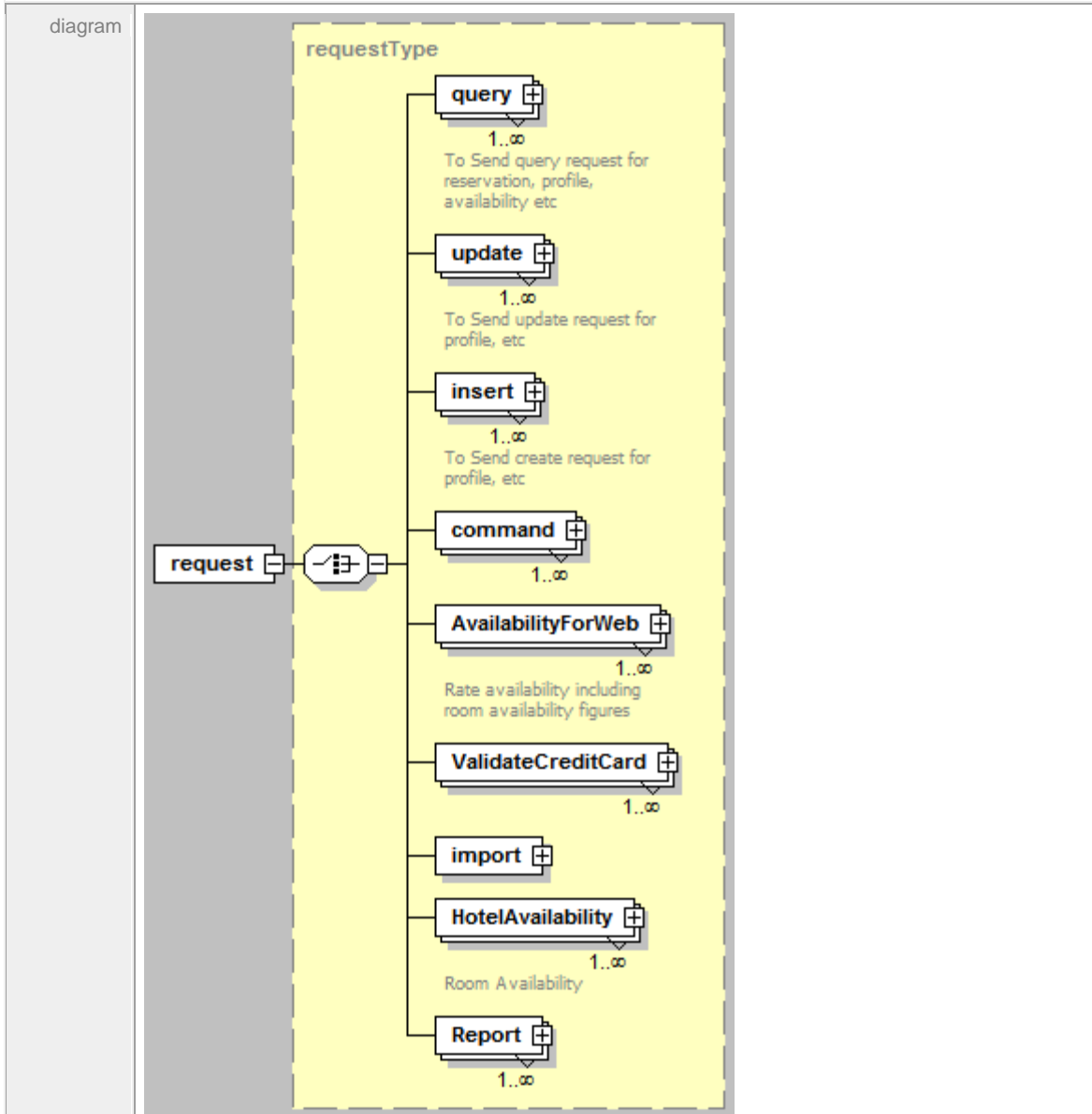
```

```

</xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>

```

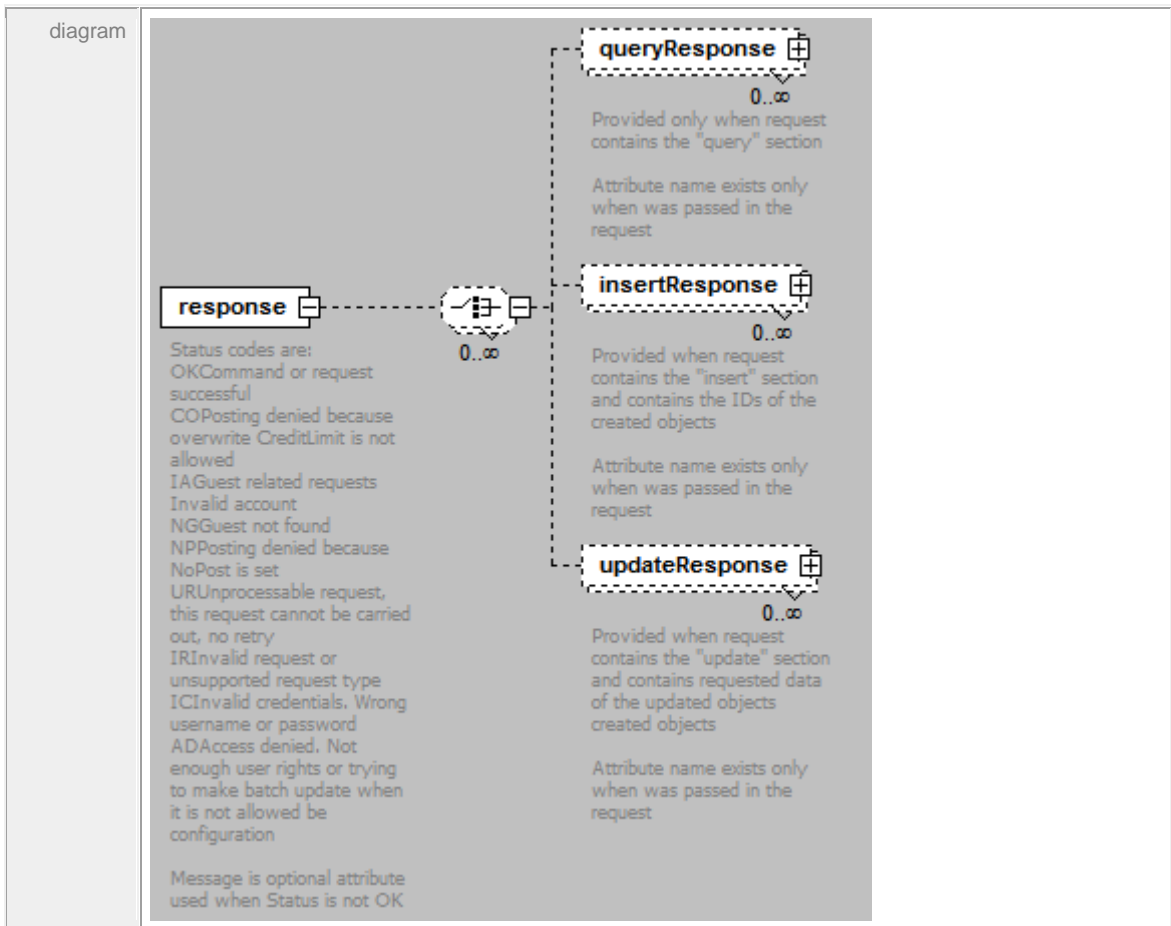
**element fidelio/request**



namespace	V8XMLInterface					
type	extension of <b>requestType</b>					
children	<b><u>query</u> <u>update</u> <u>insert</u> <u>command</u> <u>AvailabilityForWeb</u> <u>ValidateCreditCard</u> <u>import</u> <u>HotelAvailability</u> <u>Report</u></b>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	ID	xs:string	required			
	UserName	xs:string	required			
	Password	xs:string	required			

	Vendor    xs:string    optional	documentation Required when one of the licenses CRSXMLONEWAY_NCM, CRSXMLTWOWAY_NCM, CRSXMLONEWAY_FIDELIO8 or ftCRSXMLTWOWAY_FIDELIO8 is used
source	<pre> &lt;xs:element name="request"&gt;   &lt;xs:complexType&gt;     &lt;xs:complexContent&gt;       &lt;xs:extension base="requestType"&gt;         &lt;xs:attribute name="ID" type="xs:string" use="required"/&gt;         &lt;xs:attribute name="UserName" type="xs:string" use="required"/&gt;         &lt;xs:attribute name="Password" type="xs:string" use="required"/&gt;         &lt;xs:attribute name="Vendor" type="xs:string" use="optional"/&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;Required when one of the licenses CRSXMLONEWAY_NCM, CRSXMLTWOWAY_NCM, CRSXMLONEWAY_FIDELIO8 or ftCRSXMLTWOWAY_FIDELIO8 is used&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:extension&gt;     &lt;/xs:complexContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>	

**element fidelio/response**



namespace	V8XMLInterface					
type	extension of <b>postResponseType</b>					
children	<b>queryResponse insertResponse updateResponse</b>					
attributes	Name ID	Type xs:string	Use required	Default	Fixed	Annotation
	Status	xs:NMTOKENS	required			
	Message	xs:string	optional			
annotation	<p>documentation Status codes are:OK Command or request successfulCO Posting denied because overwrite CreditLimit is not allowedIA Guest related requests Invalid accountNG Guest not foundNP Posting denied because NoPost is setUR Unprocessable request, this request cannot be carried out, no retryIR Invalid request or unsupported request typeIC Invalid credentials. Wrong username or passwordAD Access denied. Not enough user rights or trying to make batch update when it is not allowed be configurationMessage is optional attribute used when Status is not OK</p>					
source	<pre> &lt;xs:element name="response"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Status codes are:OK Command or request successfulCO Posting denied because overwrite CreditLimit is not allowedIA Guest related requests Invalid accountNG Guest not foundNP Posting denied because NoPost is setURUnprocessable request, this request cannot be carried out, no retryIR Invalid request or unsupported request typeIC Invalid credentials. Wrong username or passwordADAccess denied. Not enough user rights or trying to make batch update when it is not allowed be configurationMessage is optional attribute used when Status is not OK&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:complexType&gt; &lt;xs:complexContent&gt; &lt;xs:extension base="postResponseType"&gt; &lt;xs:choice minOccurs="0" maxOccurs="unbounded"&gt; &lt;xs:element name="queryResponse" minOccurs="0" maxOccurs="unbounded"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Provided only when request contains the "query" sectionAttribute name exists only when was passed in the request&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:complexType&gt; &lt;xs:all&gt; &lt;xs:element name="rows" type="rowsType"/&gt; &lt;/xs:all&gt; &lt;xs:attribute name="Name" type="xs:string" use="optional"/&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;xs:element name="insertResponse" minOccurs="0" maxOccurs="unbounded"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Provided when request contains the "insert" section and contains the IDs of the created objectsAttribute name exists only when was passed in the request&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:complexType&gt; &lt;xs:all&gt; &lt;xs:element name="rows"&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="ProfileID" type="xs:integer" minOccurs="0"/&gt; &lt;xs:element name="FANumber" minOccurs="0"/&gt; &lt;xs:element name="GuestNum" type="xs:integer" minOccurs="0"/&gt; &lt;xs:element name="ProfileGlobalID" type="xs:integer" minOccurs="0"/&gt; &lt;xs:element name="LastUpdateTime" type="xs:string" minOccurs="0"/&gt; &lt;xs:choice minOccurs="0"&gt; &lt;xs:element name="Addresses"&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="Address" maxOccurs="unbounded"&gt; &lt;xs:complexType&gt; &lt;xs:attribute name="initID" type="xs:integer" use="required"/&gt; &lt;xs:attribute name="newID" type="xs:integer" use="required"/&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:choice&gt; &lt;xs:choice minOccurs="0"&gt; &lt;xs:element name="Communications"&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="Communication" maxOccurs="unbounded"&gt; &lt;xs:attribute name="newID" type="xs:integer" use="required"/&gt; &lt;/xs:element&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:choice&gt; &lt;xs:choice minOccurs="0"&gt; &lt;xs:element name="Notes"&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="Note" maxOccurs="unbounded"&gt; &lt;xs:attribute name="newID" type="xs:integer" use="required"/&gt; &lt;/xs:element&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:choice&gt; &lt;xs:element name="TaskID" type="xs:integer" minOccurs="0"/&gt; &lt;xs:element name="ActivityID" type="xs:integer" minOccurs="0"/&gt; &lt;xs:element name="GroupID" type="xs:integer" minOccurs="0"/&gt; &lt;xs:element name="GroupTemplateID" type="xs:integer" minOccurs="0"/&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:all&gt; &lt;xs:attribute name="Name" type="xs:string" use="optional"/&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;xs:element name="updateResponse" minOccurs="0" maxOccurs="unbounded"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Provided when request contains the "update" section and contains requested data of the updated objects created objectsAttribute name exists only when was passed in the request&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:complexType&gt; &lt;xs:all&gt; &lt;xs:element name="rows" type="rowsType"/&gt; &lt;/xs:all&gt; &lt;xs:attribute name="Name" type="xs:string" use="optional"/&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:choice&gt; &lt;xs:attribute name="ID" type="xs:string" use="required"/&gt; &lt;xs:attribute </pre>					

<pre> name="Status" use="required"&gt; &lt;xs:enumeration value="OK"&gt; &lt;xs:restriction base="xs:NMTOKENS"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Command or request successful&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:enumeration&gt; &lt;xs:enumeration value="CO"/&gt; &lt;xs:enumeration value="IA"/&gt; &lt;xs:enumeration value="NG"/&gt; &lt;xs:enumeration value="NP"/&gt; &lt;xs:enumeration value="UR"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; &lt;xs:attribute name="Message" type="xs:string" use="optional"/&gt; &lt;/xs:extension&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt; </pre>
---

### element fidelio/response/queryResponse

diagram													
namespace	V8XMLInterface												
children	<u>rows</u>												
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	Name	xs:string	optional			
Name	Type	Use	Default	Fixed	Annotation								
Name	xs:string	optional											
annotation	documentation Provided only when request contains the "query" sectionAttribute name exists only when was passed in the request												
source	<pre> &lt;xs:element name="queryResponse" minOccurs="0" maxOccurs="unbounded"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Provided only when request contains the "query" sectionAttribute name exists only when was passed in the request&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:complexType&gt; &lt;xs:all&gt; &lt;xs:element name="rows" type="rowsType"/&gt; &lt;/xs:all&gt; &lt;xs:attribute name="Name" type="xs:string" use="optional"/&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt; </pre>												

### element fidelio/response/queryResponse/rows

diagram	
namespace	V8XMLInterface
type	<u>rowsType</u>
children	<u>row</u>
source	<pre> &lt;xs:element name="rows" type="rowsType"/&gt; </pre>

## element fidelio/response/insertResponse


diagram	<p>Provided when request contains the "insert" section and contains the IDs of the created objects</p> <p>Attribute name exists only when was passed in the request</p>												
namespace	V8XMLInterface												
children	<b>rows</b>												
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	Name	xs:string	optional			
Name	Type	Use	Default	Fixed	Annotation								
Name	xs:string	optional											
annotation	documentation Provided when request contains the "insert" section and contains the IDs of the created objectsAttribute name exists only when was passed in the request												
source	<pre> &lt;xs:element name="insertResponse" minOccurs="0" maxOccurs="unbounded"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Provided when request contains the "insert" section and contains the IDs of the created objectsAttribute name exists only when was passed in the request&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:complexType&gt; &lt;xs:all&gt; &lt;xs:element name="rows"&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="ProfileID" type="xs:integer" minOccurs="0"/&gt; &lt;xs:element name="FANumber" minOccurs="0"/&gt; &lt;xs:element name="GuestNum" type="xs:integer" minOccurs="0"/&gt; &lt;xs:element name="ProfileGlobalID" type="xs:integer" minOccurs="0"/&gt; &lt;xs:element name="LastUpdateTime" type="xs:string" minOccurs="0"/&gt; &lt;xs:choice minOccurs="0"&gt; &lt;xs:element name="Addresses"&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="Address" maxOccurs="unbounded"&gt; &lt;xs:complexType&gt; &lt;xs:attribute name="initID" type="xs:integer" use="required"/&gt; &lt;xs:attribute name="newID" type="xs:integer" use="required"/&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:choice&gt; &lt;xs:choice minOccurs="0"&gt; &lt;xs:element name="Communications"&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="Communication" maxOccurs="unbounded"&gt; &lt;xs:complexType&gt; &lt;xs:attribute name="newID" type="xs:integer" use="required"/&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:choice&gt; &lt;xs:choice minOccurs="0"&gt; &lt;xs:element name="Notes"&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="Note" maxOccurs="unbounded"&gt; &lt;xs:complexType&gt; &lt;xs:attribute name="newID" type="xs:integer" use="required"/&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:choice&gt; &lt;xs:element name="TaskID" type="xs:integer" minOccurs="0"/&gt; &lt;xs:element name="ActivityID" type="xs:integer" minOccurs="0"/&gt; &lt;xs:element name="GroupID" type="xs:integer" minOccurs="0"/&gt; &lt;xs:element name="GroupTemplatelD" type="xs:integer" minOccurs="0"/&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:all&gt; &lt;xs:attribute name="Name" type="xs:string" use="optional"/&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt; </pre>												



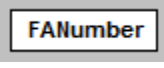
element fidelio/response/insertResponse/rows

<p>diagram</p>	
<p>namespace</p>	<p>V8XMLInterface</p>
<p>children</p>	<p><u>ProfileID</u> <u>FANumber</u> <u>GuestNum</u> <u>ProfileGlobalID</u> <u>LastUpdateTime</u> <u>Addresses</u> <u>Communications</u> <u>Notes</u> <u>TaskID</u> <u>ActivityID</u> <u>GroupID</u> <u>GroupTemplateID</u></p>
<p>source</p>	<pre> &lt;xs:element name="rows"&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="ProfileID" type="xs:integer" minOccurs="0"/&gt; &lt;xs:element name="FANumber" minOccurs="0"/&gt; &lt;xs:element name="GuestNum" type="xs:integer" minOccurs="0"/&gt; &lt;xs:element name="ProfileGlobalID" type="xs:integer" minOccurs="0"/&gt; &lt;xs:element name="LastUpdateTime" type="xs:string" minOccurs="0"/&gt; &lt;xs:choice minOccurs="0"&gt; &lt;xs:element name="Addresses"&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="Address" maxOccurs="unbounded"&gt; &lt;xs:complexType&gt; &lt;xs:attribute name="initID" type="xs:integer" use="required"/&gt; &lt;xs:attribute name="newID" type="xs:integer" use="required"/&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:choice&gt; &lt;xs:choice minOccurs="0"&gt; &lt;xs:element name="Communications"&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="Communication" maxOccurs="unbounded"&gt; &lt;xs:complexType&gt; &lt;xs:attribute name="newID" type="xs:integer" use="required"/&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:choice&gt; &lt;xs:choice minOccurs="0"&gt; &lt;xs:element name="Notes"&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="Note" maxOccurs="unbounded"&gt; &lt;xs:complexType&gt; &lt;xs:attribute name="newID" type="xs:integer" use="required"/&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:choice&gt; &lt;xs:element name="TaskID" type="xs:integer" minOccurs="0"/&gt; &lt;xs:element name="ActivityID" type="xs:integer" minOccurs="0"/&gt; &lt;xs:element name="GroupID" type="xs:integer" minOccurs="0"/&gt; &lt;xs:element name="GroupTemplateID" type="xs:integer" minOccurs="0"/&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt; </pre>


**element** fidelio/response/insertResponse/rows/ProfileID

diagram	
namespace	V8XMLInterface
type	<b>xs:integer</b>
source	<code>&lt;xs:element name="ProfileID" type="xs:integer" minOccurs="0"/&gt;</code>


**element** fidelio/response/insertResponse/rows/FANumber

diagram	
namespace	V8XMLInterface
source	<code>&lt;xs:element name="FANumber" minOccurs="0"/&gt;</code>


**element** fidelio/response/insertResponse/rows/GuestNum

diagram	
namespace	V8XMLInterface
type	<b>xs:integer</b>
source	<code>&lt;xs:element name="GuestNum" type="xs:integer" minOccurs="0"/&gt;</code>

**element** fidelio/response/insertResponse/rows/ProfileGlobalID

diagram	
namespace	V8XMLInterface
type	<b>xs:integer</b>
source	<code>&lt;xs:element name="ProfileGlobalID" type="xs:integer" minOccurs="0"/&gt;</code>

**element** fidelio/response/insertResponse/rows/LastUpdateTime

diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>
source	<code>&lt;xs:element name="LastUpdateTime" type="xs:string" minOccurs="0"/&gt;</code>

**element** fidelio/response/insertResponse/rows/Addresses

diagram	
namespace	V8XMLInterface

children	<b>Address</b>
source	<code>&lt;xs:element name="Addresses"&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="Address" maxOccurs="unbounded"&gt; &lt;xs:complexType&gt; &lt;xs:attribute name="initID" type="xs:integer" use="required"/&gt; &lt;xs:attribute name="newID" type="xs:integer" use="required"/&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt;</code>

**element** `fidelio/response/insertResponse/rows/Addresses/Address`

diagram																			
namespace	V8XMLInterface																		
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>initID</td> <td>xs:integer</td> <td>required</td> <td></td> <td></td> <td></td> </tr> <tr> <td>newID</td> <td>xs:integer</td> <td>required</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	initID	xs:integer	required				newID	xs:integer	required			
Name	Type	Use	Default	Fixed	Annotation														
initID	xs:integer	required																	
newID	xs:integer	required																	
source	<code>&lt;xs:element name="Address" maxOccurs="unbounded"&gt; &lt;xs:complexType&gt; &lt;xs:attribute name="initID" type="xs:integer" use="required"/&gt; &lt;xs:attribute name="newID" type="xs:integer" use="required"/&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt;</code>																		

**element** `fidelio/response/insertResponse/rows/Communications`

diagram	
namespace	V8XMLInterface
children	<b>Communication</b>
source	<code>&lt;xs:element name="Communications"&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="Communication" maxOccurs="unbounded"&gt; &lt;xs:complexType&gt; &lt;xs:attribute name="newID" type="xs:integer" use="required"/&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt;</code>

**element** `fidelio/response/insertResponse/rows/Communications/Communication`

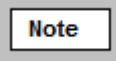
diagram													
namespace	V8XMLInterface												
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>newID</td> <td>xs:integer</td> <td>required</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	newID	xs:integer	required			
Name	Type	Use	Default	Fixed	Annotation								
newID	xs:integer	required											
source	<code>&lt;xs:element name="Communication" maxOccurs="unbounded"&gt; &lt;xs:complexType&gt; &lt;xs:attribute name="newID" type="xs:integer" use="required"/&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt;</code>												

**element** `fidelio/response/insertResponse/rows/Notes`


diagram	
namespace	V8XMLInterface
children	<b>Note</b>

source	<pre>&lt;xs:element name="Notes"&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="Note" maxOccurs="unbounded"&gt; &lt;xs:complexType&gt; &lt;xs:attribute name="newID" type="xs:integer" use="required"/&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt;</pre>
--------	---


**element** fidelio/response/insertResponse/rows/Notes/Note

diagram													
namespace	V8XMLInterface												
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>newID</td> <td>xs:integer</td> <td>required</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	newID	xs:integer	required			
Name	Type	Use	Default	Fixed	Annotation								
newID	xs:integer	required											
source	<pre>&lt;xs:element name="Note" maxOccurs="unbounded"&gt; &lt;xs:complexType&gt; &lt;xs:attribute name="newID" type="xs:integer" use="required"/&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt;</pre>												

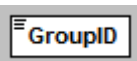
**element** fidelio/response/insertResponse/rows/TaskID

diagram	
namespace	V8XMLInterface
type	<b>xs:integer</b>
source	<pre>&lt;xs:element name="TaskID" type="xs:integer" minOccurs="0"/&gt;</pre>


**element** fidelio/response/insertResponse/rows/ActivityID

diagram	
namespace	V8XMLInterface
type	<b>xs:integer</b>
source	<pre>&lt;xs:element name="ActivityID" type="xs:integer" minOccurs="0"/&gt;</pre>

**element** fidelio/response/insertResponse/rows/GroupID

diagram	
namespace	V8XMLInterface
type	<b>xs:integer</b>
source	<pre>&lt;xs:element name="GroupID" type="xs:integer" minOccurs="0"/&gt;</pre>

**element** fidelio/response/insertResponse/rows/GroupTemplateID

diagram	
namespace	V8XMLInterface
type	<b>xs:integer</b>
source	<pre>&lt;xs:element name="GroupTemplateID" type="xs:integer" minOccurs="0"/&gt;</pre>

**element** fidelio/response/updateResponse

diagram	<p>Provided when request contains the "update" section and contains requested data of the updated objects created objects</p> <p>Attribute name exists only when was passed in the request</p>					
namespace	V8XMLInterface					
children	<b>rows</b>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	Name	xs:string	optional			
annotation	documentation	Provided when request contains the "update" section and contains requested data of the updated objects created objectsAttribute name exists only when was passed in the request				
source	<pre>&lt;xs:element name="updateResponse" minOccurs="0" maxOccurs="unbounded"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Provided when request contains the "update" section and contains requested data of the updated objects created objectsAttribute name exists only when was passed in the request&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:complexType&gt; &lt;xs:all&gt; &lt;xs:element name="rows" type="rowsType"/&gt; &lt;/xs:all&gt; &lt;xs:attribute name="Name" type="xs:string" use="optional"/&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt;</pre>					

**element** fidelio/response/updateResponse/rows

diagram						
namespace	V8XMLInterface					
type	<b>rowsType</b>					
children	<b>row</b>					
source	<pre>&lt;xs:element name="rows" type="rowsType"/&gt;</pre>					

**complexType** AvailabilityForWebType

diagram						
namespace	V8XMLInterface					
children	<b>fields</b>					
used by	element <b>requestType/AvailabilityForWeb</b>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	Name	xs:string	optional			
	Language	xs:string	optional			

	TranslationTable xs:string optional	documentation If present in the request, short descriptions in request and response will be translated based on the specified translation table
source	<pre>&lt;xs:complexType name="AvailabilityForWebType"&gt; &lt;xs:all&gt; &lt;xs:element name="fields" type="fieldsType"/&gt; &lt;/xs:all&gt; &lt;xs:attribute name="Name" type="xs:string" use="optional"/&gt; &lt;xs:attribute name="Language" type="xs:string" use="optional"/&gt; &lt;xs:attribute name="TranslationTable" type="xs:string" use="optional"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;If present in the request, short descriptions in request and response will be translated based on the specified translation table&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;&lt;/xs:complexType&gt;</pre>	

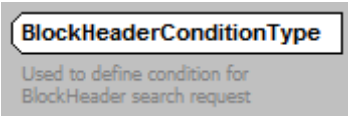
### element AvailabilityForWebType/fields

diagram						
namespace	V8XMLInterface					
type	<b>fieldsType</b>					
children	<b>field</b>					
source	<pre>&lt;xs:element name="fields" type="fieldsType"/&gt;</pre>					

### complexType BlockDetailConditionType

diagram						
namespace	V8XMLInterface					
type	restriction of <b>conditionType</b>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:NMTOKEN	required			
	operation	binaryOperation	required			
	dataType	xs:NMTOKEN	optional			
annotation	documentation Used to define condition for BlockDetail search request					
source	<pre>&lt;xs:complexType name="BlockDetailConditionType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Used to define condition for BlockDetail search request&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:simpleContent&gt; &lt;xs:restriction base="conditionType"&gt; &lt;xs:attribute name="name" use="required"&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:NMTOKEN"&gt; &lt;xs:enumeration value="BlockHeaderID"/&gt; &lt;xs:enumeration value="BlockDetailStartDate"/&gt; &lt;xs:enumeration value="BlockDetailEndDate"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleContent&gt;&lt;/xs:complexType&gt;</pre>					

## complexType BlockHeaderConditionType

diagram						
namespace	V8XMLInterface					
type	restriction of <u>conditionType</u>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:NMTOKEN	required			
	operation	binaryOperation	required			
	dataType	xs:NMTOKEN	optional			
annotation	documentation Used to define condition for BlockHeader search request					
source	<pre> &lt;xs:complexType name="BlockHeaderConditionType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Used to define condition for BlockHeader search request&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:simpleContent&gt; &lt;xs:restriction base="conditionType"&gt; &lt;xs:attribute name="name" use="required"&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:NMTOKEN"&gt; &lt;xs:enumeration value="BlockHeaderID"/&gt; &lt;xs:enumeration value="BlockHeaderCode"/&gt; &lt;xs:enumeration value="BlockHeaderName"/&gt; &lt;xs:enumeration value="BlockHeaderFrom"/&gt; &lt;xs:enumeration value="BlockHeaderTo"/&gt; &lt;xs:enumeration value="BlockHeaderReservationStatus"/&gt; &lt;xs:enumeration value="BlockHeaderOpenForPickup"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleContent&gt;&lt;/xs:complexType&gt; </pre>					

## complexType ccm\_book\_actualsType

diagram	
namespace	V8XMLInterface
children	<b><u>bookingID event_id postingDate menu_id menu_item_id misc_item_id function_space_id ResourceName DepartmentCode count total_amount OverrideReason UserName Covers</u></b>
used by	element <b><u>postType/ccm_book_actuals</u></b>
annotation	documentation POS posting
source	<pre> &lt;xs:complexType name="ccm_book_actualsType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;POS posting&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:all&gt; &lt;xs:element name="bookingID" type="xs:integer"/&gt; &lt;xs:element name="event_id" type="xs:integer"/&gt; &lt;xs:element name="postingDate" type="xs:string"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Date in format DD.MM.YYYY&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:element&gt; &lt;xs:element name="menu_id" type="xs:integer" minOccurs="0"/&gt; &lt;xs:element name="menu_item_id" type="xs:integer" minOccurs="0"/&gt; &lt;xs:element name="misc_item_id" type="xs:integer" minOccurs="0"/&gt; &lt;xs:element name="function_space_id" type="xs:integer" minOccurs="0"/&gt; &lt;xs:element name="ResourceName" type="xs:string"/&gt; &lt;xs:element name="DepartmentCode" type="xs:string"/&gt; &lt;xs:element name="count" type="xs:integer"/&gt; &lt;xs:element name="total_amount" type="xs:string"/&gt; &lt;xs:element name="OverrideReason" type="xs:string" minOccurs="0"/&gt; &lt;xs:element name="UserName" type="xs:string"/&gt; &lt;xs:element name="Covers" type="xs:integer" minOccurs="0"/&gt; &lt;/xs:all&gt;&lt;/xs:complexType&gt; </pre>


## element ccm\_book\_actualsType/bookingID

diagram	
namespace	V8XMLInterface

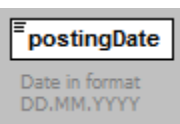


type	<b>xs:integer</b>
source	<code>&lt;xs:element name="bookingID" type="xs:integer"/&gt;</code>

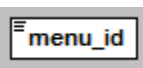
#### element ccm\_book\_actualsType/event\_id

diagram	
namespace	V8XMLInterface
type	<b>xs:integer</b>
source	<code>&lt;xs:element name="event_id" type="xs:integer"/&gt;</code>

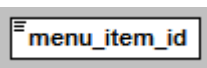
#### element ccm\_book\_actualsType/postingDate

diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>
annotation	documentation Date in format DD.MM.YYYY
source	<code>&lt;xs:element name="postingDate" type="xs:string"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Date in format DD.MM.YYYY&lt;/xs:documentation&gt; &lt;/xs:annotation&gt;&lt;/xs:element&gt;</code>


#### element ccm\_book\_actualsType/menu\_id

diagram	
namespace	V8XMLInterface
type	<b>xs:integer</b>
source	<code>&lt;xs:element name="menu_id" type="xs:integer" minOccurs="0"/&gt;</code>

#### element ccm\_book\_actualsType/menu\_item\_id


diagram	
namespace	V8XMLInterface
type	<b>xs:integer</b>
source	<code>&lt;xs:element name="menu_item_id" type="xs:integer" minOccurs="0"/&gt;</code>

#### element ccm\_book\_actualsType/misc\_item\_id

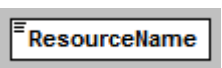
diagram	
namespace	V8XMLInterface
type	<b>xs:integer</b>

source	<code>&lt;xs:element name="misc_item_id" type="xs:integer" minOccurs="0"/&gt;</code>
--------	--


**element ccm\_book\_actualsType/function\_space\_id**

diagram	
namespace	V8XMLInterface
type	<b>xs:integer</b>
source	<code>&lt;xs:element name="function_space_id" type="xs:integer" minOccurs="0"/&gt;</code>

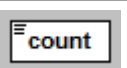
**element ccm\_book\_actualsType/ResourceName**

diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>
source	<code>&lt;xs:element name="ResourceName" type="xs:string"/&gt;</code>

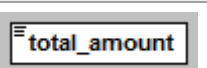
**element ccm\_book\_actualsType/DepartmentCode**

diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>
source	<code>&lt;xs:element name="DepartmentCode" type="xs:string"/&gt;</code>

**element ccm\_book\_actualsType/count**

diagram	
namespace	V8XMLInterface
type	<b>xs:integer</b>
source	<code>&lt;xs:element name="count" type="xs:integer"/&gt;</code>

**element ccm\_book\_actualsType/total\_amount**

diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>
source	<code>&lt;xs:element name="total_amount" type="xs:string"/&gt;</code>

**element** ccm\_book\_actualsType/OverrideReason

diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>
source	<code>&lt;xs:element name="OverrideReason" type="xs:string" minOccurs="0"/&gt;</code>

**element** ccm\_book\_actualsType/UserName

diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>
source	<code>&lt;xs:element name="UserName" type="xs:string"/&gt;</code>

**element** ccm\_book\_actualsType/Covers

diagram	
namespace	V8XMLInterface
type	<b>xs:integer</b>
source	<code>&lt;xs:element name="Covers" type="xs:integer" minOccurs="0"/&gt;</code>

**complexType** ChildrenCategoriesConditionType

diagram	<p>ChildrenCategoriesConditionTy... Used to define condition for Children Categories search request.</p>					
namespace	V8XMLInterface					
type	restriction of <b>conditionType</b>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:NMTOKEN	required			
	operation	binaryOperation	required			
	dataType	xs:NMTOKEN	optional			
annotation	documentation Used to define condition for Children Categories search request.					
source	<pre>&lt;xs:complexType name="ChildrenCategoriesConditionType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Used to define condition for Children Categories search request.&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:simpleContent&gt; &lt;xs:restriction base="conditionType"&gt; &lt;xs:attribute name="name" use="required"&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:NMTOKEN"&gt; &lt;xs:enumeration value="ChildrenCategoriesCode"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleContent&gt;&lt;/xs:complexType&gt;</pre>					

## complexType commandType

diagram	
namespace	V8XMLInterface
children	<b><u>post</u></b> <b><u>checkIn</u></b> <b><u>checkOut</u></b> <b><u>cancel</u></b> <b><u>resetProfilePassword</u></b>
used by	element <b><u>requestType/command</u></b>
source	<pre> &lt;xs:complexType name="commandType"&gt; &lt;xs:choice&gt; &lt;xs:element name="post" type="postType"/&gt; &lt;xs:element name="checkIn"&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="RoomNumber"/&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;xs:element name="checkOut"&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="RoomNumber"/&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;xs:element name="cancel"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Cancel the reservation&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:complexType&gt; &lt;xs:all&gt; &lt;xs:element name="GuestNum" type="xs:integer"/&gt; &lt;xs:element name="Reason" type="xs:string" minOccurs="0"/&gt; &lt;xs:element name="DeleteTraces" type="xs:int" minOccurs="0"/&gt; &lt;xs:element name="SendEmail" minOccurs="0"/&gt; &lt;xs:complexType&gt; &lt;xs:simpleContent&gt; &lt;xs:extension base="xs:int"&gt; &lt;xs:attribute name="TO" type="xs:string" use="required"/&gt; &lt;xs:attribute name="CC" type="xs:string" use="optional"/&gt; &lt;xs:attribute name="BCC" type="xs:string" use="optional"/&gt; &lt;/xs:extension&gt; &lt;/xs:simpleContent&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:all&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;xs:element name="resetProfilePassword"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Resets the user password. New password sent by email. Used from myFidelio Homepage&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="userName" type="xs:string"/&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:choice&gt;&lt;/xs:complexType&gt; </pre>
source	<pre> &lt;xs:complexType name="commandType"&gt; &lt;xs:choice&gt; &lt;xs:element name="post" type="postType"/&gt; &lt;xs:element name="checkIn"&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="RoomNumber"/&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;xs:element name="checkOut"&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="RoomNumber"/&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;xs:element name="cancel"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Cancel the reservation&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:complexType&gt; &lt;xs:all&gt; &lt;xs:element name="GuestNum" type="xs:integer"/&gt; &lt;xs:element name="Reason" type="xs:string" minOccurs="0"/&gt; &lt;xs:element name="DeleteTraces" type="xs:int" minOccurs="0"/&gt; &lt;xs:element name="SendEmail" minOccurs="0"/&gt; &lt;xs:complexType&gt; &lt;xs:simpleContent&gt; &lt;xs:extension base="xs:int"&gt; &lt;xs:attribute name="TO" type="xs:string" use="required"/&gt; &lt;xs:attribute name="CC" type="xs:string" use="optional"/&gt; &lt;xs:attribute name="BCC" type="xs:string" use="optional"/&gt; &lt;/xs:extension&gt; &lt;/xs:simpleContent&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:all&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;xs:element name="resetProfilePassword"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Resets the user password. New password sent by email. Used from myFidelio Homepage&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="userName" type="xs:string"/&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:choice&gt;&lt;/xs:complexType&gt; </pre>

**element** commandType/post

diagram	<p>The diagram shows a 'post' element box connected to a choice box (a circle with a vertical line and a plus sign). This choice box branches into two boxes: 'ccm_book_actuals' with a plus sign and the text 'Used to post CCM transaction', and 'crm_posting' with a plus sign and the text 'Used to post any transaction based on reservation'. The entire choice structure is highlighted in a yellow dashed box labeled 'postType'.</p>
namespace	V8XMLInterface
type	<b>postType</b>
children	<b>ccm_book_actuals</b> <b>crm_posting</b>
source	<code>&lt;xs:element name="post" type="postType"/&gt;</code>

**element** commandType/checkIn

diagram	<p>The diagram shows a 'checkIn' element box connected to a sequence box (a circle with three dots), which is then connected to a 'RoomNumber' element box.</p>
namespace	V8XMLInterface
children	<b>RoomNumber</b>
source	<code>&lt;xs:element name="checkIn"&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="RoomNumber"/&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt;</code>

**element** commandType/checkIn/RoomNumber

diagram	<p>The diagram shows a single 'RoomNumber' element box.</p>
namespace	V8XMLInterface
source	<code>&lt;xs:element name="RoomNumber"/&gt;</code>

**element** commandType/checkOut

diagram	<p>The diagram shows a 'checkOut' element box connected to a sequence box (a circle with three dots), which is then connected to a 'RoomNumber' element box.</p>
namespace	V8XMLInterface
children	<b>RoomNumber</b>
source	<code>&lt;xs:element name="checkOut"&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="RoomNumber"/&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt;</code>

**element** commandType/checkOut/RoomNumber

diagram	<p>The diagram shows a single 'RoomNumber' element box.</p>
namespace	V8XMLInterface

source	<code>&lt;xs:element name="RoomNumber"/&gt;</code>
--------	--

### element commandType/cancel

diagram	
namespace	V8XMLInterface
children	<b><u>GuestNum</u></b> <b><u>Reason</u></b> <b><u>DeleteTraces</u></b> <b><u>SendEmail</u></b>
annotation	documentation Cancel the reservation
source	<pre> &lt;xs:element name="cancel"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Cancel the reservation&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:complexType&gt; &lt;xs:all&gt; &lt;xs:element name="GuestNum" type="xs:integer"/&gt; &lt;xs:element name="Reason" type="xs:string" minOccurs="0"/&gt; &lt;xs:element name="DeleteTraces" type="xs:int" minOccurs="0"/&gt; &lt;xs:element name="SendEmail" minOccurs="0"&gt; &lt;xs:complexType&gt; &lt;xs:simpleContent&gt; &lt;xs:extension base="xs:int"&gt; &lt;xs:attribute name="TO" type="xs:string" use="required"/&gt; &lt;xs:attribute name="CC" type="xs:string" use="optional"/&gt; &lt;xs:attribute name="BCC" type="xs:string" use="optional"/&gt; &lt;/xs:extension&gt; &lt;/xs:simpleContent&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:all&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt; </pre>

### element commandType/cancel/GuestNum

diagram	
namespace	V8XMLInterface
type	<b>xs:integer</b>
source	<code>&lt;xs:element name="GuestNum" type="xs:integer"/&gt;</code>

### element commandType/cancel/Reason

diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>
source	<code>&lt;xs:element name="Reason" type="xs:string" minOccurs="0"/&gt;</code>

### element commandType/cancel/DeleteTraces

diagram	
namespace	V8XMLInterface
type	<b>xs:int</b>
source	<code>&lt;xs:element name="DeleteTraces" type="xs:int" minOccurs="0"/&gt;</code>

### element commandType/cancel/SendEmail

diagram						
namespace	V8XMLInterface					
type	extension of <b>xs:int</b>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	TO	xs:string	required			
	CC	xs:string	optional			
	BCC	xs:string	optional			
source	<pre>&lt;xs:element name="SendEmail" minOccurs="0"&gt; &lt;xs:complexType&gt; &lt;xs:simpleContent&gt; &lt;xs:extension base="xs:int"&gt; &lt;xs:attribute name="TO" type="xs:string" use="required"/&gt; &lt;xs:attribute name="CC" type="xs:string" use="optional"/&gt; &lt;xs:attribute name="BCC" type="xs:string" use="optional"/&gt; &lt;/xs:extension&gt; &lt;/xs:simpleContent&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt;</pre>					

### element commandType/resetProfilePassword

diagram						
namespace	V8XMLInterface					
children	<u><b>userName</b></u>					
annotation	documentation Resets the user password. New password sent by email. Used from myFidelio Homepage					
source	<pre>&lt;xs:element name="resetProfilePassword"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Resets the user password. New password sent by email. Used from myFidelio Homepage&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="userName" type="xs:string"/&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt;</pre>					

### element commandType/resetProfilePassword/userName

diagram						
namespace	V8XMLInterface					
type	<b>xs:string</b>					
source	<pre>&lt;xs:element name="userName" type="xs:string"/&gt;</pre>					

### complexType conditionsType

diagram						
namespace	V8XMLInterface					
children	<u><b>condition link</b></u>					

used by	elements <b><u>linkType/condition</u></b> <b><u>queryType/conditions</u></b> <b><u>updateType/conditions</u></b> <b><u>importType/delete/conditions</u></b>
annotation	documentation Defines list of conditions for search
source	<code>&lt;xs:complexType name="conditionsType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Defines list of conditions for search&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:sequence&gt; &lt;xs:element name="condition" type="conditionType"/&gt; &lt;xs:element name="link" type="linkType" minOccurs="0"/&gt; &lt;/xs:sequence&gt;&lt;/xs:complexType&gt;</code>

### element conditionsType/condition

diagram																									
namespace	V8XMLInterface																								
type	<b><u>conditionType</u></b>																								
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>name</td> <td>xs:anySimpleType</td> <td>required</td> <td></td> <td></td> <td></td> </tr> <tr> <td>operation</td> <td>binaryOperation</td> <td>required</td> <td></td> <td></td> <td></td> </tr> <tr> <td>dataType</td> <td>xs:NMTOKEN</td> <td>optional</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	name	xs:anySimpleType	required				operation	binaryOperation	required				dataType	xs:NMTOKEN	optional			
Name	Type	Use	Default	Fixed	Annotation																				
name	xs:anySimpleType	required																							
operation	binaryOperation	required																							
dataType	xs:NMTOKEN	optional																							
source	<code>&lt;xs:element name="condition" type="conditionType"/&gt;</code>																								

### element conditionsType/link

diagram													
namespace	V8XMLInterface												
type	<b><u>linkType</u></b>												
children	<b><u>condition</u></b>												
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>operator</td> <td>xs:NMTOKENS</td> <td>required</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	operator	xs:NMTOKENS	required			
Name	Type	Use	Default	Fixed	Annotation								
operator	xs:NMTOKENS	required											
source	<code>&lt;xs:element name="link" type="linkType" minOccurs="0"/&gt;</code>												

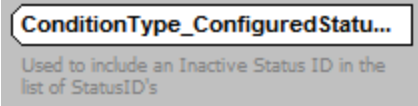
### complexType conditionType

diagram	
namespace	V8XMLInterface
type	extension of <b><u>conditionSimpleType</u></b>
used by	element <b><u>conditionsType/condition</u></b>




complexTypes	<u><a href="#">BlockDetailConditionType</a></u> <u><a href="#">BlockHeaderConditionType</a></u> <u><a href="#">ChildrenCategoriesConditionType</a></u> <u><a href="#">ConditionType</a></u> <u><a href="#">ConfiguredStatusList</a></u> <u><a href="#">ConditionType</a></u> <u><a href="#">InfoBoard</a></u> <u><a href="#">CreditCardTypesConditionType</a></u> <u><a href="#">CurrencyListConditionType</a></u> <u><a href="#">DepartmentCodeConditionType</a></u> <u><a href="#">PostngConditionType</a></u> <u><a href="#">ProfileConditionType</a></u> <u><a href="#">ReservationConditionType</a></u> <u><a href="#">SystemDateConditionType</a></u> <u><a href="#">SystemInformationConditionType</a></u>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:anySimpleType	required			
	operation	binaryOperation	required			
	dataType	xs:NMTOKEN	optional			
annotation	documentation	defines a condition set on a single field. This is abstract type. The real type should be selected according to the request type (ReservationConditionType, ProfileConditionType etc.)				
source	<pre> &lt;xs:complexType name="conditionType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;defines a condition set on a single field. This is abstract type. The real type should be selected according to the request type (ReservationConditionType, ProfileConditionType etc.)&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:simpleContent&gt; &lt;xs:extension base="conditionSimpleType"&gt; &lt;xs:attribute name="name" type="xs:anySimpleType" use="required"/&gt; &lt;xs:attribute name="operation" type="binaryOperation" use="required"/&gt; &lt;xs:attribute name="dataType" use="optional"&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:NMTOKEN"&gt; &lt;xs:enumeration value="String"/&gt; &lt;xs:enumeration value="Date"/&gt; &lt;xs:enumeration value="Int"/&gt; &lt;xs:enumeration value="Float"/&gt; &lt;xs:enumeration value="Plain"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; &lt;/xs:extension&gt; &lt;/xs:simpleContent&gt;&lt;/xs:complexType&gt; </pre>					

### complexType ConditionType\_ConfiguredStatusList

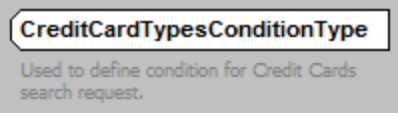
diagram						
namespace	V8XMLInterface					
type	restriction of <u><a href="#">conditionType</a></u>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:NMTOKEN	optional			
	operation	binaryOperation	required			
	dataType	xs:NMTOKEN	optional			
annotation	documentation	Used to include an Inactive Status ID in the list of StatusID's				
source	<pre> &lt;xs:complexType name="ConditionType_ConfiguredStatusList"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Used to include an Inactive Status ID in the list of StatusID's&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:simpleContent&gt; &lt;xs:restriction base="conditionType"&gt; &lt;xs:attribute name="name" use="optional"&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:NMTOKEN"&gt; &lt;xs:enumeration value="StatusInactiveID"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleContent&gt;&lt;/xs:complexType&gt; </pre>					

### complexType ConditionType\_InfoBoard

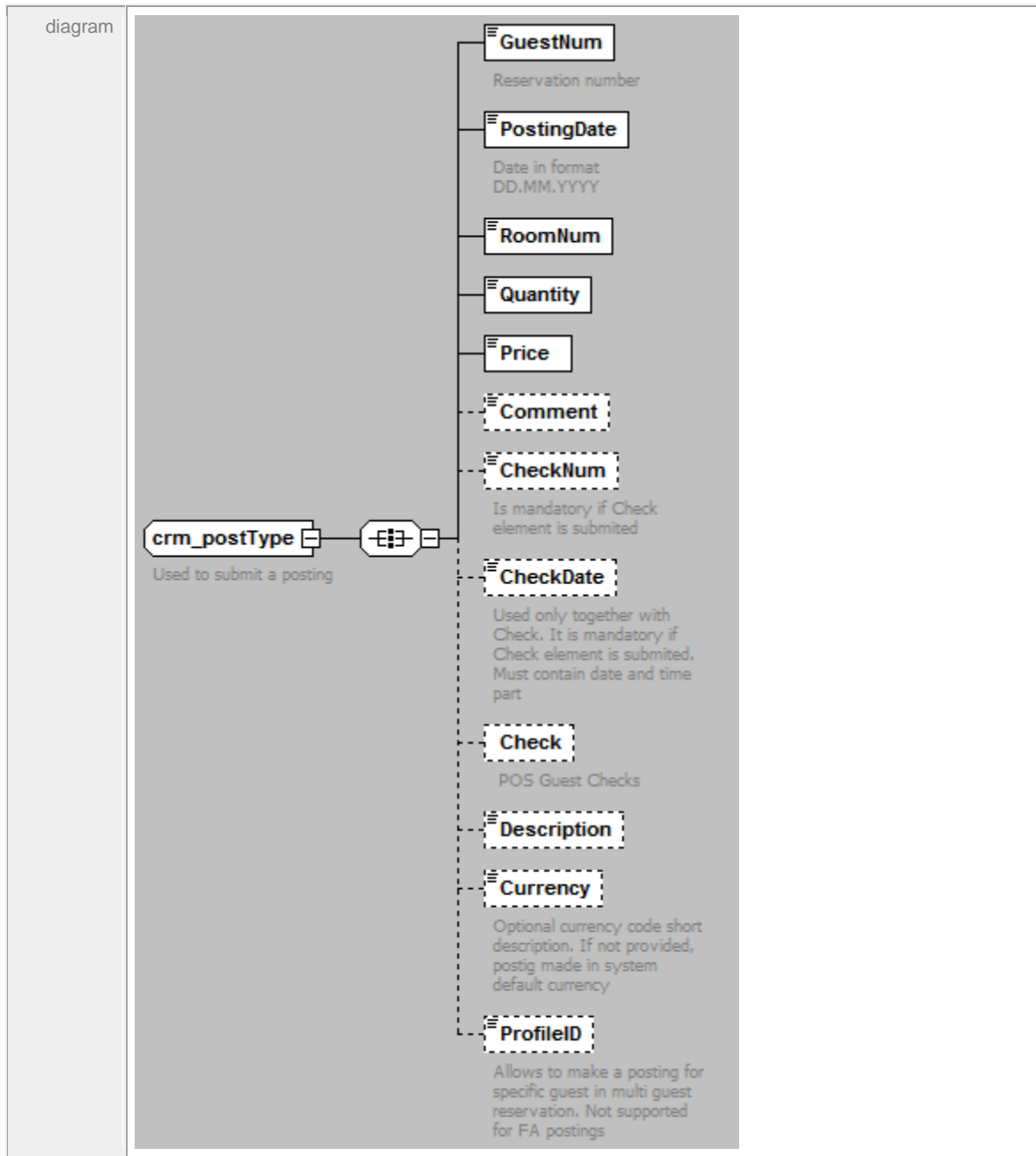
diagram						
namespace	V8XMLInterface					
type	extension of <u><a href="#">conditionType</a></u>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:anySimpleType	required			

	operation	binaryOperation	required
	dataType	xs:NMTOKEN	optional
source	<pre>&lt;xs:complexType name="ConditionType_InfoBoard"&gt; &lt;xs:simpleContent&gt; &lt;xs:extension base="conditionType"/&gt; &lt;/xs:simpleContent&gt;&lt;/xs:complexType&gt;</pre>		

### complexType CreditCardTypesConditionType

diagram						
namespace	V8XMLInterface					
type	restriction of <b>conditionType</b>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:NMTOKEN	required			
	operation	binaryOperation	required			
	dataType	xs:NMTOKEN	optional			
annotation	documentation Used to define condition for Credit Cards search request.					
source	<pre>&lt;xs:complexType name="CreditCardTypesConditionType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Used to define condition for Credit Cards search request.&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:simpleContent&gt; &lt;xs:restriction base="conditionType"&gt; &lt;xs:attribute name="name" use="required"&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:NMTOKEN"&gt; &lt;xs:enumeration value="CreditCardCode"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleContent&gt;&lt;/xs:complexType&gt;</pre>					


## complexType crm\_postType



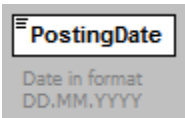
namespace	V8XMLInterface
children	<b><u>GuestNum</u></b> <b><u>PostingDate</u></b> <b><u>RoomNum</u></b> <b><u>Quantity</u></b> <b><u>Price</u></b> <b><u>Comment</u></b> <b><u>CheckNum</u></b> <b><u>CheckDate</u></b> <b><u>Check</u></b> <b><u>Description</u></b> <b><u>Currency</u></b> <b><u>ProfileID</u></b>
used by	element <b><u>postType/crm_posting</u></b>
annotation	documentation Used to submit a posting
source	<pre> &lt;xs:complexType name="crm_postType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Used to submit a posting&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:all&gt;     &lt;xs:element name="GuestNum" type="xs:integer"/&gt;     &lt;xs:element name="PostingDate" type="xs:string"/&gt;     &lt;xs:element name="RoomNum" type="xs:string"/&gt;     &lt;xs:element name="Quantity" type="xs:integer"/&gt;     &lt;xs:element name="Price" type="xs:string"/&gt;     &lt;xs:element name="Comment" type="xs:string" minOccurs="0"/&gt;     &lt;xs:element name="CheckNum" type="xs:string" minOccurs="0"/&gt;     &lt;xs:element name="CheckDate" type="xs:string" minOccurs="0"/&gt;     &lt;xs:element name="Check" type="xs:string" minOccurs="0"/&gt;     &lt;xs:element name="Description" type="xs:string" minOccurs="0"/&gt;     &lt;xs:element name="Currency" type="xs:string" minOccurs="0"/&gt;     &lt;xs:element name="ProfileID" type="xs:string" minOccurs="0"/&gt;   &lt;/xs:all&gt; &lt;/xs:complexType&gt; </pre>

	<pre> type="xs:string" minOccurs="0"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Used only together with Check. It is mandatory if Check element is submitted. Must contain date and time part&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:element&gt; &lt;xs:element name="Check" minOccurs="0"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt; POS Guest Checks&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:element&gt; &lt;xs:element name="Description" type="xs:string" minOccurs="0"/&gt; &lt;xs:element name="Currency" type="xs:string" minOccurs="0"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Optional currency code short description. If not provided, postig made in system default currency&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:element&gt; &lt;xs:element name="ProfileID" type="xs:integer" minOccurs="0"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Allows to make a posting for specific guest in multi guest reservation. Not supported for FA postings&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:element&gt; &lt;/xs:all&gt;&lt;/xs:complexType&gt; </pre>
--	--


#### element crm\_postType/GuestNum

diagram	
namespace	V8XMLInterface
type	<b>xs:integer</b>
annotation	documentation Reservation number
source	<pre>&lt;xs:element name="GuestNum" type="xs:integer"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Reservation number&lt;/xs:documentation&gt; &lt;/xs:annotation&gt;&lt;/xs:element&gt;</pre>

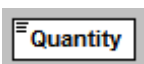
#### element crm\_postType/PostingDate

diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>
annotation	documentation Date in format DD.MM.YYYY
source	<pre>&lt;xs:element name="PostingDate" type="xs:string"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Date in format DD.MM.YYYY&lt;/xs:documentation&gt; &lt;/xs:annotation&gt;&lt;/xs:element&gt;</pre>

#### element crm\_postType/RoomNum


diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>
source	<pre>&lt;xs:element name="RoomNum" type="xs:string"/&gt;</pre>

#### element crm\_postType/Quantity


diagram	
namespace	V8XMLInterface
type	<b>xs:integer</b>

source	<code>&lt;xs:element name="Quantity" type="xs:integer"/&gt;</code>
--------	--

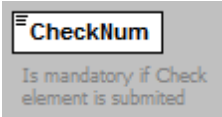
### element crm\_postType/Price

diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>
source	<code>&lt;xs:element name="Price" type="xs:string"/&gt;</code>

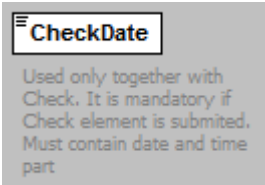
### element crm\_postType/Comment

diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>
source	<code>&lt;xs:element name="Comment" type="xs:string" minOccurs="0"/&gt;</code>

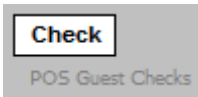
### element crm\_postType/CheckNum

diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>
annotation	documentation Is mandatory if Check element is submitted
source	<code>&lt;xs:element name="CheckNum" type="xs:string" minOccurs="0"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Is mandatory if Check element is submitted&lt;/xs:documentation&gt; &lt;/xs:annotation&gt;&lt;/xs:element&gt;</code>


### element crm\_postType/CheckDate

diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>
annotation	documentation Used only together with Check. It is mandatory if Check element is submitted. Must contain date and time part
source	<code>&lt;xs:element name="CheckDate" type="xs:string" minOccurs="0"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Used only together with Check. It is mandatory if Check element is submitted. Must contain date and time part&lt;/xs:documentation&gt; &lt;/xs:annotation&gt;&lt;/xs:element&gt;</code>

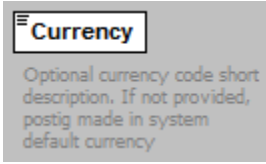
**element** crm\_postType/Check

diagram	
namespace	V8XMLInterface
annotation	documentation POS Guest Checks
source	<code>&lt;xs:element name="Check" minOccurs="0"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt; POS Guest Checks&lt;/xs:documentation&gt; &lt;/xs:annotation&gt;&lt;/xs:element&gt;</code>

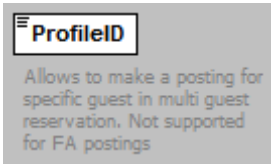
**element** crm\_postType/Description

diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>
source	<code>&lt;xs:element name="Description" type="xs:string" minOccurs="0"/&gt;</code>

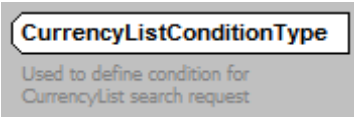
**element** crm\_postType/Currency

diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>
annotation	documentation Optional currency code short description. If not provided, postig made in system default currency
source	<code>&lt;xs:element name="Currency" type="xs:string" minOccurs="0"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Optional currency code short description. If not provided, postig made in system default currency&lt;/xs:documentation&gt; &lt;/xs:annotation&gt;&lt;/xs:element&gt;</code>

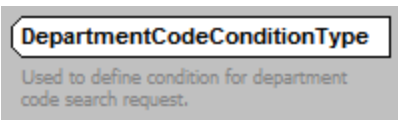
**element** crm\_postType/ProfileID

diagram	
namespace	V8XMLInterface
type	<b>xs:integer</b>
annotation	documentation Allows to make a posting for specific guest in multi guest reservation. Not supported for FA postings
source	<code>&lt;xs:element name="ProfileID" type="xs:integer" minOccurs="0"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Allows to make a posting for specific guest in multi guest reservation. Not supported for FA postings&lt;/xs:documentation&gt; &lt;/xs:annotation&gt;&lt;/xs:element&gt;</code>

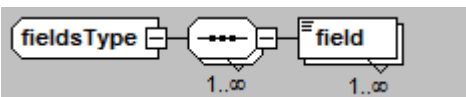
### complexType CurrencyListConditionType

diagram						
namespace	V8XMLInterface					
type	restriction of <u>conditionType</u>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:NMTOKEN	required			
	operation	binaryOperation	required			
	dataType	xs:NMTOKEN	optional			
annotation	documentation Used to define condition for CurrencyList search request					
source	<pre>&lt;xs:complexType name="CurrencyListConditionType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Used to define condition for CurrencyList search request&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:simpleContent&gt; &lt;xs:restriction base="conditionType"&gt; &lt;xs:attribute name="name" use="required"&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:NMTOKEN"&gt; &lt;xs:enumeration value="CurrencyShortdesc"/&gt; &lt;xs:enumeration value="CurrencyISO3"/&gt; &lt;xs:enumeration value="CurrencyDepartment"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleContent&gt;&lt;/xs:complexType&gt;</pre>					

### complexType DepartmentCodeConditionType


diagram						
namespace	V8XMLInterface					
type	restriction of <u>conditionType</u>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:NMTOKEN	required			
	operation	binaryOperation	required			
	dataType	xs:NMTOKEN	optional			
annotation	documentation Used to define condition for department code search request.					
source	<pre>&lt;xs:complexType name="DepartmentCodeConditionType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Used to define condition for department code search request.&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:simpleContent&gt; &lt;xs:restriction base="conditionType"&gt; &lt;xs:attribute name="name" use="required"&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:NMTOKEN"&gt; &lt;xs:enumeration value="DepartmentCode"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;DepartmentCode means 3 letter short description&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:enumeration&gt; &lt;xs:enumeration value="DepartmentName"/&gt; &lt;xs:enumeration value="DepartmentShortDesc"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleContent&gt;&lt;/xs:complexType&gt;</pre>					

### complexType fieldsType


diagram						
namespace	V8XMLInterface					
children	<u>field</u>					

used by	elements <u><a href="#">insertType/fields</a></u> <u><a href="#">queryType/fields</a></u> <u><a href="#">rowType/fields</a></u> <u><a href="#">AvailabilityForWebType/fields</a></u> <u><a href="#">updateType/fields</a></u> <u><a href="#">HotelAvailabilityType/fields</a></u>
source	<code>&lt;xs:complexType name="fieldsType"&gt; &lt;xs:sequence maxOccurs="unbounded"&gt; &lt;xs:element name="field" type="fieldType" maxOccurs="unbounded"/&gt; &lt;/xs:sequence&gt;&lt;/xs:complexType&gt;</code>

### element fieldsType/field

diagram						
namespace	V8XMLInterface					
type	<u><a href="#">fieldType</a></u>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:NMTOKEN	required			
	attr	xs:string	optional			documentation Could be used as additional identifier for the field. For example Chaild category of type of reservation attribute
	addData	xs:string	optional			
source	<code>&lt;xs:element name="field" type="fieldType" maxOccurs="unbounded"/&gt;</code>					

### complexType fieldType

diagram						
namespace	V8XMLInterface					
type	extension of <b>xs:string</b>					
used by	element <u><a href="#">fieldsType/field</a></u>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:NMTOKEN	required			
	attr	xs:string	optional			documentation Could be used as additional identifier for the field. For example Chaild category of type of reservation attribute
	addData	xs:string	optional			
annotation	documentation Fields which can be requested					



source	<pre> &lt;xs:complexType name="fieldType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Fields which can be requested&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:simpleContent&gt; &lt;xs:extension base="xs:string"&gt; &lt;xs:attribute name="name" use="required"&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:NMTOKEN"&gt; &lt;xs:enumeration value="GuestArrival"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Arrival date&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:enumeration&gt; &lt;xs:enumeration value="ReservationComment1"/&gt; &lt;xs:enumeration value="ReservationComment2"/&gt; &lt;xs:enumeration value="CheckinDateTime"/&gt; &lt;xs:enumeration value="NoAvailReason"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;1 = Waitlist 2 = Offer 3 = Canceled 4 = No Show&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:enumeration&gt; &lt;xs:enumeration value="ProfileID"/&gt; &lt;xs:enumeration value="PaymentMethod"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;The department code which is associated with the payment&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:enumeration&gt; &lt;xs:enumeration value="GuestDeparture"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Departure date&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:enumeration&gt; &lt;xs:enumeration value="GuestFirstname"/&gt; &lt;xs:enumeration value="GuestName"/&gt; &lt;xs:enumeration value="Name2"/&gt; &lt;xs:enumeration value="GuestNum"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Reservation Number&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:enumeration&gt; &lt;xs:enumeration value="ReservationStatus"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;0=tentative 1=definitive 3=not affecting availability&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:enumeration&gt; &lt;xs:enumeration value="RoomNum"/&gt; &lt;xs:enumeration value="RoomType"/&gt; &lt;xs:enumeration value="ProfileType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;1 - Company 2 - Individual 3 – Employee&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:enumeration&gt; &lt;xs:enumeration value="ProfileCategory"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;1=Company 2=Travel Agent 3=Source 4=Individual 5=Vendors 6=Meeting Planner&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:enumeration&gt; &lt;xs:enumeration value="DepartmentCode"/&gt; &lt;xs:enumeration value="DepartmentName"/&gt; &lt;xs:enumeration value="DepartmentShortDesc"/&gt; &lt;xs:enumeration value="DepartmentType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;1 = debit 2 = tax 3 = credit &lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:enumeration&gt; &lt;xs:enumeration value="DepartmentArrangement"/&gt; &lt;xs:enumeration value="AddressGreeting"/&gt; &lt;xs:enumeration value="Birthday"/&gt; &lt;xs:enumeration value="City"/&gt; &lt;xs:enumeration value="CountryISO2"/&gt; &lt;xs:enumeration value="CountryISO3"/&gt; &lt;xs:enumeration value="Email"/&gt; &lt;xs:enumeration value="Fax"/&gt; &lt;xs:enumeration value="Gender"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;M – male, F – Female&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:enumeration&gt; &lt;xs:enumeration value="DeleteCode"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Prfile delete code. 0 –profile is active 1 – profile is not active&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:enumeration&gt; &lt;xs:enumeration value="GuestType"/&gt; &lt;xs:enumeration value="LangISO2"/&gt; &lt;xs:enumeration value="LangISO3"/&gt; &lt;xs:enumeration value="LetterGreeting"/&gt; &lt;xs:enumeration value="NationalityISO2"/&gt; &lt;xs:enumeration value="NationalityISO3"/&gt; &lt;xs:enumeration value="VipCode"/&gt; &lt;xs:enumeration value="ProfileID"/&gt; &lt;xs:enumeration value="Street1"/&gt; &lt;xs:enumeration value="Street2"/&gt; &lt;xs:enumeration value="Street3"/&gt; &lt;xs:enumeration value="State"/&gt; &lt;xs:enumeration value="Telephone"/&gt; &lt;xs:enumeration value="Title"/&gt; &lt;xs:enumeration value="ZIP"/&gt; &lt;xs:enumeration value="SystemDate"/&gt; &lt;xs:enumeration value="LicenseName"/&gt; &lt;xs:enumeration value="LicenseNoOfRooms"/&gt; &lt;xs:enumeration value="CrsSystem"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;ID of the CRS system the reservation originates from&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:enumeration&gt; &lt;xs:enumeration value="CrsResNumber"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Unique ID coming from CRS System, linked to CrsSystem in order to support multiple CRS systems.&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:enumeration&gt; &lt;xs:enumeration value="Notes"/&gt; &lt;xs:enumeration value="NoOfRooms"/&gt; &lt;xs:enumeration value="NoOfAdults"/&gt; &lt;xs:enumeration value="RateCode"/&gt; &lt;xs:enumeration value="RateValue"/&gt; &lt;xs:enumeration value="ForeignRateValue"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Rate in foreign currency&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:enumeration&gt; &lt;xs:enumeration value="CurrencyCode"/&gt; &lt;xs:enumeration value="MarketCode"/&gt; &lt;xs:enumeration value="CompanyID"/&gt; &lt;xs:enumeration value="TravelAgentID"/&gt; &lt;xs:enumeration value="SourceID"/&gt; &lt;xs:enumeration value="CompanyName"/&gt; &lt;xs:enumeration value="TravelAgentName"/&gt; &lt;xs:enumeration value="SourceName"/&gt; &lt;xs:enumeration value="ReservationAttribute"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Could be use with conjunction with optional attribute "attr" which contains the required attribute type.For Example &lt;field name="ReservationAttribute" attr="ABC"/&gt; If the attribute is not specified than all available attributes will be included in the responsel self this element used in the update or insert than user must specify attribute attr&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:enumeration&gt; &lt;xs:enumeration value="Child"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Could be used with conjunction with optional attribute "attr" which contains the required child type.For Example &lt;field name="Child" attr="JUN"/&gt; for juniorIf the attribute is not specified than all available attributes will be included in the responseThe attr mandatory for update / insert statements&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:enumeration&gt; &lt;xs:enumeration value="SourceCode"/&gt; &lt;xs:enumeration value="ChannelCode"/&gt; &lt;xs:enumeration value="PostingPrice"/&gt; &lt;xs:enumeration value="PostingID"/&gt; &lt;xs:enumeration value="PostingQuantity"/&gt; &lt;xs:enumeration value="PostingForeignPrice"/&gt; &lt;xs:enumeration value="PostingGuestNumOrig"/&gt; &lt;xs:enumeration value="PostingGuestNumOrig"/&gt; &lt;xs:enumeration value="PostingProfileOrigID"/&gt; &lt;xs:enumeration value="PostingComment"/&gt; &lt;xs:enumeration value="PostingCheckNr"/&gt; &lt;xs:enumeration value="PostingDescription"/&gt; &lt;xs:enumeration value="PostingDate"/&gt; </pre>
--------	--

```

<xs:enumeration value="LoginName"/> <xs:enumeration value="LoginPassword"/>
<xs:enumeration value="ChildrenCategoriesCode"/> <xs:enumeration
value="ChildrenCategoriesName"/> <xs:enumeration value="ChildrenCategoriesFromAge"/>
<xs:enumeration value="ChildrenCategoriesToAge"/> <xs:enumeration value="CreditCardCode"/>
<xs:enumeration value="CreditCardName"/> <xs:enumeration value="FANumber"/>
<xs:enumeration value="FADescription"/> <xs:enumeration value="ExpArrival"/>
<xs:enumeration value="ExpDeparture"/> <xs:enumeration value="ActArrival"/>
<xs:enumeration value="ActDeparture"/> <xs:enumeration value="CreditLimitType"/>
<xs:enumeration value="CreditLimitAmount"/> <xs:enumeration value="CreditCardNo"/>
<xs:enumeration value="ExpiryMonth"/> <xs:enumeration value="ExpiryYear"/>
<xs:enumeration value="CardholderName"/> <xs:enumeration value="CreditCardCode"/>
<xs:enumeration value="FAGroup"/> <xs:enumeration value="TryToMerge"/>
<xs:annotation> <xs:documentation>Can be used with Profile -> Insert command only.Values 1 -
true, 0 - falsef Value is 1 XML interface will try to find existing profiles which has similar data (according to
profile merge setup)</xs:documentation> </xs:annotation> </xs:enumeration>
<xs:enumeration value="HtmlId"/> <xs:enumeration value="HtmlName"/> <xs:enumeration
value="HtmlContent"/> <xs:enumeration value="ProfileGlobalID"/> <xs:enumeration
value="CentralSyncTime"/> <xs:enumeration value="SeatingID"/> <xs:enumeration
value="SeatingShortDesc"/> <xs:enumeration value="SeatingLongDesc"/> <xs:enumeration
value="SeatingDisplayOrder"/> <xs:enumeration value="StatusID"/> <xs:enumeration
value="StatusShortDesc"/> <xs:enumeration value="StatusLongDesc"/> <xs:enumeration
value="StatusDisplayOrder"/> <xs:enumeration value="StatusWebName"/> <xs:enumeration
value="StatusInactiveID"/> <xs:enumeration value="Booking_CompanyID"/>
<xs:enumeration value="Booking_ContactID"/> <xs:enumeration value="Booking_ArrivalDate"/>
<xs:enumeration value="Booking_DepartureDate"/> <xs:enumeration value="Booking_Name"/>
<xs:enumeration value="Booking_ConferenceStatus"/> <xs:enumeration
value="Booking_RoomStatus"/> <xs:enumeration value="Booking_ConferenceOnly"/>
<xs:enumeration value="Booking_RoomsOnly"/> <xs:enumeration
value="Booking_ConferenceAndRooms"/> <xs:enumeration value="Booking_Notes"/>
<xs:enumeration value="Booking_NoOfAttendes"/> <xs:enumeration
value="Booking_MarketCode"/> <xs:enumeration value="Booking_SourceCode"/>
<xs:enumeration value="Booking_NoteCategory"/> <xs:enumeration
value="ConferencePackageID"/> <xs:enumeration value="ConferencePackageName"/>
<xs:enumeration value="ConferencePackageValidFrom"/> <xs:enumeration
value="ConferencePackageValidTo"/> <xs:enumeration value="EventID"/> <xs:enumeration
value="EventName"/> <xs:enumeration value="EventStartDate"/> <xs:enumeration
value="EventStartTime"/> <xs:enumeration value="EventEndDate"/> <xs:enumeration
value="EventEndTime"/> <xs:enumeration value="EventInfoBoard"/> <xs:enumeration
value="EventProfileName"/> <xs:enumeration value="EventPax"/> <xs:enumeration
value="EventRoomName"/> <xs:enumeration value="TimeStamp"/> <xs:enumeration
value="EventStatus"/> <xs:enumeration value="EventRoomId"/> <xs:enumeration
value="ActivityId"/> <xs:enumeration value="ActivityCompanyName"/> <xs:enumeration
value="ActivityContactName"/> <xs:enumeration value="ActivityDescription"/>
<xs:enumeration value="ActivityStartDate"/> <xs:enumeration value="ActivityEndDate"/>
<xs:enumeration value="ActivityManager"/> <xs:enumeration value="CreditStatus"/>
<xs:enumeration value="CreditReason"/> <xs:enumeration value="CreditStatusChangeTime"/>
<xs:enumeration value="LinkedProfile"/> <xs:enumeration value="AddProfileLink"/>
<xs:enumeration value="DeleteProfileLink"/> <xs:enumeration value="EditProfileLink"/>
<xs:enumeration value="ActivityCompanyID"/> <xs:enumeration
value="ActivityCompanyGlobalID"/> <xs:enumeration value="ActivityContactID"/>
<xs:enumeration value="ActivityContactGlobalID"/> <xs:enumeration value="ActivityManagerID"/>
<xs:enumeration value="ActivityTypeID"/> <xs:enumeration value="ActivityAllDayID"/>
<xs:enumeration value="ActivityShowsID"/> <xs:enumeration value="ActivityReminderID"/>
<xs:enumeration value="ActivityNeedReminder"/> <xs:enumeration value="ActivityPrivate"/>
<xs:enumeration value="ActivityStartDateTime"/> <xs:enumeration value="ActivityEndDateTime"/>
<xs:enumeration value="ActivityCompletedDate"/> <xs:enumeration
value="ActivityLinkedBookingId"/> <xs:enumeration value="ActivityLinkedCommunicationId"/>
<xs:enumeration value="ActivityCheckAvailability"/> <xs:enumeration value="ActivityNotes"/>
<xs:enumeration value="BirthPlace"/> <xs:enumeration value="Department"/>
<xs:enumeration value="Position"/> <xs:enumeration value="GroupID"/> <xs:enumeration
value="GroupName"/> <xs:enumeration value="GroupContactID"/> <xs:enumeration
value="GroupFrom"/> <xs:enumeration value="GroupTo"/> <xs:enumeration
value="GroupTemplate"/> <xs:enumeration value="GroupTemplateID"/> <xs:enumeration
value="GroupInactive"/> <xs:enumeration value="GroupCompanyName"/> <xs:enumeration
value="GroupCompanyID"/> <xs:enumeration value="GroupTravelAgentName"/>
<xs:enumeration value="GroupTravelAgentID"/> <xs:enumeration value="GroupSourceName"/>
<xs:enumeration value="GroupSourceID"/> <xs:enumeration value="GroupIndividualName"/>
<xs:enumeration value="GroupIndividualID"/> <xs:enumeration value="GroupVendorName"/>
<xs:enumeration value="GroupVendorID"/> <xs:enumeration
value="GroupMeetingPlannerName"/> <xs:enumeration value="GroupMeetingPlannerID"/>
<xs:enumeration value="GroupComment"/> <xs:enumeration value="BlockDetailID"/>
<xs:enumeration value="BlockDetailDate"/> <xs:enumeration value="BlockDetailRoomType"/>
<xs:enumeration value="BlockDetailCutOffDate"/> <xs:enumeration
value="BlockDetailInitNoOfRooms"/> <xs:enumeration value="BlockDetailActualNoOfRooms"/>

```

```

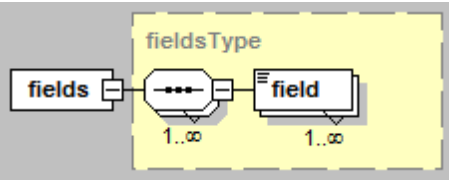
<xs:enumeration value="BlockDetailPickedUp"/> <xs:enumeration value="BlockDetailCancelled"/>
<xs:enumeration value="BlockHeaderID"/> <xs:enumeration value="BlockHeaderCode"/>
<xs:enumeration value="BlockHeaderName"/> <xs:enumeration value="BlockHeaderFrom"/>
<xs:enumeration value="BlockHeaderTo"/> <xs:enumeration value="BlockHeaderStatusCode"/>
<xs:enumeration value="BlockHeaderStatus"/> <xs:enumeration value="BlockHeaderReservationStatus"/>
<xs:enumeration value="BlockHeaderElastic"/> <xs:enumeration value="BlockHeaderRate"/>
<xs:enumeration value="TotalStay"/> <xs:enumeration value="Guarantee"/>
<xs:enumeration value="RoomTypeDesc"/> <xs:enumeration value="ForeignTotalStay"/>
<xs:enumeration value="Date"/> <xs:enumeration value="RoomTypeLongDesc"/>
<xs:enumeration value="BaseCurrencyPrice"/> <xs:enumeration value="Price"/> <xs:enumeration value="RatelInfo"/>
<xs:enumeration value="ApprovalAmount"/> <xs:enumeration value="ApprovalCode"/> <xs:enumeration value="CurrencyShortdesc"/>
<xs:enumeration value="CurrencyName"/> <xs:enumeration value="CurrencyISO3"/> <xs:enumeration value="CurrencyDigits"/>
<xs:enumeration value="CurrencyDepartment"/> <xs:enumeration value="HurdleLOS"/> <xs:enumeration value="HurdleRateAmount"/>
<xs:documentation> Amount which could be exceeded for given arrival date and LOS to pass the hurdle
</xs:documentation> </xs:annotation> </xs:enumeration> <xs:documentation> variation of the hurdle rate
value="HurdleDelta"/> <xs:documentation> each time a reservation is made. Not allowed if LOS is not 0</xs:documentation>
</xs:annotation> </xs:enumeration> <xs:enumeration value="HurdleClosed"/> <xs:documentation> 1 - closed for this arrival day and this length of stay</xs:documentation>
</xs:annotation> </xs:enumeration> <xs:enumeration value="HurdleMaxRateAmount"/>
<xs:documentation> maximal value of the hurdle rate</xs:documentation>
</xs:annotation> </xs:enumeration> <xs:enumeration value="OverbookingDate"/>
<xs:enumeration value="OverbookingType"/> <xs:documentation> 1 = House Overbooking 2 = Room Type Overbooking</xs:documentation>
</xs:annotation> <xs:enumeration value="OverbookingNoOfRooms"/> <xs:documentation> Number of Rooms for this overbooking</xs:documentation>
</xs:annotation> <xs:enumeration value="PackageName"/> <xs:enumeration value="PackageCode"/> <xs:enumeration value="PackageFrequencyName"/>
<xs:enumeration value="PackageFrequencyCode"/> <xs:enumeration value="PackageFrequencyName"/> <xs:enumeration value="PackageCurrency"/>
<xs:enumeration value="PackageStart"/> <xs:enumeration value="PackageEnd"/>
<xs:enumeration value="PackagePrice"/> <xs:enumeration value="PackageIncluded"/>
<xs:enumeration value="PackageSinglePerRes"/> <xs:enumeration value="PackageShowInRes"/>
<xs:enumeration value="PackagePriceType"/> <xs:documentation> 0 - per child category 1 - per adult 2 - flat</xs:documentation>
</xs:annotation> <xs:enumeration value="PackagePriceChildCategory"/> <xs:documentation> Presented only PackagePriceType is 0</xs:documentation>
</xs:annotation> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="attr" type="xs:string" use="optional"/>
<xs:documentation> Could be used as additional identifier for the field. For example Child category or type of reservation
attribute</xs:documentation> </xs:annotation> </xs:attribute> <xs:attribute name="addData" type="xs:string" use="optional"/>
</xs:extension> </xs:simpleContent></xs:complexType>

```

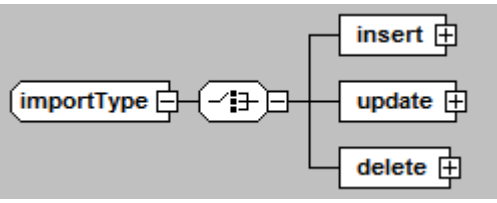
**complexType** HotelAvailabilityType

diagram						
namespace	V8XMLInterface					
children	<b>fields</b>					
used by	element <b>requestType/HotelAvailability</b>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	Name	xs:string	optional			
	TranslationTable	xs:string	optional			
source	<xs:complexType name="HotelAvailabilityType"> <xs:all> <xs:element name="fields" type="fieldsType"/> </xs:all> <xs:attribute name="Name" type="xs:string" use="optional"/> <xs:attribute name="TranslationTable" type="xs:string" use="optional"/></xs:complexType>					

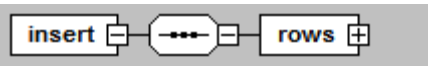
**element** HotelAvailabilityType/fields

diagram	
namespace	V8XMLInterface
type	<b>fieldsType</b>
children	<b>field</b>
source	<code>&lt;xs:element name="fields" type="fieldsType"/&gt;</code>

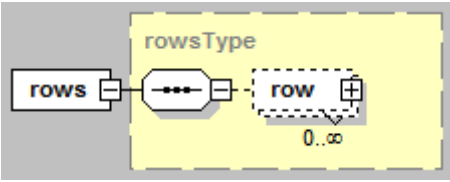
**complexType** importType

diagram																									
namespace	V8XMLInterface																								
children	<b>insert update delete</b>																								
used by	element <b>requestType/import</b>																								
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>TableName</td> <td>xs:string</td> <td>required</td> <td></td> <td></td> <td></td> </tr> <tr> <td>HotelID</td> <td>xs:string</td> <td>required</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Name</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	TableName	xs:string	required				HotelID	xs:string	required				Name	xs:string	optional			
Name	Type	Use	Default	Fixed	Annotation																				
TableName	xs:string	required																							
HotelID	xs:string	required																							
Name	xs:string	optional																							
source	<code>&lt;xs:complexType name="importType"&gt; &lt;xs:choice&gt; &lt;xs:element name="insert"&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="rows" type="rowsType"/&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;xs:element name="update" type="updateType"/&gt; &lt;xs:element name="delete"&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="conditions" type="conditionsType" minOccurs="0"/&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:choice&gt; &lt;xs:attribute name="TableName" type="xs:string" use="required"/&gt; &lt;xs:attribute name="HotelID" type="xs:string" use="required"/&gt; &lt;xs:attribute name="Name" type="xs:string" use="optional"/&gt;&lt;/xs:complexType&gt;</code>																								

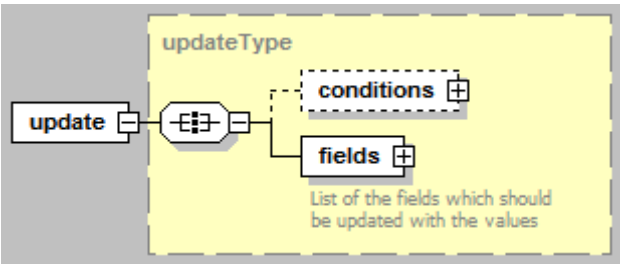
**element** importType/insert

diagram	
namespace	V8XMLInterface
children	<b>rows</b>
source	<code>&lt;xs:element name="insert"&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="rows" type="rowsType"/&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt;</code>

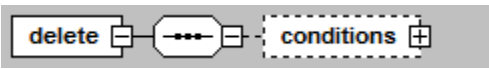
**element importType/insert/rows**

diagram	
namespace	V8XMLInterface
type	<u>rowsType</u>
children	<u>row</u>
source	<code>&lt;xs:element name="rows" type="rowsType"/&gt;</code>

**element importType/update**

diagram																									
namespace	V8XMLInterface																								
type	<u>updateType</u>																								
children	<u>conditions</u> <u>fields</u>																								
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>ReqType</td> <td>xs:NMTOKEN</td> <td>required</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Name</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> <td></td> </tr> <tr> <td>TranslationTable</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> <td>documentation If present in the request, short descriptions in request and response will be translated based on the specified translation table</td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	ReqType	xs:NMTOKEN	required				Name	xs:string	optional				TranslationTable	xs:string	optional			documentation If present in the request, short descriptions in request and response will be translated based on the specified translation table
Name	Type	Use	Default	Fixed	Annotation																				
ReqType	xs:NMTOKEN	required																							
Name	xs:string	optional																							
TranslationTable	xs:string	optional			documentation If present in the request, short descriptions in request and response will be translated based on the specified translation table																				
source	<code>&lt;xs:element name="update" type="updateType"/&gt;</code>																								

**element importType/delete**

diagram	
namespace	V8XMLInterface
children	<u>conditions</u>

source	<code>&lt;xs:element name="delete"&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="conditions" type="conditionsType" minOccurs="0"/&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt;</code>
--------	--

### element importType/delete/conditions

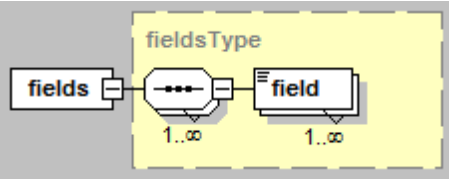
diagram	
namespace	V8XMLInterface
type	<b>conditionsType</b>
children	<b>condition link</b>
source	<code>&lt;xs:element name="conditions" type="conditionsType" minOccurs="0"/&gt;</code>

### complexType insertType

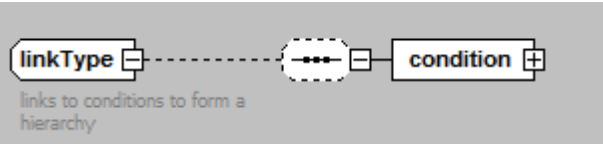
diagram																									
namespace	V8XMLInterface																								
children	<b>fields</b>																								
used by	element <b>requestType/insert</b>																								
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>ReqType</td> <td>xs:NMTOKEN</td> <td>required</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Name</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> <td></td> </tr> <tr> <td>TranslationTable</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> <td>documentation If present in the request, short descriptions in request and response will be translated based on the specified translation table</td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	ReqType	xs:NMTOKEN	required				Name	xs:string	optional				TranslationTable	xs:string	optional			documentation If present in the request, short descriptions in request and response will be translated based on the specified translation table
Name	Type	Use	Default	Fixed	Annotation																				
ReqType	xs:NMTOKEN	required																							
Name	xs:string	optional																							
TranslationTable	xs:string	optional			documentation If present in the request, short descriptions in request and response will be translated based on the specified translation table																				
annotation	documentation Not described yet																								
source	<code>&lt;xs:complexType name="insertType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Not described yet&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:sequence&gt; &lt;xs:element name="fields" type="fieldsType"/&gt; &lt;/xs:sequence&gt; &lt;xs:attribute name="ReqType" use="required"&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:NMTOKEN"&gt; &lt;xs:enumeration value="Profile"/&gt; &lt;xs:enumeration value="Reservation"/&gt; &lt;xs:enumeration value="FinancialAccount"/&gt; &lt;xs:enumeration value="ConferenceBooking"/&gt; &lt;xs:enumeration value="Activity"/&gt; &lt;xs:enumeration value="Task"/&gt; &lt;xs:enumeration value="Group"/&gt; &lt;xs:enumeration value="GroupTemplate"/&gt; &lt;xs:enumeration value="CreditCardApproval"/&gt; &lt;xs:enumeration value="Report"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; &lt;xs:attribute name="Name" type="xs:string" use="optional"/&gt; &lt;xs:attribute name="TranslationTable" type="xs:string" use="optional"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;If present in the request, short descriptions in request and response will be translated</code>																								

	based on the specified translation table</xs:documentation> </xs:annotation> </xs:attribute></xs:complexType>
--	--

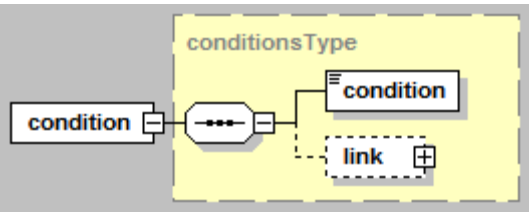
### element insertType/fields

diagram	
namespace	V8XMLInterface
type	<b>fieldsType</b>
children	<b>field</b>
source	<xs:element name="fields" type="fieldsType"/>

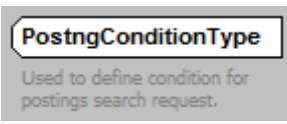
### complexType linkType

diagram													
namespace	V8XMLInterface												
children	<b>condition</b>												
used by	element <b>conditionsType/link</b>												
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>operator</td> <td>xs:NMTOKENS</td> <td>required</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	operator	xs:NMTOKENS	required			
Name	Type	Use	Default	Fixed	Annotation								
operator	xs:NMTOKENS	required											
annotation	documentation links to conditions to form a hierarchy												
source	<xs:complexType name="linkType"> </xs:annotation> </xs:documentation>links to conditions to form a hierarchy</xs:documentation> </xs:annotation> <xs:sequence minOccurs="0"> <xs:element name="condition" type="conditionsType"/> </xs:sequence> <xs:attribute name="operator" use="required"> <xs:simpleType> <xs:restriction base="xs:NMTOKENS"> <xs:enumeration value="AND"/> <xs:enumeration value="OR"/> <xs:enumeration value="NOT"/> </xs:restriction> </xs:simpleType> </xs:attribute></xs:complexType>												


### element linkType/condition

diagram	
namespace	V8XMLInterface
type	<b>conditionsType</b>
children	<b>condition link</b>
source	<xs:element name="condition" type="conditionsType"/>

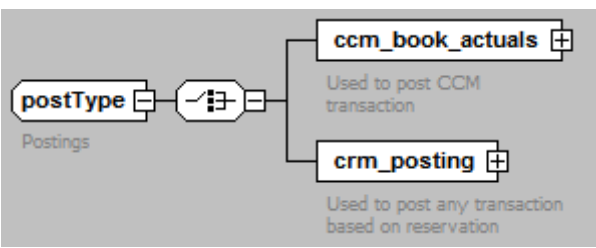
### complexType PostngConditionType

diagram						
namespace	V8XMLInterface					
type	restriction of <u>conditionType</u>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:NMTOKEN	required			
	operation	binaryOperation	required			
	dataType	xs:NMTOKEN	optional			
annotation	documentation Used to define condition for postngs search request.					
source	<pre>&lt;xs:complexType name="PostngConditionType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Used to define condition for postngs search request. &lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:simpleContent&gt; &lt;xs:restriction base="conditionType"&gt; &lt;xs:attribute name="name" use="required"&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:NMTOKEN"&gt; &lt;xs:enumeration value="PostngID"/&gt; &lt;xs:enumeration value="ProfileID"/&gt; &lt;xs:enumeration value="GuestNum"/&gt; &lt;xs:enumeration value="DepartmentCode"/&gt; &lt;xs:enumeration value="PostngDescription"/&gt; &lt;xs:enumeration value="PostngDate"/&gt; &lt;xs:enumeration value="PostngProfileOrigID"/&gt; &lt;xs:enumeration value="PostngGuestNumOrig"/&gt; &lt;xs:enumeration value="PostngComment"/&gt; &lt;xs:enumeration value="PostngCheckNr"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleContent&gt;&lt;/xs:complexType&gt;</pre>					

### complexType postResponseType

diagram						
namespace	V8XMLInterface					
used by	element <u>fidelio/response</u>					
annotation	documentation Returning by XML interface as a response for postng					
source	<pre>&lt;xs:complexType name="postResponseType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Returning by XML interface as a response for postng&lt;/xs:documentation&gt; &lt;/xs:annotation&gt;&lt;/xs:complexType&gt;</pre>					

### complexType postType

diagram						
namespace	V8XMLInterface					
children	<u>ccm_book_actuals</u> <u>crm_postng</u>					
used by	element <u>commandType/post</u>					
annotation	documentation Postngs					

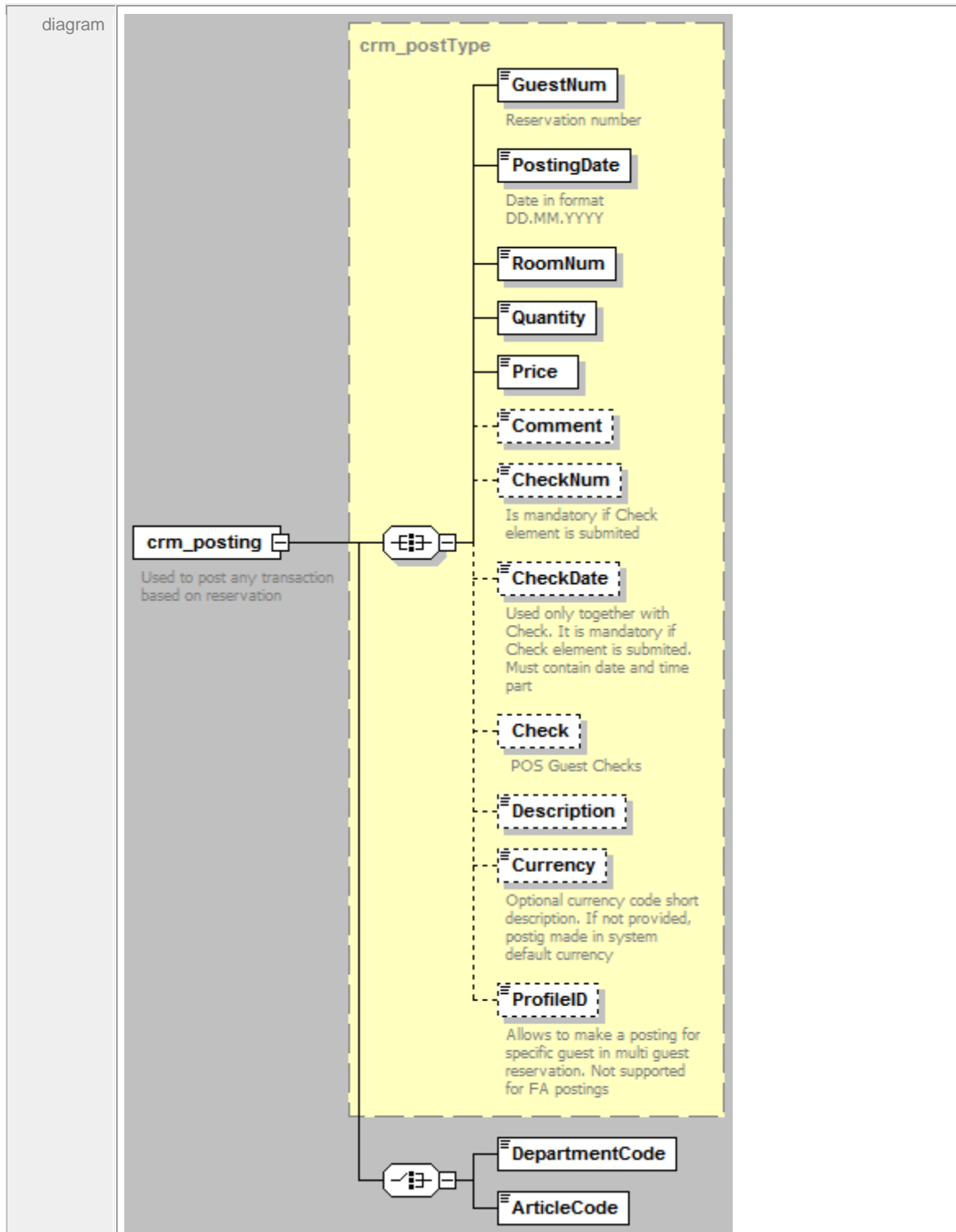


source	<pre> &lt;xs:complexType name="postType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Postings&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:choice&gt; &lt;xs:element name="ccm_book_actuals" type="ccm_book_actualsType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Used to post CCM transaction&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:element&gt; &lt;xs:element name="crm_posting"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Used to post any transaction based on reservation&lt;/xs:documentation&gt; &lt;/xs:documentation&gt;Used to post any transaction based on reservation&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:complexType id="PostingDepartmentChoice"&gt; &lt;xs:complexContent&gt; &lt;xs:extension base="crm_postType"&gt; &lt;xs:choice&gt; &lt;xs:element name="DepartmentCode" type="xs:string"/&gt; &lt;/xs:choice&gt; &lt;/xs:extension&gt; name="ArticleCode" type="xs:string"/&gt; &lt;/xs:choice&gt; &lt;/xs:extension&gt; &lt;/xs:complexContent&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:choice&gt;&lt;/xs:complexType&gt; </pre>
--------	--

**element postType/ccm\_book\_actuals**

diagram	
namespace	V8XMLInterface
type	<b><u>ccm_book_actualsType</u></b>
children	<b><u>bookingID event_id postingDate menu_id menu_item_id misc_item_id function_space_id ResourceName DepartmentCode count total_amount OverrideReason UserName Covers</u></b>
annotation	documentation Used to post CCM transaction
source	<pre> &lt;xs:element name="ccm_book_actuals" type="ccm_book_actualsType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Used to post CCM transaction&lt;/xs:documentation&gt; &lt;/xs:annotation&gt;&lt;/xs:element&gt; </pre>

element postType/crm\_posting



namespace	V8XMLInterface
type	extension of <u>crm_postType</u>
children	<u>GuestNum</u> <u>PostingDate</u> <u>RoomNum</u> <u>Quantity</u> <u>Price</u> <u>Comment</u> <u>CheckNum</u> <u>CheckDate</u> <u>Check</u> <u>Description</u> <u>Currency</u> <u>ProfileID</u> <u>DepartmentCode</u> <u>ArticleCode</u>
annotation	documentation Used to post any transaction based on reservation documentation Used to post any transaction based on reservation

source	<pre> &lt;xs:element name="crm_posting"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Used to post any transaction based on reservation&lt;/xs:documentation&gt; &lt;xs:documentation&gt;Used to post any transaction based on reservation&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:complexType id="PostingDepartmentChoice"&gt; &lt;xs:complexContent&gt; &lt;xs:extension base="crm_postType"&gt; &lt;xs:choice&gt; &lt;xs:element name="DepartmentCode" type="xs:string"/&gt; &lt;xs:element name="ArticleCode" type="xs:string"/&gt; &lt;/xs:choice&gt; &lt;/xs:extension&gt; &lt;/xs:complexContent&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt; </pre>
--------	--

### element postType/crm\_posting/DepartmentCode

diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>
source	<pre>&lt;xs:element name="DepartmentCode" type="xs:string"/&gt;</pre>

### element postType/crm\_posting/ArticleCode

diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>
source	<pre>&lt;xs:element name="ArticleCode" type="xs:string"/&gt;</pre>

### complexType ProfileConditionType

diagram						
namespace	V8XMLInterface					
type	restriction of <b>conditionType</b>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:NMTOKEN	required			
	operation	binaryOperation	required			
	dataType	xs:NMTOKEN	optional			
annotation	documentation Used to define condition for profile search request					
source	<pre> &lt;xs:complexType name="ProfileConditionType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Used to define condition for profile search request&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:simpleContent&gt; &lt;xs:restriction base="conditionType"&gt; &lt;xs:attribute name="name" use="required"&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:NMTOKEN"&gt; &lt;xs:enumeration value="ProfileID"/&gt; &lt;xs:enumeration value="GuestName"/&gt; &lt;xs:enumeration value="GuestFirstname"/&gt; &lt;xs:enumeration value="ProfileType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;1 - Company 2 - Individual 3 - Employee&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:enumeration&gt; &lt;xs:enumeration value="ProfileCategory"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;1=Company 2=Travel Agent 3=Source 4=Individual 5=Vendors 6=Meeting Planner&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:enumeration&gt; &lt;xs:enumeration value="LoginName"/&gt; &lt;xs:enumeration value="ProfileGlobalID"/&gt; &lt;xs:enumeration value="City"/&gt; &lt;xs:enumeration value="ZIP"/&gt; &lt;xs:enumeration value="CentralSyncTime"/&gt; &lt;xs:enumeration value="CorporateID"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleContent&gt;&lt;/xs:complexType&gt; </pre>					

## complexType queryType

<p>diagram</p>																															
<p>namespace</p>	<p>V8XMLInterface</p>																														
<p>children</p>	<p><b><u>conditions</u></b> <b><u>fields</u></b> <b><u>froms</u></b> <b><u>orders</u></b> <b><u>groups</u></b></p>																														
<p>used by</p>	<p>element <b><u>requestType/query</u></b></p>																														
<p>attributes</p>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>ReqType</td> <td>xs:NMTOKEN</td> <td>required</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Name</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Language</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> <td></td> </tr> <tr> <td>TranslationTable</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> <td>documentation If present in the request, short descriptions in request and response will be translated based on the specified translation table</td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	ReqType	xs:NMTOKEN	required				Name	xs:string	optional				Language	xs:string	optional				TranslationTable	xs:string	optional			documentation If present in the request, short descriptions in request and response will be translated based on the specified translation table
Name	Type	Use	Default	Fixed	Annotation																										
ReqType	xs:NMTOKEN	required																													
Name	xs:string	optional																													
Language	xs:string	optional																													
TranslationTable	xs:string	optional			documentation If present in the request, short descriptions in request and response will be translated based on the specified translation table																										
<p>annotation</p>	<p>documentation Types of accepted queriesReqType - type of the rquestName - name of the request. It will be returned as Name attribute for the query element in the response. Used to match response to the request in case more than one query were passed</p>																														
<p>source</p>	<pre>&lt;xs:complexType name="queryType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Types of accepted queriesReqType - type of the rquestName - name of the request. It will be returned as Name attribute for the query element in the response. Used to match response to the request in case more than one query were passed&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:sequence&gt; &lt;xs:element name="conditions" type="conditionsType" minOccurs="0"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Search conditions&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:element&gt; &lt;xs:element name="fields" type="fieldsType" minOccurs="0"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;List of the fields which should be included in the response statementThis element must exist for all kind of queries except SystemDate and SystemInfo&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:element&gt; &lt;xs:element name="froms" minOccurs="0"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Used in CustomQuery request&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:element&gt; &lt;/xs:sequence&gt;</pre>																														

```

<xs:element name="from" type="xs:string" maxOccurs="unbounded"/> </xs:sequence>
</xs:complexType> </xs:element> <xs:element name="orders" minOccurs="0"> <xs:annotation>
<xs:documentation>Used in CustomQuery request. Order By clause</xs:documentation>
</xs:annotation> <xs:complexType> <xs:sequence> <xs:element name="order"
type="xs:string" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element>
<xs:element name="groups" minOccurs="0"> <xs:annotation> <xs:documentation>Used in
CustomQuery request. Group By clause</xs:documentation> </xs:annotation> <xs:complexType>
<xs:sequence> <xs:element name="group" type="xs:string" maxOccurs="unbounded"/>
</xs:sequence> </xs:complexType> </xs:element> <xs:sequence> <xs:attribute name="ReqType"
use="required"> <xs:simpleType> <xs:restriction base="xs:NMTOKEN"> <xs:enumeration
value="Availability"/> <xs:enumeration value="Profile"/> <xs:enumeration value="Reservation"/>
<xs:enumeration value="Department"/> <xs:enumeration value="SystemDate"> <xs:annotation>
<xs:documentation>All conditions and fields list are ignored</xs:documentation> </xs:annotation>
</xs:enumeration> <xs:enumeration value="SystemInfo"> <xs:annotation>
<xs:documentation>All conditions and fields list are ignored</xs:documentation> </xs:annotation>
</xs:enumeration> <xs:enumeration value="Posting"/> <xs:enumeration
value="ChildrenCategories"/> <xs:enumeration value="CreditCardTypes"/> <xs:enumeration
value="UserLogin"/> <xs:enumeration value="HtmlInfo"/> <xs:enumeration
value="RequestGlobalID"/> <xs:enumeration value="SeatingArrangments"/> <xs:enumeration
value="ConfiguredStatusList"/> <xs:enumeration value="ConferencePackagesList"/>
<xs:enumeration value="GetEventInfoBoardDetails"/> <xs:enumeration value="GetActivitiesDetails"/>
<xs:enumeration value="Group"/> <xs:enumeration value="LanguagesLis"/> <xs:enumeration
value="FutureReservations"/> <xs:enumeration value="SystemTime"/> <xs:enumeration
value="Activity"/> <xs:enumeration value="Task"/> <xs:enumeration
value="GetEventInfoBoardDetails"/> <xs:enumeration value="ConferencePackagesList"/>
<xs:enumeration value="ConfiguredArrangments"/> <xs:enumeration value="SeatingArrangments"/>
<xs:enumeration value="GroupTemplate"/> <xs:enumeration value="BlockHeader"/>
<xs:enumeration value="BlockDetail"/> <xs:enumeration value="CurrencyList"/> <xs:enumeration
value="Package"/> </xs:restriction> </xs:simpleType> </xs:attribute> <xs:attribute name="Name"
type="xs:string" use="optional"/> <xs:attribute name="Language" type="xs:string" use="optional">
<xs:annotation> <xs:documentation source="If data in the response should be translated"/>
</xs:annotation> </xs:attribute> <xs:attribute name="TranslationTable" type="xs:string" use="optional">
<xs:annotation> <xs:documentation>If present in the request, short descriptions in request and
response will be translated based on the specified translation table</xs:documentation> </xs:annotation>
</xs:attribute></xs:complexType>

```

**element queryType/conditions**

diagram	
namespace	V8XMLInterface
type	<b><u>conditionsType</u></b>
children	<b><u>condition</u></b> <b><u>link</u></b>
annotation	documentation Search conditions
source	<pre> &lt;xs:element name="conditions" type="conditionsType" minOccurs="0"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Search conditions&lt;/xs:documentation&gt; &lt;/xs:annotation&gt;&lt;/xs:element&gt; </pre>

### element queryType/fields

diagram	<p>List of the fields which should be included in the response statement This element must exist for all kind of queries except SystemDate and SystemInfo</p>
namespace	V8XMLInterface
type	<b>fieldsType</b>
children	<b>field</b>
annotation	documentation List of the fields which should be included in the response statement This element must exist for all kind of queries except SystemDate and SystemInfo
source	<code>&lt;xs:element name="fields" type="fieldsType" minOccurs="0"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;List of the fields which should be included in the response statement This element must exist for all kind of queries except SystemDate and SystemInfo&lt;/xs:documentation&gt; &lt;/xs:annotation&gt;&lt;/xs:element&gt;</code>

### element queryType/froms

diagram	<p>Used in CustomQuery request</p>
namespace	V8XMLInterface
children	<b>from</b>
annotation	documentation Used in CustomQuery request
source	<code>&lt;xs:element name="froms" minOccurs="0"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Used in CustomQuery request&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="from" type="xs:string" maxOccurs="unbounded"/&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt;</code>

### element queryType/froms/from


diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>
source	<code>&lt;xs:element name="from" type="xs:string" maxOccurs="unbounded"/&gt;</code>

### element queryType/orders

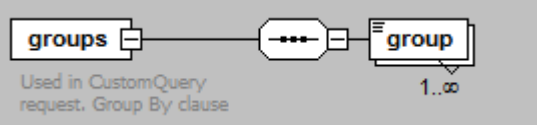
diagram	<p>Used in CustomQuery request, Order By clause</p>
namespace	V8XMLInterface

children	<b><u>order</u></b>
annotation	documentation Used in CustomQuery request. Order By clause
source	<code>&lt;xs:element name="orders" minOccurs="0"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Used in CustomQuery request. Order By clause&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="order" type="xs:string" maxOccurs="unbounded"/&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt;</code>


#### element queryType/orders/order

diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>
source	<code>&lt;xs:element name="order" type="xs:string" maxOccurs="unbounded"/&gt;</code>

#### element queryType/groups

diagram	
namespace	V8XMLInterface
children	<b><u>group</u></b>
annotation	documentation Used in CustomQuery request. Group By clause
source	<code>&lt;xs:element name="groups" minOccurs="0"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Used in CustomQuery request. Group By clause&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:complexType&gt; &lt;xs:sequence&gt; &lt;xs:element name="group" type="xs:string" maxOccurs="unbounded"/&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt;</code>

#### element queryType/groups/group

diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>
source	<code>&lt;xs:element name="group" type="xs:string" maxOccurs="unbounded"/&gt;</code>

## complexType ReportType

diagram													
namespace	V8XMLInterface												
children	<b>ReportID ReportType Parameter Destination ExportFormat Email ConfirmationArchiveRequest</b>												
used by	element <b>requestType/Report</b>												
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>xs:string</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	Name	xs:string				
Name	Type	Use	Default	Fixed	Annotation								
Name	xs:string												
source	<pre> &lt;xs:complexType name="ReportType"&gt; &lt;xs:sequence&gt; &lt;xs:element name="ReportID" type="xs:integer" minOccurs="0"/&gt; &lt;xs:element name="ReportType" minOccurs="0"&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:string"&gt; &lt;xs:enumeration value="Confirmation"/&gt; &lt;xs:enumeration value="CCMConfirmation"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; &lt;/xs:element&gt; &lt;xs:element name="Parameter" minOccurs="0" maxOccurs="unbounded"&gt; &lt;xs:complexType&gt; &lt;xs:attribute name="Name" type="xs:string"/&gt; &lt;xs:attribute name="DataType"&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:string"&gt; &lt;xs:enumeration value="String"/&gt; &lt;xs:enumeration value="Date"/&gt; &lt;xs:enumeration value="Int"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;xs:element name="Destination"&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:string"&gt; &lt;xs:enumeration value="Email"/&gt; &lt;xs:enumeration value="Printer"/&gt; &lt;xs:enumeration value="Export"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; &lt;/xs:element&gt; &lt;xs:element name="ExportFormat" minOccurs="0"&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:string"&gt; &lt;xs:enumeration value="PDF"/&gt; &lt;xs:enumeration value="RTF"/&gt; &lt;xs:enumeration value="Text"/&gt; &lt;xs:enumeration value="HTML"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; &lt;/xs:element&gt; &lt;xs:element name="Email" minOccurs="0"&gt; &lt;xs:complexType&gt; &lt;xs:attribute name="Address" type="xs:string" use="required"/&gt; &lt;xs:attribute name="SendAsAttachment" type="xs:boolean" use="required"/&gt; &lt;xs:attribute name="AttachmentFormat" use="optional"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;If Attachment type is "Text" then report will be sent as a body&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:string"&gt; &lt;xs:enumeration value="RTF"/&gt; &lt;xs:enumeration value="PDF"/&gt; &lt;xs:enumeration value="HTML"/&gt; &lt;xs:enumeration value="Text"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; &lt;xs:attribute name="Subject" type="xs:string" use="optional"/&gt; &lt;xs:attribute name="DefaultBody" type="xs:string" use="optional"/&gt; &lt;xs:attribute name="CC" type="xs:string" use="optional"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;List of CC email addresses separated by comma&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:attribute&gt; &lt;xs:attribute name="BCC" type="xs:string" use="optional"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;List of BCC email addresses separated by comma&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:attribute&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;xs:element name="ConfirmationArchiveRequest" type="xs:integer" minOccurs="0"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Should contain archive record ID (for internal use : yrco_id)&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:element&gt; &lt;/xs:sequence&gt; &lt;xs:attribute name="Name" type="xs:string"/&gt;&lt;/xs:complexType&gt; </pre>												


## element ReportType/ReportID

diagram	
---------	--



namespace	V8XMLInterface
type	<b>xs:integer</b>
source	<code>&lt;xs:element name="ReportID" type="xs:integer" minOccurs="0"/&gt;</code>


### element ReportType/ReportType

diagram	
namespace	V8XMLInterface
type	restriction of <b>xs:string</b>
facets	enumeration Confirmation enumeration CCMConfirmation
source	<code>&lt;xs:element name="ReportType" minOccurs="0"&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:string"&gt; &lt;xs:enumeration value="Confirmation"/&gt; &lt;xs:enumeration value="CCMConfirmation"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;&lt;/xs:element&gt;</code>

### element ReportType/Parameter

diagram																			
namespace	V8XMLInterface																		
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>xs:string</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>DataType</td> <td>xs:string</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	Name	xs:string					DataType	xs:string				
Name	Type	Use	Default	Fixed	Annotation														
Name	xs:string																		
DataType	xs:string																		
source	<code>&lt;xs:element name="Parameter" minOccurs="0" maxOccurs="unbounded"&gt; &lt;xs:complexType&gt; &lt;xs:attribute name="Name" type="xs:string"/&gt; &lt;xs:attribute name="DataType"&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:string"&gt; &lt;xs:enumeration value="String"/&gt; &lt;xs:enumeration value="Date"/&gt; &lt;xs:enumeration value="Int"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt;</code>																		

### element ReportType/Destination

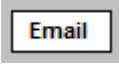
diagram	
namespace	V8XMLInterface
type	restriction of <b>xs:string</b>
facets	enumeration Email enumeration Printer enumeration Export
source	<code>&lt;xs:element name="Destination"&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:string"&gt; &lt;xs:enumeration value="Email"/&gt; &lt;xs:enumeration value="Printer"/&gt; &lt;xs:enumeration value="Export"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;&lt;/xs:element&gt;</code>

### element ReportType/ExportFormat

diagram	
namespace	V8XMLInterface

type	restriction of <b>xs:string</b>
facets	enumeration PDF enumeration RTF enumeration Text enumeration HTML
source	<code>&lt;xs:element name="ExportFormat" minOccurs="0"&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:string"&gt; &lt;xs:enumeration value="PDF"/&gt; &lt;xs:enumeration value="RTF"/&gt; &lt;xs:enumeration value="Text"/&gt; &lt;xs:enumeration value="HTML"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;&lt;/xs:element&gt;</code>

### element ReportType/Email

diagram						
namespace	V8XMLInterface					
attributes	Name	Type	Use	Default	Fixed	Annotation
	Address	xs:string	required			
	SendAsAttachment	xs:boolean	required			
	AttachmentFormat	xs:string	optional			documentation If Attachment type is "Text" then report will be sent as a body
	Subject	xs:string	optional			
	DefaultBody	xs:string	optional			
	CC	xs:string	optional			documentation List of CC email addresses separated by comma
	BCC	xs:string	optional			documentation List of BCC email addresses separated by comma
source	<pre>&lt;xs:element name="Email" minOccurs="0"&gt; &lt;xs:complexType&gt; &lt;xs:attribute name="Address" type="xs:string" use="required"/&gt; &lt;xs:attribute name="SendAsAttachment" type="xs:boolean" use="required"/&gt; &lt;xs:attribute name="AttachmentFormat" use="optional"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;If Attachment type is "Text" then report will be sent as a body&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:string"&gt; &lt;xs:enumeration value="RTF"/&gt; &lt;xs:enumeration value="PDF"/&gt; &lt;xs:enumeration value="HTML"/&gt; &lt;xs:enumeration value="Text"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; &lt;xs:attribute name="Subject" type="xs:string" use="optional"/&gt; &lt;xs:attribute name="DefaultBody" type="xs:string" use="optional"/&gt; &lt;xs:attribute name="CC" type="xs:string" use="optional"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;List of CC email addresses separated by comma&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:attribute&gt; &lt;xs:attribute name="BCC" type="xs:string" use="optional"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;List of BCC email addresses separated by comma&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:attribute&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt;</pre>					

### element ReportType/ConfirmationArchiveRequest

diagram	
namespace	V8XMLInterface

type	<b>xs:integer</b>
annotation	documentation Should contain archive record ID (for internal use : yrco_id)
source	<xs:element name="ConfirmationArchiveRequest" type="xs:integer" minOccurs="0"> <xs:annotation> <xs:documentation>Should contain archive record ID (for internal use : yrco_id)</xs:documentation> </xs:annotation></xs:element>

### complexType requestType

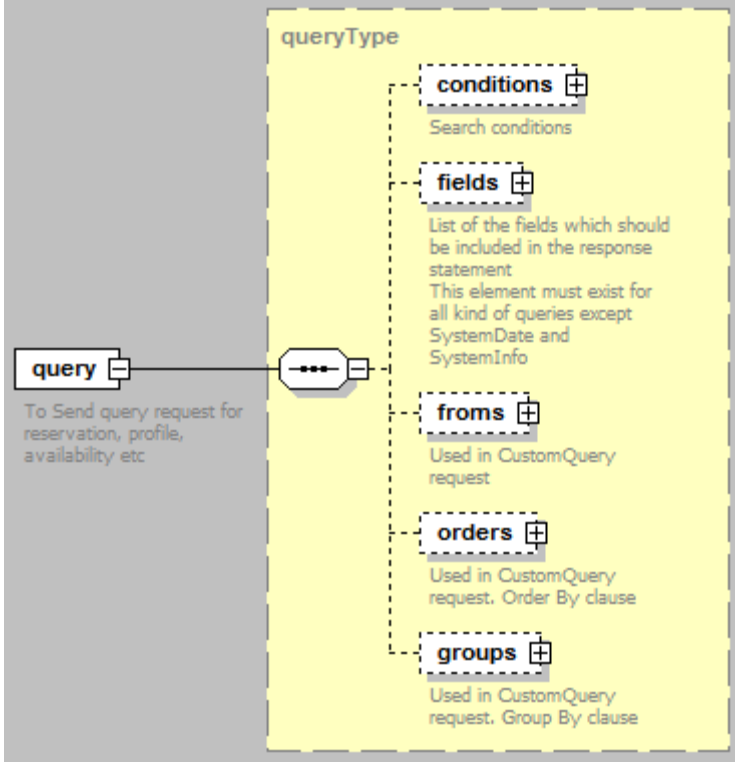
diagram	
namespace	V8XMLInterface
children	<b><u>query</u></b> <b><u>update</u></b> <b><u>insert</u></b> <b><u>command</u></b> <b><u>AvailabilityForWeb</u></b> <b><u>ValidateCreditCard</u></b> <b><u>import</u></b> <b><u>HotelAvailability</u></b> <b><u>Report</u></b>
used by	element <b><u>fidelio/request</u></b>
annotation	documentation Defines a request figure
source	<xs:complexType name="requestType"> <xs:annotation> <xs:documentation>Defines a request figure</xs:documentation> </xs:annotation> <xs:choice> <xs:element name="query" maxOccurs="unbounded"> <xs:annotation> <xs:documentation>To Send query request for reservation, profile, availability etc</xs:documentation> </xs:annotation> <xs:complexType> <xs:complexContent> <xs:extension base="queryType"/> </xs:complexContent> </xs:complexType> </xs:element> <xs:element name="update" type="updateType"

```

maxOccurs="unbounded"> <xs:annotation> <xs:documentation>To Send update request for
profile, etc</xs:documentation> </xs:annotation> </xs:element> <xs:element name="insert"
type="insertType" maxOccurs="unbounded"> <xs:annotation> <xs:documentation>To Send create
request for profile, etc</xs:documentation> </xs:annotation> </xs:element> <xs:element
name="command" maxOccurs="unbounded"> <xs:complexType> <xs:complexContent>
<xs:extension base="commandType"> <xs:attribute name="Name" type="xs:string"
use="optional"/> <xs:attribute name="TranslationTable" type="xs:string" use="optional">
</xs:annotation> <xs:documentation>If present in the request, short descriptions in request and
response will be translated based on the specified translation table</xs:documentation>
</xs:annotation> </xs:attribute> </xs:extension> </xs:complexContent>
</xs:complexType> </xs:element> <xs:element name="AvailabilityForWeb"
type="AvailabilityForWebType" maxOccurs="unbounded"> <xs:annotation>
<xs:documentation>Rate availability including room availability figures</xs:documentation>
</xs:annotation> </xs:element> <xs:element name="ValidateCreditCard"
type="ValidateCreditCardType" maxOccurs="unbounded"/> <xs:element name="import"
type="importType"/> <xs:element name="HotelAvailability" type="HotelAvailabilityType"
maxOccurs="unbounded"> <xs:annotation> <xs:documentation>Room Availability
</xs:documentation> </xs:annotation> </xs:element> <xs:element name="Report"
type="ReportType" maxOccurs="unbounded"/> </xs:choice></xs:complexType>

```

**element requestType/query**

<p>diagram</p> 																															
<p>namespace</p>	<p>V8XMLInterface</p>																														
<p>type</p>	<p>extension of <b>queryType</b></p>																														
<p>children</p>	<p><b>conditions fields froms orders groups</b></p>																														
<p>attributes</p>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>ReqType</td> <td>xs:NMTOKEN</td> <td>required</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Name</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Language</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> <td></td> </tr> <tr> <td>TranslationTable</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> <td>documentation If present in the request, short descriptions in request and</td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	ReqType	xs:NMTOKEN	required				Name	xs:string	optional				Language	xs:string	optional				TranslationTable	xs:string	optional			documentation If present in the request, short descriptions in request and
Name	Type	Use	Default	Fixed	Annotation																										
ReqType	xs:NMTOKEN	required																													
Name	xs:string	optional																													
Language	xs:string	optional																													
TranslationTable	xs:string	optional			documentation If present in the request, short descriptions in request and																										

		response will be translated based on the specified translation table
annotation	documentation	To Send query request for reservation, profile, availability etc
source	<pre>&lt;xs:element name="query" maxOccurs="unbounded"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;To Send query request for reservation, profile, availability etc&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:complexType&gt; &lt;xs:complexContent&gt; &lt;xs:extension base="queryType"/&gt; &lt;/xs:complexContent&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt;</pre>	

**element requestType/update**

diagram						
namespace	V8XMLInterface					
type	<b>updateType</b>					
children	<b>conditions fields</b>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	ReqType	xs:NMTOKEN	required			
	Name	xs:string	optional			
	TranslationTable	xs:string	optional			documentation If present in the request, short descriptions in request and response will be translated based on the specified translation table
annotation	documentation	To Send update request for profile, etc				
source	<pre>&lt;xs:element name="update" type="updateType" maxOccurs="unbounded"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;To Send update request for profile, etc&lt;/xs:documentation&gt; &lt;/xs:annotation&gt;&lt;/xs:element&gt;</pre>					

**element requestType/insert**

diagram						
namespace	V8XMLInterface					
type	<b>insertType</b>					
children	<b>fields</b>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	ReqType	xs:NMTOKEN	required			
	Name	xs:string	optional			
	TranslationTable	xs:string	optional			documentation If present in the request, short descriptions in request and response will be translated based on the specified translation table
annotation	documentation To Send create request for profile, etc					
source	<pre>&lt;xs:element name="insert" type="insertType" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;To Send create request for profile, etc&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>					

**element requestType/command**

diagram						
namespace	V8XMLInterface					
type	extension of <b>commandType</b>					
children	<b>post checkIn checkOut cancel resetProfilePassword</b>					

attributes	Name Name	Type xs:string	Use optional	Default	Fixed	Annotation
	TranslationTable	xs:string	optional			documentation If present in the request, short descriptions in request and response will be translated based on the specified translation table
source	<pre>&lt;xs:element name="command" maxOccurs="unbounded"&gt; &lt;xs:complexType&gt; &lt;xs:complexContent&gt; &lt;xs:extension base="commandType"&gt; &lt;xs:attribute name="Name" type="xs:string" use="optional"/&gt; &lt;xs:attribute name="TranslationTable" type="xs:string" use="optional"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;If present in the request, short descriptions in request and response will be translated based on the specified translation table&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:attribute&gt; &lt;/xs:extension&gt; &lt;/xs:complexContent&gt; &lt;/xs:complexType&gt;&lt;/xs:element&gt;</pre>					

### element requestType/AvailabilityForWeb

diagram	<pre> classDiagram     class AvailabilityForWebType {         AvailabilityForWeb         fields     }     class AvailabilityForWeb {         "Rate availability including room availability figures"     }     AvailabilityForWebType -- AvailabilityForWeb     AvailabilityForWebType -- fields </pre>					
namespace	V8XMLInterface					
type	<b><u>AvailabilityForWebType</u></b>					
children	<b><u>fields</u></b>					
attributes	Name Name	Type xs:string	Use optional	Default	Fixed	Annotation
	Language	xs:string	optional			
	TranslationTable	xs:string	optional			documentation If present in the request, short descriptions in request and response will be translated based on the specified translation table
annotation	documentation Rate availability including room availability figures					
source	<pre>&lt;xs:element name="AvailabilityForWeb" type="AvailabilityForWebType" maxOccurs="unbounded"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Rate availability including room availability figures&lt;/xs:documentation&gt; &lt;/xs:annotation&gt;&lt;/xs:element&gt;</pre>					

**element** requestType/ValidateCreditCard

diagram													
namespace	V8XMLInterface												
type	<b><u>ValidateCreditCardType</u></b>												
children	<b><u>CreditCardNo</u></b> <b><u>ExpiryMonth</u></b> <b><u>ExpiryYear</u></b> <b><u>CardholderName</u></b> <b><u>CreditCardCode</u></b>												
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	Name	xs:string	optional			
Name	Type	Use	Default	Fixed	Annotation								
Name	xs:string	optional											
source	<code>&lt;xs:element name="ValidateCreditCard" type="ValidateCreditCardType" maxOccurs="unbounded"/&gt;</code>												

**element** requestType/import

diagram																									
namespace	V8XMLInterface																								
type	<b><u>importType</u></b>																								
children	<b><u>insert</u></b> <b><u>update</u></b> <b><u>delete</u></b>																								
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>TableName</td> <td>xs:string</td> <td>required</td> <td></td> <td></td> <td></td> </tr> <tr> <td>HotelID</td> <td>xs:string</td> <td>required</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Name</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	TableName	xs:string	required				HotelID	xs:string	required				Name	xs:string	optional			
Name	Type	Use	Default	Fixed	Annotation																				
TableName	xs:string	required																							
HotelID	xs:string	required																							
Name	xs:string	optional																							
source	<code>&lt;xs:element name="import" type="importType"/&gt;</code>																								

**element** requestType/HotelAvailability

diagram	
namespace	V8XMLInterface



type	<b>HotelAvailabilityType</b>					
children	<b>fields</b>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	Name	xs:string	optional			
	TranslationTable	xs:string	optional			
annotation	documentation Room Availability					
source	<pre>&lt;xs:element name="HotelAvailability" type="HotelAvailabilityType" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Room Availability &lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;&lt;/xs:element&gt;</pre>					

### element requestType/Report

diagram						
namespace	V8XMLInterface					
type	<b>ReportType</b>					
children	<b>ReportID ReportType Parameter Destination ExportFormat Email ConfirmationArchiveRequest</b>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	Name	xs:string				
source	<pre>&lt;xs:element name="Report" type="ReportType" maxOccurs="unbounded"/&gt;</pre>					

### complexType ReservationConditionType

diagram						
namespace	V8XMLInterface					
type	restriction of <b>conditionType</b>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:NMTOKEN	required			

	operation	binaryOperation	required
	dataType	xs:NMTOKEN	optional
annotation	documentation	Used to define condition for reservation search request	
source	<pre> &lt;xs:complexType name="ReservationConditionType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Used to define condition for reservation search request&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:simpleContent&gt; &lt;xs:restriction base="conditionType"&gt; &lt;xs:attribute name="name" use="required"&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:NMTOKEN"&gt; &lt;xs:enumeration value="GuestFirstname"/&gt; &lt;xs:enumeration value="GuestName"/&gt; &lt;xs:enumeration value="RoomNum"/&gt; &lt;xs:enumeration value="RoomType"/&gt; &lt;xs:enumeration value="GuestArrival"/&gt; &lt;xs:enumeration value="GuestNum"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Means Reservation Number&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:enumeration&gt; &lt;xs:enumeration value="ReservationStatus"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Allowed values are: Reservation                                0 - In   1 - Checked Out  2 - Checked Canceled                                    3 - Show   4 - No &lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:enumeration&gt; &lt;xs:enumeration value="ProfileID"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleContent&gt;&lt;/xs:complexType&gt; </pre>		

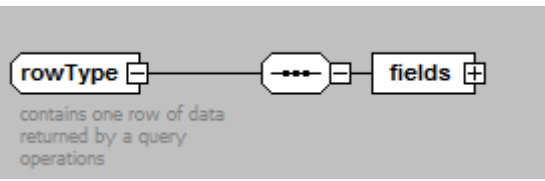
### complexType rowsType

diagram	
namespace	V8XMLInterface
children	<b>row</b>
used by	elements <b>fidelio/response/queryResponse/rows</b> <b>fidelio/response/updateResponse/rows</b> <b>importType/insert/rows</b>
annotation	documentation contains a set of rows returned by a query operations
source	<pre> &lt;xs:complexType name="rowsType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;contains a set of rows returned by a query operations&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:sequence&gt; &lt;xs:element name="row" type="rowType" minOccurs="0" maxOccurs="unbounded"/&gt; &lt;/xs:sequence&gt;&lt;/xs:complexType&gt; </pre>

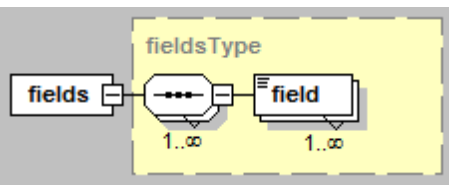
### element rowsType/row

diagram	
namespace	V8XMLInterface
type	<b>rowType</b>
children	<b>fields</b>
source	<pre> &lt;xs:element name="row" type="rowType" minOccurs="0" maxOccurs="unbounded"/&gt; </pre>

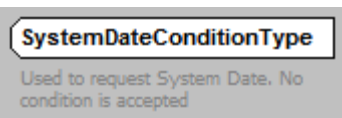
## complexType rowType

diagram	 <p>contains one row of data returned by a query operations</p>
namespace	V8XMLInterface
children	<b>fields</b>
used by	element <b>rowsType/row</b>
annotation	documentation contains one row of data returned by a query operations
source	<pre>&lt;xs:complexType name="rowType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;contains one row of data returned by a query operations&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:sequence&gt; &lt;xs:element name="fields" type="fieldsType"/&gt; &lt;/xs:sequence&gt;&lt;/xs:complexType&gt;</pre>

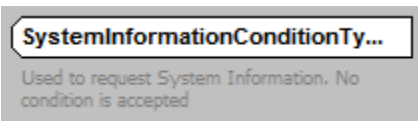
## element rowType/fields

diagram	
namespace	V8XMLInterface
type	<b>fieldsType</b>
children	<b>field</b>
source	<pre>&lt;xs:element name="fields" type="fieldsType"/&gt;</pre>

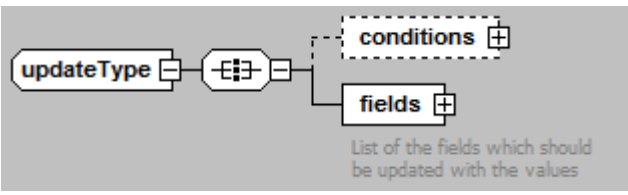
## complexType SystemDateConditionType

diagram	 <p>Used to request System Date. No condition is accepted</p>																								
namespace	V8XMLInterface																								
type	restriction of <b>conditionType</b>																								
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>name</td> <td>xs:anySimpleType</td> <td>prohibited</td> <td></td> <td></td> <td></td> </tr> <tr> <td>operation</td> <td>binaryOperation</td> <td>prohibited</td> <td></td> <td></td> <td></td> </tr> <tr> <td>dataType</td> <td>xs:NMTOKEN</td> <td>optional</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	name	xs:anySimpleType	prohibited				operation	binaryOperation	prohibited				dataType	xs:NMTOKEN	optional			
Name	Type	Use	Default	Fixed	Annotation																				
name	xs:anySimpleType	prohibited																							
operation	binaryOperation	prohibited																							
dataType	xs:NMTOKEN	optional																							
annotation	documentation Used to request System Date. No condition is accepted																								
source	<pre>&lt;xs:complexType name="SystemDateConditionType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Used to request System Date. No condition is accepted&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:simpleContent&gt; &lt;xs:restriction base="conditionType"&gt; &lt;xs:attribute name="operation" use="prohibited"/&gt; &lt;xs:attribute name="name" use="prohibited"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleContent&gt;&lt;/xs:complexType&gt;</pre>																								

## complexType SystemInformationConditionType

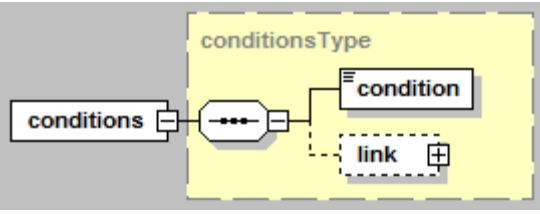
diagram						
namespace	V8XMLInterface					
type	restriction of <b>conditionType</b>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:anySimpleType	prohibited			
	operation	binaryOperation	prohibited			
	dataType	xs:NMTOKEN	optional			
annotation	documentation Used to request System Information. No condition is accepted					
source	<pre>&lt;xs:complexType name="SystemInformationConditionType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Used to request System Information. No condition is accepted&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:simpleContent&gt; &lt;xs:restriction base="conditionType"&gt; &lt;xs:attribute name="operation" use="prohibited"/&gt; &lt;xs:attribute name="name" use="prohibited"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleContent&gt;&lt;/xs:complexType&gt;</pre>					

## complexType updateType

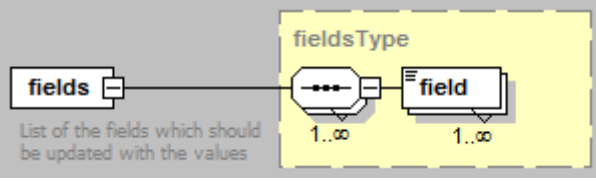
diagram						
namespace	V8XMLInterface					
children	<b>conditions fields</b>					
used by	elements <b>requestType/update importType/update</b>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	ReqType	xs:NMTOKEN	required			
	Name	xs:string	optional			
	TranslationTable	xs:string	optional			documentation If present in the request, short descriptions in request and response will be translated based on the specified translation table
source	<pre>&lt;xs:complexType name="updateType"&gt; &lt;xs:all&gt; &lt;xs:element name="conditions" type="conditionsType" minOccurs="0"/&gt; &lt;xs:element name="fields" type="fieldsType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;List of the fields which should be updated with the values&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:element&gt; &lt;/xs:all&gt; &lt;xs:attribute name="ReqType" use="required"&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:NMTOKEN"&gt; &lt;xs:enumeration value="Profile"/&gt; &lt;xs:enumeration value="Reservation"/&gt; &lt;xs:enumeration value="Group"/&gt; &lt;xs:enumeration value="GroupTemplate"/&gt; &lt;xs:enumeration value="Activity"/&gt; &lt;xs:enumeration value="Task"/&gt; &lt;xs:enumeration value="HurdleRate"/&gt; &lt;xs:enumeration value="Overbooking"/&gt; &lt;/xs:enumeration</pre>					

	<pre> value="Report"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; &lt;xs:attribute name="Name" type="xs:string" use="optional"/&gt; &lt;xs:attribute name="TranslationTable" type="xs:string" use="optional"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;If present in the request, short descriptions in request and response will be translated based on the specified translation table&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;&lt;/xs:complexType&gt; </pre>
--	---

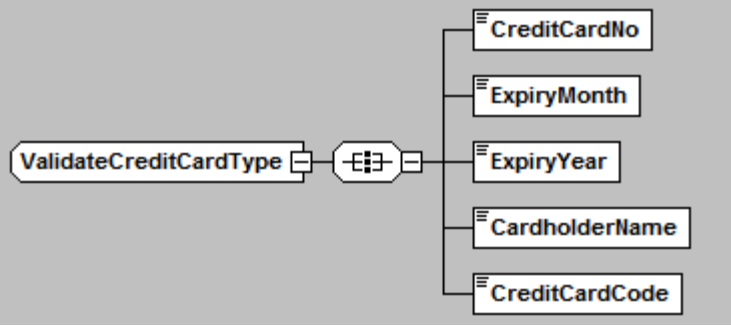
### element updateType/conditions

diagram	 <p>The diagram shows a box labeled 'conditions' connected to a container labeled 'conditionsType'. Inside 'conditionsType', there is a 'condition' box and a 'link' box. The 'condition' box is connected to the 'conditionsType' container, and the 'link' box is connected to the 'condition' box.</p>
namespace	V8XMLInterface
type	<u>conditionsType</u>
children	<u>condition</u> <u>link</u>
source	<code>&lt;xs:element name="conditions" type="conditionsType" minOccurs="0"/&gt;</code>

### element updateType/fields

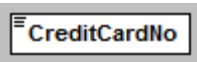
diagram	 <p>The diagram shows a box labeled 'fields' connected to a container labeled 'fieldsType'. Inside 'fieldsType', there is a 'field' box. The 'field' box is connected to the 'fieldsType' container. Below the 'field' box, the cardinality '1..∞' is indicated. A note below the diagram says: 'List of the fields which should be updated with the values'.</p>
namespace	V8XMLInterface
type	<u>fieldsType</u>
children	<u>field</u>
annotation	documentation List of the fields which should be updated with the values
source	<code>&lt;xs:element name="fields" type="fieldsType"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;List of the fields which should be updated with the values&lt;/xs:documentation&gt; &lt;/xs:annotation&gt;&lt;/xs:element&gt;</code>

### complexType ValidateCreditCardType


diagram	 <p>The diagram shows a box labeled 'ValidateCreditCardType' connected to a container labeled 'ValidateCreditCardType'. Inside the container, there are five boxes representing child elements: 'CreditCardNo', 'ExpiryMonth', 'ExpiryYear', 'CardholderName', and 'CreditCardCode'.</p>
namespace	V8XMLInterface
children	<u>CreditCardNo</u> <u>ExpiryMonth</u> <u>ExpiryYear</u> <u>CardholderName</u> <u>CreditCardCode</u>
used by	element <u>requestType/ValidateCreditCard</u>

attributes	Name	Type	Use	Default	Fixed	Annotation
	Name	xs:string	optional			
source	<pre>&lt;xs:complexType name="ValidateCreditCardType"&gt; &lt;xs:all&gt; &lt;xs:element name="CreditCardNo" type="xs:string"/&gt; &lt;xs:element name="ExpiryMonth" type="xs:string"/&gt; &lt;xs:element name="ExpiryYear" type="xs:string"/&gt; &lt;xs:element name="CardholderName" type="xs:string"/&gt; &lt;xs:element name="CreditCardCode" type="xs:string"/&gt; &lt;/xs:all&gt; &lt;xs:attribute name="Name" type="xs:string" use="optional"/&gt;&lt;/xs:complexType&gt;</pre>					


#### element ValidateCreditCardType/CreditCardNo

diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>
source	<pre>&lt;xs:element name="CreditCardNo" type="xs:string"/&gt;</pre>

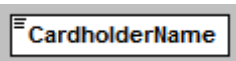
#### element ValidateCreditCardType/ExpiryMonth

diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>
source	<pre>&lt;xs:element name="ExpiryMonth" type="xs:string"/&gt;</pre>

#### element ValidateCreditCardType/ExpiryYear

diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>
source	<pre>&lt;xs:element name="ExpiryYear" type="xs:string"/&gt;</pre>

#### element ValidateCreditCardType/CardholderName

diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>
source	<pre>&lt;xs:element name="CardholderName" type="xs:string"/&gt;</pre>

#### element ValidateCreditCardType/CreditCardCode

diagram	
namespace	V8XMLInterface
type	<b>xs:string</b>

source	<code>&lt;xs:element name="CreditCardCode" type="xs:string"/&gt;</code>
--------	---

### simpleType binaryOperation

namespace	V8XMLInterface
type	restriction of <b>xs:string</b>
used by	attribute <b><u>conditionType/@operation</u></b>
facets	enumeration lt enumeration le enumeration gt enumeration ge enumeration eq enumeration neq enumeration in enumeration like
annotation	documentation defines a binary operation documentation lt = less documentation le = less or equal documentation gt = greater documentation ge = greater or equal documentation eq = equal documentation neq = not equal documentation like = oracle like condition
source	<code>&lt;xs:simpleType name="binaryOperation"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;defines a binary operation&lt;/xs:documentation&gt; &lt;xs:documentation&gt;lt = less&lt;/xs:documentation&gt; &lt;xs:documentation&gt;le = less or equal&lt;/xs:documentation&gt; &lt;xs:documentation&gt;gt = greater&lt;/xs:documentation&gt; &lt;xs:documentation&gt;ge = greater or equal&lt;/xs:documentation&gt; &lt;xs:documentation&gt;eq = equal&lt;/xs:documentation&gt; &lt;xs:documentation&gt;neq = not equal&lt;/xs:documentation&gt; &lt;xs:documentation&gt;like = oracle like condition&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:restriction base="xs:string"&gt; &lt;xs:enumeration value="lt"/&gt; &lt;xs:enumeration value="le"/&gt; &lt;xs:enumeration value="gt"/&gt; &lt;xs:enumeration value="ge"/&gt; &lt;xs:enumeration value="eq"/&gt; &lt;xs:enumeration value="neq"/&gt; &lt;xs:enumeration value="in"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;Not supported yet&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:enumeration&gt; &lt;xs:enumeration value="like"/&gt; &lt;/xs:restriction&gt;&lt;/xs:simpleType&gt;</code>

### simpleType conditionSimpleType

namespace	V8XMLInterface
type	<b>xs:string</b>
used by	complexType <b><u>conditionType</u></b>
source	<code>&lt;xs:simpleType name="conditionSimpleType"&gt; &lt;xs:restriction base="xs:string"/&gt;&lt;/xs:simpleType&gt;</code>