# Oracle Hospitality Integration Platform

## BE Meetings and Events

Release 25.1
G29723-02
August 2025

Oracle Hospitality Integration Platform BE Meetings and Events, Release 25.1

G29723-02

# Contents

# Preface

Oracle Hospitality Integration Cloud Service and OPERA Cloud Foundation users are authorized to access the following modules and features:

- Oracle Hospitality Integration Platform including Oracle Hospitality Developer Portal and Hospitality REST APIs.

The Oracle Hospitality Developer Portal enables users to discover Oracle Hospitality APIs, subscribe to Oracle Hospitality APIs, and to get the necessary information to consume them.

The Oracle Hospitality Developer Portal's APIs page provides information about published Oracle Hospitality APIs. Here you can find and evaluate Oracle Hospitality APIs to use with your applications.

After you discover the Oracle Hospitality APIs that you want to use, register an application and then register those APIs to your application.

To view more details about the Oracle Hospitality Developer Portal, on the user menu drop-down, click the drop-down, and select **About Developer Portal**.

### Purpose

This guide describes the meeting room booking engine.

### Audience

This guide is intended for customers and partners who develop applications with the Oracle Hospitality Integration Platform.

### Documentation

Oracle Hospitality product documentation is available on the Oracle Help Center at https://docs.oracle.com/en/industries/hospitality/.

### Revision History

| Date | Description of Change |
|---|---|
| June 2025 | Initial publication |
| August 2025 | Formatting changes |

1

# Meetings and Events Overview

### Business Context

A meeting room booking engine is a one-stop shop for event organizers to book function spaces for corporate meetings, conferences, or private events. This may include additional requirements, such as coffee breaks, meals, or audio-visual equipment. The meeting room booking engine also provides the ability to book hotel rooms for attendees if needed.

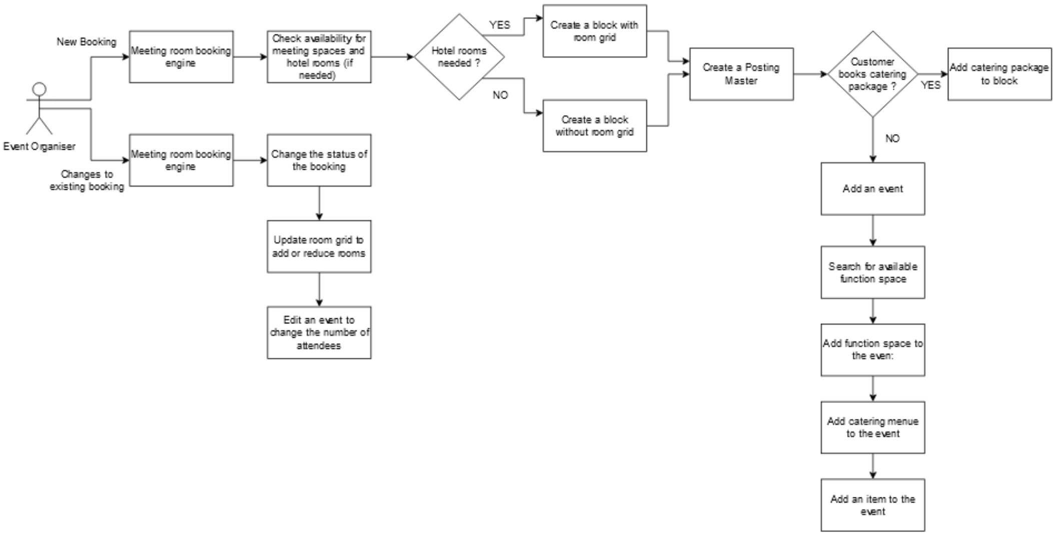
The meeting room booking engine is a software application for reserving meeting spaces and rooms in a property and for streamlining the booking process by providing real-time availability, pricing, and instant confirmation. This enhances efficiency for both the property and the customer, reduces the administrative workload, and improves customer satisfaction by offering a convenient, 24/7 accessible platform for group reservations and meeting planning.

> ⓘ **Note**
>
> API usage and capabilities are subject to change. For more information on API versioning, see the Versioning topic in the Oracle Hospitality Integration Platform User Guide.
>
> For the most comprehensive and up-to-date information, including additional query parameters and detailed descriptions, always refer to the latest API specifications in GitHub. This will ensure that you are accessing the most current and complete set of features available.

### Workflow Diagram

The below diagram illustrates the high-level process for how an event organizer is planning and booking an event, making changes, and expecting those changes to be reflected and updated in real time. All this can be accomplished with the block and event REST APIs.

**Figure 1-1    Meetings and Events Sample Workflow**


New Booking
Meeting room booking engine
Check availability for meeting spaces and hotel rooms (if needed)
Hotel rooms needed ?
YES
Create a block with room grid
NO
Create a block without room grid
Create a Posting Master
Customer books catering package ?
YES
Add catering package to block
NO
Add an event
Search for available function space
Add function space to the even:
Add catering menue to the event
Add an item to the event
Event Organiser
Changes to existing booking
Meeting room booking engine
Change the status of the booking
Update room grid to add or reduce rooms
Edit an event to change the number of attendees

To address the variety of use cases for a meeting room booking engine, this implementation guide contains the following chapters:

1 - Create a Block with or without Room Grid

2 - Create Block Posting Master

3 - Create Event with Function Space

4 - Change Block Status

5 - Update Room and Rate Grid

6 - Edit an Event (for example, Number of Attendees or Event Description)

7 - Add or Change a Function Space for an Existing Event

8 - Add a Catering Menu to an Event

9 - Add an Item to an Event

10 - Add a Catering Package to a Block

# 2 Create a Block with or without Room Grid

**Business Context**

A block is an allocation of rooms for multiple room types (reserved for a specific date period) that does not require you to create individually named reservations. Blocks are used to reserve multiple rooms for events, such as weddings or conferences, inbound tour groups or airline crews, or for long-term allocations, such as airline allotments. You can have the following types of blocks:

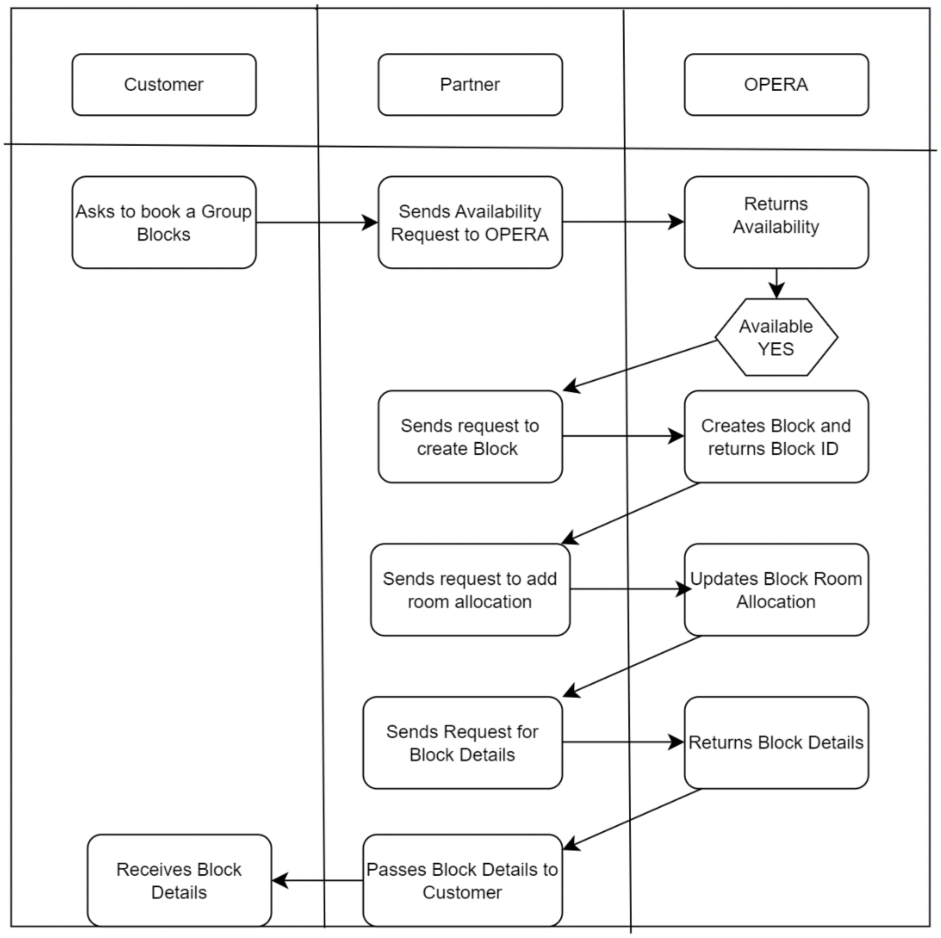
1. Blocks with rooms only (that is, wedding room blocks where the bride and groom reserve rooms for their attendees to book at the discounted group rate).
2. Blocks with rooms and events (that is, conventions where the group has rooms and meetings at the same property).
3. Blocks with events only (that is, local companies that decide to meet off premises to have their meetings at a hotel facility).

In examples 1 and 2, you need a room grid; therefore, you must create a block with room allocation since you have rooms allocated for your group. If you are in the scenario of example 3, then you do not need a room allocation as all you need to create is an event space.

The below workflow diagram explains the use case for creating a block with room allocation.

- Customer — The groom wants to reserve a block with 10 rooms for 2 nights for the wedding weekend at the property nearby the wedding venue. The rooms must be set aside at a group rate for the guests attending the wedding.
- Partner — The partner sends the requests to the property to check availability. The response determines the availability at the property. In the case of availability, the next step is the block creation and the room allocation.
- OPERA Cloud — The application checks for availability, returns availability (if any), creates the block, and returns the final block with details and rooms allocated.

**Figure 2-1 Workflow Diagram**


Customer
Partner
OPERA
Asks to book a Group Blocks
Sends Availability Request to OPERA
Returns Availability
Available YES
Sends request to create Block
Creates Block and returns Block ID
Sends request to add room allocation
Updates Block Room Allocation
Sends Request for Block Details
Returns Block Details
Receives Block Details
Passes Block Details to Customer

## Prerequisites

**Table 2-1 Required Software Tools**

| Tool | Description | Links |
|---|---|---|
| Postman | Postman is an API Platform that allows you to design, create and test API's.<br>Use Postman to send API requests or to use the Postman collections provided. | Using the Oracle Hospitality APIs<br>Tool |

**Table 2-2 Current Versions Required**

| OPERA Cloud Platform Module | Description | Minimum Version |
|---|---|---|
| OPERA Cloud Services | The customer must have a subscription to OPERA Cloud Foundation | |
| Oracle Hospitality Integration Platform (OHIP) | For customers, OHIP is included in the subscription to OPERA Cloud Foundation.<br>Partners need a subscription to the Oracle Hospitality Integration Cloud Service. | |

### Configuration of OPERA Controls

The below software tools are required to follow the steps listed:

To activate the Create Block functionality, you must ensure the following OPERA Controls are active in the target property. You can use the following API operations to validate the settings:

**Ensure that application function BLOCKS - BUSINESS BLOCK is active**

RQ: ent/config/v1/settings?hotelId={{HotelId}}&parameterNameWildCard=BUSINESS BLOCK

**Check if the site is using OCCUPANCY SPLIT BY ROOM TYPE**

RQ: ent/config/v1/settings?hotelId={{HotelId}}&parameterNameWildCard=OCCUPANCY SPLIT PER ROOM TYPE

**Check if the site is using RATE CODE MANDATORY BLOCK HEADER**

RQ: ent/config/v1/settings?hotelId={{HotelId}}&parameterNameWildCard=RATE CODE MANDATORY BLOCK HEADER

**Check if the site is using USE SINGLE BLOCK STATUS**

RQ: ent/config/v1/settings?hotelId={{HotelId}}&parameterNameWildCard=USE SINGLE BLOCK STATUS

### Mandatory Fields

The creation of a block requires to pass certain mandatory fields. Use the API operations below to retrieve the available, valid values:

**Market Segment**

RQ: rsv/config/v1/marketCodes?includeInactive=false&hotelIds={{hotelId}}

**Source Code**

RQ: rsv/config/v1/hotels/{{hotelId}}/sourceCodes?includeInactive=false&hotelIds={{hotelId}}

**Block Status**

RQ: blk/config/v1/blockStatusCodes

**Reservation Type**

RQ: rsv/config/v1/guaranteeCodes?pageNumber=1&hotelIds={{hotelId}}

**Rate Code (if mandatory)**

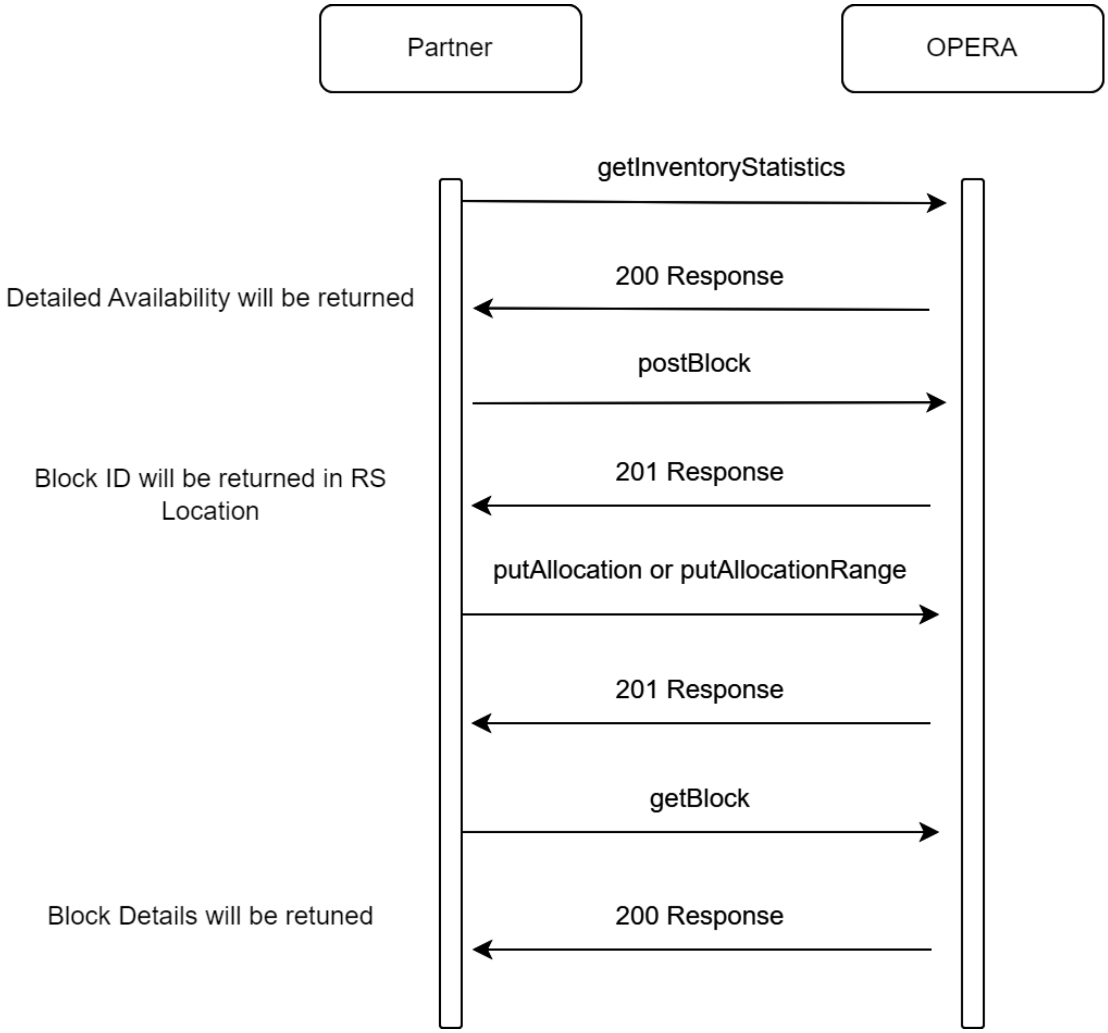
RQ: rtp/v1/ratePlans?includeRateInformation=false&includeInactive=true&offset=1&limit=200&sellDate=2024-02-08&hotelId={{hotelId}}

**Room Types**

RQ: rm/config/v1/hotels/{{hotelId}}/roomTypes?accessible=false&includeInactive=false&offset=1&hotelIds={{hoteldId}}&summaryInfo=true&limit=25&physical=true&pseudo=false

### Workflow

**Figure 2-2 The sequence of API calls (in any direction)**


Partner
OPERA
getInventoryStatistics
200 Response
Detailed Availability will be returned
postBlock
201 Response
Block ID will be returned in RS Location
putAllocation or putAllocationRange
201 Response
getBlock
Block Details will be retuned
200 Response

### Description of Steps

A typical process to create a new block header with room grid is as follows:

1. Search for guest room availability for the block date(s) using **getInventoryStatistics**.
2. Create the block header using **postBlock**.
   a. Ensure that mandatory fields Market Code, Source Code, Status, and Reservation Type are sent with valid values. Note that other fields may be mandatory depending on the application settings.
   b. Ensure that the block status code is a Starting Status.
      i. The blk/config/v1/blockStatusCodes returns that information.
   c. Ensure that the reservation type sent is correct based on the block status code. Blocks on INQUIRY or NON DEDUCT status require a non-deduct reservation type. Blocks on DEDUCT or ACTUAL status require a DEDUCT reservation type.

      i. rsv/config/v1/guaranteeCodes returns the status information.

   d. If you want to add events to this block, ensure that the elements for cateringStatus and eventAttendees are present and contain valid values. If the block does not have any events attached, these elements can be left blank.

   e. If the parameter 'BLOCKS - USE SINGLE BLOCK STATUS' is active, ensure that the element for cateringStatus always carries the same value as the element bookingStatus. The element eventAttendees cannot be null in this scenario (but it can be zero).

3. Once the block ID of the newly created Block is returned in the location header, add the room block using putBlockAllocation or putBlockAllocationRange

   a. Note that putBlockAllocation and putBlockAllocationRange are also available as async operations. A separate topic explains how to use these calls. This is recommended for blocks longer than 30 days.

   b. When the OCCUPANCY PER ROOM TYPE parameter is not active, only submit inventory updates for the onePerson element. When the parameter is active, inventory updates can be done for onePerson, twoPerson, threePerson and FourPerson. Rates can be submitted for all occupancy levels, regardless of the parameter setting.

   c. When the block is not in an open-for-pickup status, the allocation element must carry the value INITIAL. When the block is open for pickup, any inventory updates must be done for allocation = ACTUAL.

   d. If a valid rate code has been submitted as part of the block creation process, rate amounts should not be sent during the grid update process. Rate amounts in OPERA Cloud are populated automatically from the rate code on the block header.

4. Retrieve the newly created block using getBlock with fetchInstructions for the room grid to ensure that the block was created with the correct attributes (optional).

**Sample calls**: Sample API calls for steps described in the diagram.

*Sample for* **getInventoryStatistics**

**Check House Availability**

```
200 response:

[
    {
        "statistics": [
            {
                "statisticDate": [
                    {
                        "inventory": [
                            {
                                "value": 160,
                                "code": "RoomsSold"
                            },
                            {
                                "value": 0,
                                "code": "OverBookingRooms"
                            },
                            {
                                "value": 448,
                                "code": "SellLimitRooms"
                            },
                            {
                                "value": 288,
                                "code": "AvailableRooms"
                            },
                            {
                                "value": 0,
                                "code": "Cancelled"
                            },
                            {
                                "value": 288,
                                "code": "MinAvailableRooms"
                            },
                            {
                                "value": 288,
                                "code": "AvailableRoomsExcludingOverbooking"
                            }
                        ],
                        "statisticDate": "2025-01-01",
                        "weekendDate": false
                    },
                    {
                        "inventory": [
                            {
                                "value": 160,
                                "code": "RoomsSold"
                            },
                            {
                                "value": 0,
                                "code": "OverBookingRooms"
                            },
                            {
                                "value": 448,
                                "code": "SellLimitRooms"
                            },
                            {
                                "value": 288,
                                "code": "AvailableRooms"
                            },
                            {
                                "value": 0,
                                "code": "Cancelled"
                            },
                            {
                                "value": 288,
                                "code": "MinAvailableRooms"
                            },
                            {
                                "value": 288,
                                "code": "AvailableRoomsExcludingOverbooking"
                            }
                        ],
                        "statisticDate": "2025-01-02",
                        "weekendDate": false
                    }
                ],
                "statCode": "BHOTEL",
                "statCategoryCode": "HotelCode"
            },
            {
                "statisticDate": [
                    {
                        "inventory": [
                            {
                                "value": 2,
                                "code": "AvailableRooms"
                            },
                            {
                                "value": 1,
                                "code": "SequenceId"
                            }
                        ],
                        "statisticDate": "2025-01-01",
                        "weekendDate": false
                    },
                    {
                        "inventory": [
                            {
                                "value": 2,
                                "code": "AvailableRooms"
                            },
                            {
                                "value": 1,
                                "code": "SequenceId"
                            }
                        ],
                        "statisticDate": "2025-01-02",
                        "weekendDate": false
                    }
                ],
                "statCode": "A1K",
                "statCategoryCode": "HotelRoomCode",
                "statCodeClass": "STANDARD",
                "description": "A1K to test meetingroom_yn flag (Meeting CB &
Events CB)"
            },
            {
                "statisticDate": [
                    {
                        "inventory": [
                            {
                                "value": 18,
                                "code": "AvailableRooms"
                            },
                            {
                                "value": 1,
                                "code": "SequenceId"
                            }
                        ],
                        "statisticDate": "2025-01-01",
                        "weekendDate": false
                    },
                    {
```

```
                "inventory": [
                    {
                        "value": 18,
                        "code": "AvailableRooms"
                    },
                    {
                        "value": 1,
                        "code": "SequenceId"
                    }
                ],
                "statisticDate": "2025-01-02",
                "weekendDate": false
            }
        ],
        "statCode": "DLXK",
        "statCategoryCode": "HotelRoomCode",
        "statCodeClass": "DELX",
        "description": "Deluxe King Guestroom"
    },
    {
        "statisticDate": [
            {
                "inventory": [
                    {
                        "value": 5,
                        "code": "AvailableRooms"
                    },
                    {
                        "value": 1,
                        "code": "SequenceId"
                    }
                ],
                "statisticDate": "2025-01-01",
                "weekendDate": false
            },
            {
                "inventory": [
                    {
                        "value": 5,
                        "code": "AvailableRooms"
                    },
                    {
                        "value": 1,
                        "code": "SequenceId"
                    }
                ],
                "statisticDate": "2025-01-02",
                "weekendDate": false
            }
        ],
        "statCode": "DLXQ",
        "statCategoryCode": "HotelRoomCode",
        "statCodeClass": "DELX",
        "description": "Deluxe Queen Bedded Guestroom"
    },
    {
```

```
                            "statisticDate": [
                                {
                                    "inventory": [
                                        {
                                            "value": 21,
                                            "code": "AvailableRooms"
                                        },
                                        {
                                            "value": 1,
                                            "code": "SequenceId"
                                        }
                                    ],
                                    "statisticDate": "2025-01-01",
                                    "weekendDate": false
                                },
                                {
                                    "inventory": [
                                        {
                                            "value": 21,
                                            "code": "AvailableRooms"
                                        },
                                        {
                                            "value": 1,
                                            "code": "SequenceId"
                                        }
                                    ],
                                    "statisticDate": "2025-01-02",
                                    "weekendDate": false
                                }
                            ],
                            "statCode": "EXEC",
                            "statCategoryCode": "HotelRoomCode",
                            "statCodeClass": "SUPERIOR",
                            "description": "Executive Suite"
                        },
                        {
                            "statisticDate": [
                                {
                                    "inventory": [
                                        {
                                            "value": 5,
                                            "code": "AvailableRooms"
                                        },
                                        {
                                            "value": 2,
                                            "code": "SequenceId"
                                        }
                                    ],
                                    "statisticDate": "2025-01-01",
                                    "weekendDate": false
                                },
                                {
                                    "inventory": [
                                        {
                                            "value": 5,
                                            "code": "AvailableRooms"
```

```
                }, 
                {
                    "value": 2,
                    "code": "SequenceId"
                }
            ],
            "statisticDate": "2025-01-02",
            "weekendDate": false
        }
    ],
    "statCode": "PRES",
    "statCategoryCode": "HotelRoomCode",
    "statCodeClass": "ALL",
    "description": "Presidential Suite"
},
{
    "statisticDate": [
        {
            "inventory": [
                {
                    "value": 3,
                    "code": "AvailableRooms"
                },
                {
                    "value": 50,
                    "code": "SequenceId"
                }
            ],
            "statisticDate": "2025-01-01",
            "weekendDate": false
        },
        {
            "inventory": [
                {
                    "value": 3,
                    "code": "AvailableRooms"
                },
                {
                    "value": 50,
                    "code": "SequenceId"
                }
            ],
            "statisticDate": "2025-01-02",
            "weekendDate": false
        }
    ],
    "statCode": "CCOTHER",
    "statCategoryCode": "HotelRoomCode",
    "statCodeClass": "ALL",
    "description": "Castle Club Other"
},
{
    "statisticDate": [
        {
            "inventory": [
                {
```

```
                                            "value": 1,
                                            "code": "AvailableRooms"
                                        }
                                    ],
                                    "statisticDate": "2025-01-01",
                                    "weekendDate": false
                                },
                                {
                                    "inventory": [
                                        {
                                            "value": 1,
                                            "code": "AvailableRooms"
                                        }
                                    ],
                                    "statisticDate": "2025-01-02",
                                    "weekendDate": false
                                }
                            ],
                            "statCode": "2965",
                            "statCategoryCode": "HotelRoomCode",
                            "statCodeClass": "DELX",
                            "description": "Cottage Test"
                        },
                        {
                            "statisticDate": [
                                {
                                    "inventory": [
                                        {
                                            "value": 4,
                                            "code": "AvailableRooms"
                                        }
                                    ],
                                    "statisticDate": "2025-01-01",
                                    "weekendDate": false
                                },
                                {
                                    "inventory": [
                                        {
                                            "value": 4,
                                            "code": "AvailableRooms"
                                        }
                                    ],
                                    "statisticDate": "2025-01-02",
                                    "weekendDate": false
                                }
                            ],
                            "statCode": "CLK",
                            "statCategoryCode": "HotelRoomCode",
                            "statCodeClass": "ALL",
                            "description": "Club King Room"
                        },
                        {
                            "statisticDate": [
                                {
                                    "inventory": [
                                        {
                                            "value": 1,
                                            "code": "AvailableRooms"
                                        }
                                    ],
                                    "statisticDate": "2025-01-01",
                                    "weekendDate": false
                                },
                                {
                                    "inventory": [
                                        {
                                            "value": 1,
                                            "code": "AvailableRooms"
                                        }
                                    ],
                                    "statisticDate": "2025-01-02",
                                    "weekendDate": false
                                }
                            ],
                            "statCode": "JS1",
                            "statCategoryCode": "HotelRoomCode",
                            "statCodeClass": "ALL",
                            "description": "Junior Suite"
                        },
                        {
                            "statisticDate": [
                                {
                                    "inventory": [
                                        {
                                            "value": 1,
                                            "code": "AvailableRooms"
                                        }
                                    ],
                                    "statisticDate": "2025-01-01",
                                    "weekendDate": false
                                },
                                {
                                    "inventory": [
                                        {
                                            "value": 1,
                                            "code": "AvailableRooms"
                                        }
                                    ],
                                    "statisticDate": "2025-01-02",
                                    "weekendDate": false
                                }
                            ],
                            "statCode": "SK1009",
                            "statCategoryCode": "HotelRoomCode",
                            "statCodeClass": "STANDARD",
                            "description": "SK Room type -1009"
                        },
                        {
                            "statisticDate": [
                                {
                                    "inventory": [
                                        {
```

```
                                "value": 114,
                                "code": "AvailableRooms"
                            }
                        ],
                        "statisticDate": "2025-01-01",
                        "weekendDate": false
                    },
                    {
                        "inventory": [
                            {
                                "value": 114,
                                "code": "AvailableRooms"
                            }
                        ],
                        "statisticDate": "2025-01-02",
                        "weekendDate": false
                    }
                ],
                "statCode": "STDK",
                "statCategoryCode": "HotelRoomCode",
                "statCodeClass": "ALL",
                "description": "Standard King Guestroom"
            },
            {
                "statisticDate": [
                    {
                        "inventory": [
                            {
                                "value": 118,
                                "code": "AvailableRooms"
                            }
                        ],
                        "statisticDate": "2025-01-01",
                        "weekendDate": false
                    },
                    {
                        "inventory": [
                            {
                                "value": 118,
                                "code": "AvailableRooms"
                            }
                        ],
                        "statisticDate": "2025-01-02",
                        "weekendDate": false
                    }
                ],
                "statCode": "STDQ",
                "statCategoryCode": "HotelRoomCode",
                "statCodeClass": "ALL",
                "description": "Standard Double Queen Bedded Room"
            }
        ],
        "hotelName": "BHOTEL",
        "reportCode": "RoomsAvailabilitySummary"
    }
]
```

*Sample for* **postBlock**

**Create Block Header**

POST**{{HostName}}**/blk/v1/hotels/**{{HotelId}}**/block

```
Body:

{
    "blocks": {
        "blockInfo": [
            {
                "block": {
                       "blockDetails": {
                        "blockCode": "REDS1224",
                        "blockName": "Red Ski  Conference XXXX",
                        "blockAliases": [],
                        "timeSpan": {
                            "startDate": "2023-12-01",
                            "endDate": "2023-12-03"
                        },
                        "shoulderDates": {},
                        "blockStatus": {
                            "bookingStatus": {
                                "status": {
                                    "code": "INQ"
                                }
                            }
                        },
                        "reservationType": {
                            "reservationType": "GNON"
                        },
                        "marketCode": {
                            "marketCode": "CORI"
                        },
                        "sourceOfSale": {
                            "sourceCode": {
                                "sourceCode": "GSALES"
                            }
                        },
                        "reservationMethod": {},
                        "bookingType": {},
                        "rateProtectionDetails": {
                            "criteria": "None"
                        },
                        "nonCompeteDetails": {
                            "criteria": "None"
                        },
                        "currencyCode": "USD",
                        "blockClassification": "RegularBooking",
                        "cateringOnlyBlock": false,
                        "allowRateOverride": false,
                        "manualCutOff": false,
                        "wholesaleBlock": false,
                        "controlBlockLocally": false
                    },
                    "externalReferences": [
                        {
                            "id": "12345",
                            "idExtension": "12345",
                            "idContext": "XXXXXXX"
                        },
                         {
                            "id": "53455",
                            "idExtension": "45345",
                            "idContext": "YYYYYYY"
                        },
                         {
                            "id": "123645",
                            "idExtension": "456666",
                            "idContext": "ZZZZZZZ"
                        }
                    ],
                    "blockOwners": {
                        "owner": [
                            {
                                "ownership": "Block",
                                "ownerCode": "ALL",
                                "primary": true
                            },
                            {
                                "ownership": "Catering",
                                "ownerCode": "ALL",
                                "primary": true
                            },
                            {
                                "ownership": "Rooms",
                                "ownerCode": "ALL",
                                "primary": true
                            }
                        ],
                        "lockBlockOwners": false,
                        "lockRoomsOwners": false,
                        "lockCateringOwners": false
                    },
                    "reservationDetails": {
                        "ratePlanCode": [
                            {
                                "primary": true,
                                "ratePlanCode": "BAR"
                            }
                        ],
                        "traceCode": {},
                        "breakfast": {
                            "breakfastIncluded": false,
                            "price": {
                                "currencyCode": "USD"
                            }
                        },
                        "porterage": {
                            "porterageIncluded": false,
                            "price": {
                                "currencyCode": "USD"
```

```
                            }
                        },
                        "elastic": "1",
                        "suppressRate": false,
                        "housing": true
                    },
                    "catering": {
                        "cateringStatus": {
                            "bookingStatus": {
                                "status": {}
                            }
                        },
                        "eventAttendees": {},
                        "overrideEventsProcessingWarnings": true
                    },
                    "blockProfiles": {
                        "fullOverlay": false
                    },
                    "blockPackages": [],
                    "externalAttributes": {
                        "eventType": "Convention",
                        "rollEndDate": false
                    },
                    "hotelId": "{{HotelId}}",
                    "markAsRecentlyAccessed": true
                }
            }
        ]
    }
}
```

**201 Response: Created**

*Sample for* **putBlockAllocation**

### Note

Although there is a start and end date for each grid cell, in this call, the start and end date in each collection must be the same (and the collection must be repeated for any additional dates).

This call is for a block that is not open for pickup. Please note the element: "allocation": "INITIAL". When running this RQ for an open-for-pickup call, the element must be :"allocation": "ACTUAL."

**Create / Update Room Grid** (Cell by Cell)

PUT**{{HostName}}**/blk/v0/hotels/**{{HotelId}}**/blocks/**{{BlockId}}**/allocation

```
Body:

{
    "criteria": {
        "hotelId": "{{HotelId}}",
        "blockId": {
            "type": "Block",
            "idContext": "OPERA",
            "id": "{{BlockId}}"
        },
        "allocationRoomTypes": [
            {
                "allocationGridDates": [
                    {
                        "roomAllocationInfo": [
                            {
                                "inventory": {
                                    "forceOverbook": false
                                },
                                "rate": {
                                    "onePerson": "240",
                                    "twoPerson": "240",
                                    "threePerson": "260"
                                },
                                "start": "2023-12-01",
                                "end": "2023-12-01"
                            },
                            {
                                "inventory": {
                                    "forceOverbook": false
                                },
                                "rate": {
                                    "onePerson": "200",
                                    "twoPerson": "220"
                                },
                                "start": "2023-12-02",
                                "end": "2023-12-02"
                            }
                        ],
                        "allocation": "RATES"
                    },
                    {
                        "roomAllocationInfo": [
                            {
                                "inventory": {
                                    "forceOverbook": false,
                                    "onePerson": "4",
                                    "twoPerson": "4"
                                },
                                "rate": {},
                                "start": "2023-12-01",
                                "end": "2023-12-01"
                            },
                            {
                                "inventory": {
                                    "forceOverbook": false,
                                    "onePerson": "3",
                                    "twoPerson": "3"
                                },
                                "rate": {},
                                "start": "2023-12-02",
                                "end": "2023-12-02"
                            }
                        ],
                        "allocation": "INITIAL"
                    }
                ],
                "roomType": "EXEC"
            }
        ],
        "genericRoomType": false
    }
}
```

**200 Response:** OK

*Sample for* **putBlockAllocationRange**

> **Note**
>
> The start and end date in the collection define the date range for which the given values should be populated. Only one range can be defined per call.
>
> This call is for a block that is open for pickup. Please note the element: "allocation": "Actual". When running this RQ for a not open-for-pickup call, the elements must be :"allocation": "Initial."

**Create / Update Room Grid** (Range)

PUT**{{HostName}}**/blk/v1/hotels/**{{HotelId}}**/blocks/**{{BlockId}}**/allocationRange

```
Body:

{
    "blockAllocationRange": {
        "blockId": {
            "type": "Block",
            "idContext": "OPERA",
            "id": "{{BlockId}}"
        },
        "hotelId": "{{HotelId}} ",
        "roomTypes": [
            "DLXQ"
        ],
        "beginDate": "2023-12-01",
        "endDate": "2023-12-02",
        "allocationType": "Actual",
        "incrementFlag": false,
        "blockInventory": {
            "onePerson": "8",
            "twoPerson": "3"
        },
        "blockRates": {
            "onePerson": "240",
            "twoPerson": "280"
        },
        "includedDays": "1111111",
        "rangeMode": "Core",
        "genericRoomType": false
    }
}
```

**200 Response:** OK

*Sample for* **getBlock**

**Fetch Block with Grid Details**

GET**{{HostName}}**/blk/v1/hotels/{{HotelId}}/blocks/{{BlockId}}?fetchAllocatedRoomTypes=true&markAsRecentlyAccessed=false&roomTypes=DLXQ&roomTypes=STDK&roomTypes=STDQ&roomAllocationCriteria=Initial&fetchInstructions=RateGrid&numberOfDays=2&startDate=2024-09-06&roomAllocationCriteria=CutoffDate

**200 Response:** OK

```
{
    "blocks": {
        "blockInfo": [
            {
                "block": {
                    "blockIdList": [
                        {
                            "id": "{{BLockId}}",
                            "type": "Block",
                            "idContext": "OPERA"
                        }
                    ],
                    "roomAllocations": {
                        "roomAllocationType": [
                            {
                                "roomAllocationInfo": [
                                    {
                                        "roomGridInvSummary": [
                                            {}
                                        ],
                                        "availableRoomGridInvSummary": [
                                            {}
                                        ],
                                        "stayDate": "2024-09-06",
                                        "extendedStay": false
                                    },
                                    {
                                        "roomGridInvSummary": [
                                            {}
                                        ],
                                        "availableRoomGridInvSummary": [
                                            {}
                                        ],
                                        "stayDate": "2022-05-19",
                                        "extendedStay": false
                                    },
                                    {
```

```
                                                        "roomGridInvSummary": [
                                                            {}
                                                        ],
                                                        "availableRoomGridInvSummary": [
                                                            {}
                                                        ],
                                                        "stayDate": "2022-05-20",
                                                        "extendedStay": false
                                                    },
                                                    {
                                                        "roomGridInvSummary": [
                                                            {}
                                                        ],
                                                        "availableRoomGridInvSummary": [
                                                            {}
                                                        ],
                                                        "stayDate": "2022-05-21",
                                                        "extendedStay": false
                                                    },
                                                    {
                                                        "roomGridInvSummary": [
                                                            {}
                                                        ],
                                                        "availableRoomGridInvSummary": [
                                                            {}
                                                        ],
                                                        "stayDate": "2022-05-22",
                                                        "extendedStay": false
                                                    },
                                                    {
                                                        "roomGridInvSummary": [
                                                            {}
                                                        ],
                                                        "availableRoomGridInvSummary": [
                                                            {}
                                                        ],
                                                        "stayDate": "2022-05-23",
                                                        "extendedStay": false
                                                    },
                                                    {
                                                        "roomGridInvSummary": [
                                                            {}
                                                        ],
                                                        "availableRoomGridInvSummary": [
                                                            {}
                                                        ],
                                                        "stayDate": "2022-05-24",
                                                        "extendedStay": false
                                                    }
                                                ],
                                                "allocation": "Initial"
                                            },
                                            {
                                                "roomAllocationInfo": [
                                                    {
                                                        "roomGridInvSummary": [
        {
            "onePerson": 0,
            "twoPerson": 0,
            "threePerson": 0,
            "fourPerson": 0
        }
    ],
    "availableRoomGridInvSummary": [
        {}
    ],
    "stayDate": "2022-05-18",
    "extendedStay": false
},
{
    "roomGridInvSummary": [
        {
            "onePerson": 0,
            "twoPerson": 0,
            "threePerson": 0,
            "fourPerson": 0
        }
    ],
    "availableRoomGridInvSummary": [
        {}
    ],
    "stayDate": "2022-05-19",
    "extendedStay": false
},
{
    "roomGridInvSummary": [
        {
            "onePerson": 0,
            "twoPerson": 0,
            "threePerson": 0,
            "fourPerson": 0
        }
    ],
    "availableRoomGridInvSummary": [
        {}
    ],
    "stayDate": "2022-05-20",
    "extendedStay": false
},
{
    "roomGridInvSummary": [
        {
            "onePerson": 0,
            "twoPerson": 0,
            "threePerson": 0,
            "fourPerson": 0
        }
    ],
    "availableRoomGridInvSummary": [
        {}
    ],
    "stayDate": "2022-05-21",
                "extendedStay": false
            },
            {
                "roomGridInvSummary": [
                    {
                        "onePerson": 0,
                        "twoPerson": 0,
                        "threePerson": 0,
                        "fourPerson": 0
                    }
                ],
                "availableRoomGridInvSummary": [
                    {}
                ],
                "stayDate": "2022-05-22",
                "extendedStay": false
            },
            {
                "roomGridInvSummary": [
                    {
                        "onePerson": 0,
                        "twoPerson": 0,
                        "threePerson": 0,
                        "fourPerson": 0
                    }
                ],
                "availableRoomGridInvSummary": [
                    {}
                ],
                "stayDate": "2022-05-23",
                "extendedStay": false
            },
            {
                "roomGridInvSummary": [
                    {
                        "onePerson": 0,
                        "twoPerson": 0,
                        "threePerson": 0,
                        "fourPerson": 0
                    }
                ],
                "availableRoomGridInvSummary": [
                    {}
                ],
                "stayDate": "2022-05-24",
                "extendedStay": false
            }
        ],
        "allocation": "CutoffDate"
    }
],
"masterInfo": {
    "roomTypes": []
},
"startDate": "2022-05-18",
"numberOfDays": 7
```

```
                    },
                    "hotelId": "BHOTEL",
                    "markAsRecentlyAccessed": false
                }
            }
        ]
    },
    "links": []
}
```

### References

- For more information, review the Anti-Patterns chapter in the OHIP user guide to learn more about functional and technical anti-patterns.
- Fore more information, refer to the Blocks introduction topic in the OPERA Cloud Services user guide.

3

# Create Block Posting Master

### Business context

Posting Master reservations linked to a block are used for two purposes:

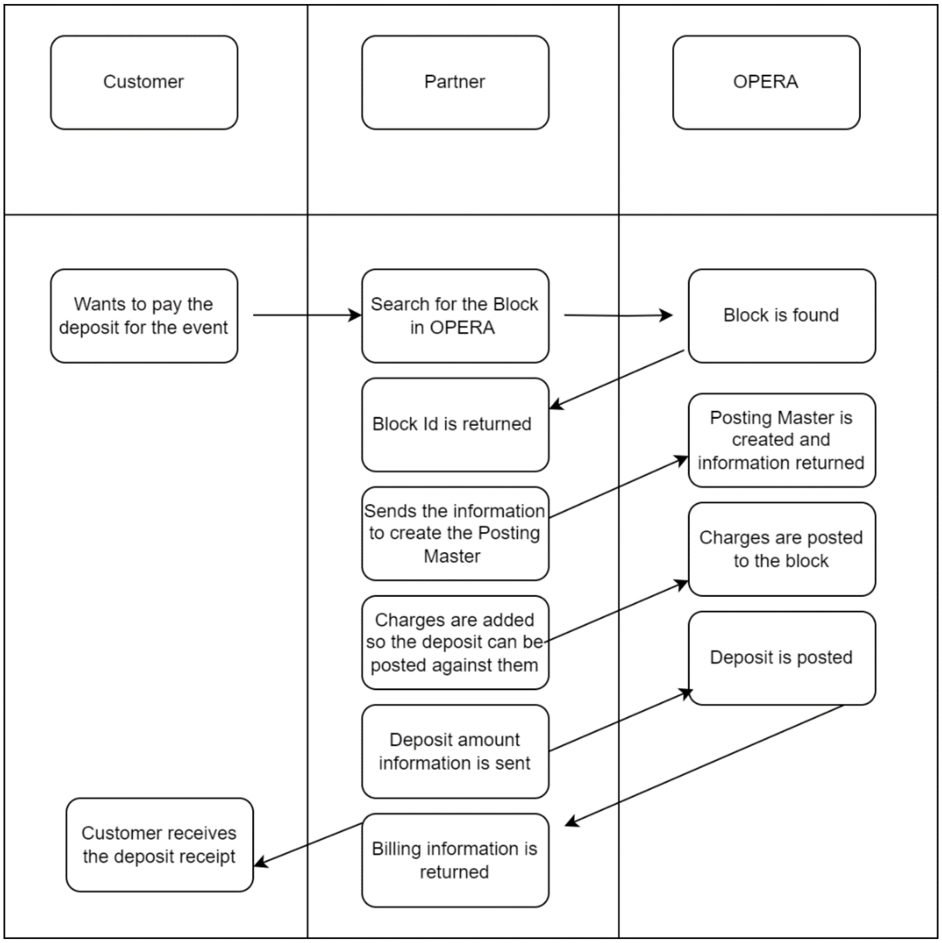
1. Billing — Where various sales charges must be routed or transferred from individual block room reservation accounts to a 'Block master account' for settlement.
2. Rooming List — Where the selected Posting Master reservation is used as a template for reservations created through a rooming list. Reservations created inherit their attributes from the selected post master reservation. You can create multiple PM templates reservations according to your needs.

Within the OPERA UI, the Posting Master reservation is created automatically when either the block status is changed, when the deposit / cancellation screen is accessed, or when the rooming list operation is accessed. The Posting Master creation does not happen automatically when these changes are made through the REST APIs; therefore, it is important to check if a Posting Master already exists before creating any new ones using the REST APIs by posting deposits or any other operations that involve creating Posting Masters.

### Workflow Diagram

The below workflow diagram explains typical use cases for adding a menu to an event.

**Figure 3-1 Use Cases for adding a menu to an event**


Customer
Partner
OPERA
Wants to pay the deposit for the event
Search for the Block in OPERA
Block is found
Block Id is returned
Posting Master is created and information returned
Sends the information to create the Posting Master
Charges are posted to the block
Charges are added so the deposit can be posted against them
Deposit is posted
Deposit amount information is sent
Customer receives the deposit receipt
Billing information is returned

Customer — The meeting planner is ready to pay the deposit for the upcoming event.

Partner — The partner communicates to the property the amount the customer wants to pay as a deposit for the event, so the billing information must be ready.

OPERA Cloud — The application creates the Posting Master with the correct charges to apply as a deposit against it.

### Prerequisites

**Table 3-1 Prerequisites**

| Tool | Description | Links |
|---|---|---|
| Postman | Postman is an API Platform that allows you to design, create and test API's.<br>Use Postman to send API requests or to use the Postman collections provided. | Using the Oracle Hospitality APIs<br>Tool |

**Table 3-2 Current Versions Required**

| OPERA Cloud Platform Module | Description | Minimum Version |
|---|---|---|
| OPERA Cloud Services | The customer must have a subscription to OPERA Cloud Foundation | |
| Oracle Hospitality Integration Platform (OHIP) | For custopmers OHIP is included in the subscription to OPERA Cloud Foundation.<br>Partners need a subscription to Oracle Hospitality Integration Cloud Service | |

### Configuration of OPERA Controls

To activate the Create Block functionality, you must ensure the following OPERA Controls are active in the target property. You can use the following API operations to validate the settings:

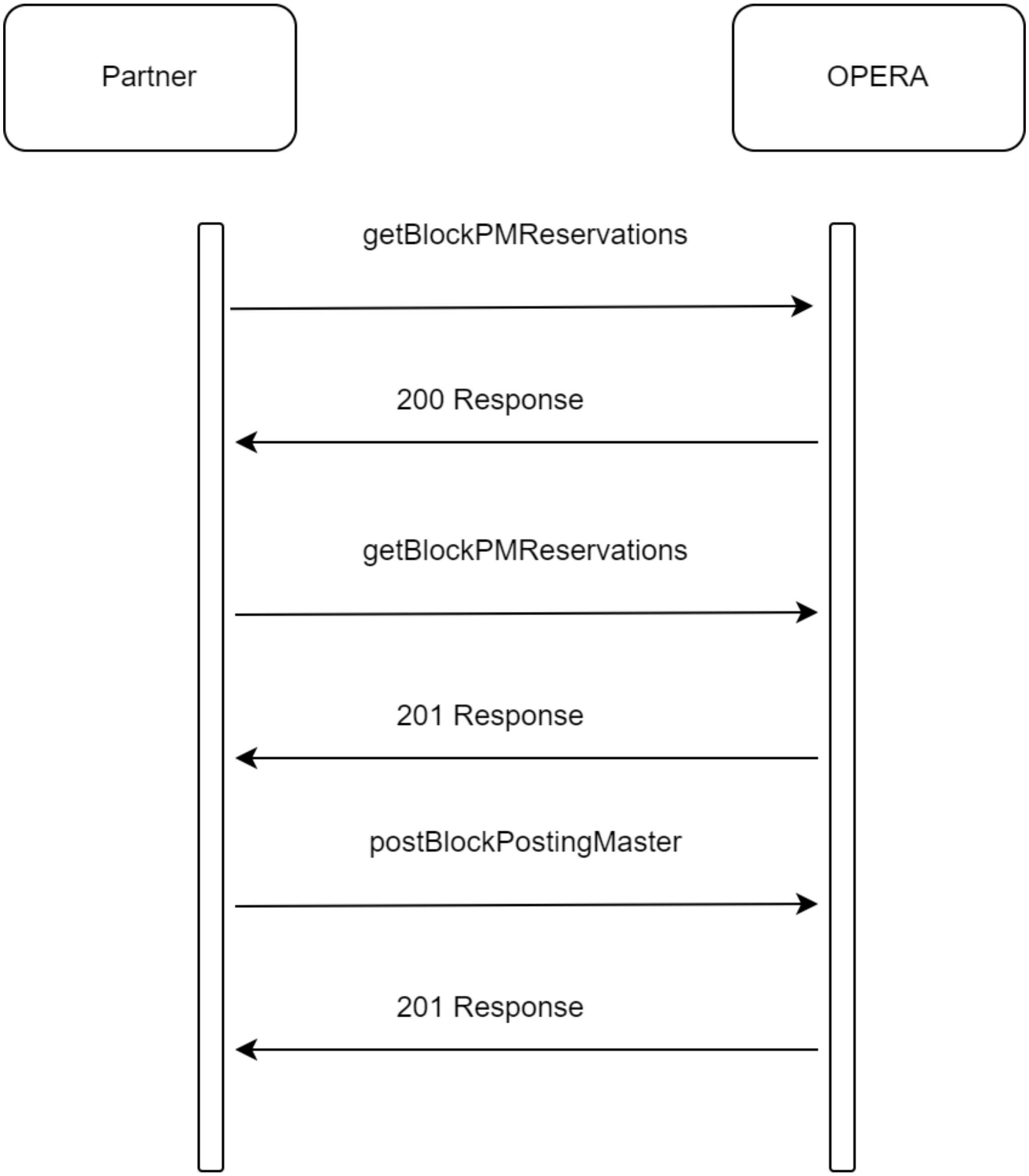
**Ensure that application function BLOCKS - BUSINESS BLOCK is active**

RQ: ent/config/v1/settings?hotelId={{HotelId}}&parameterNameWildCard=BUSINESS BLOCKblk/config/v1/blockStatusCodes

### Workflow

Step by step solution including:

**Figure 3-2 The sequence of API calls (in any direction)**



### Description of Steps

A typical process to create a block posting master is as follows:

1. Check if the block in question already has one or several posting master reservations using **getBlockPMReservations.**
2. If existing Posting Master Reservations are returned, use one of these as applicable.
3. If no Posting Master Reservations exist, you can create one using the **postBlockPostingMaster** operation.
   a. All reservation details for the PM are defaulted from the Block.

4. Check for the Posting Master using **getBlockPMReservations** (optional).

**Sample calls**: Sample API calls for steps described in the diagram.

Sample for **getBlockPMReservations**

**Get List of existing Posting Masters for the Block**

GET**{{HostName}}**/blk/v1/hotels/{{HotelId}}/blocks/{{BlockId}}/postingMaster/reservations

**No Body**

**200 Response: OK**

**Body:**

```
{
    "reservations": {
        "reservationInfo": [
            {
                "reservationIdList": [
                    {
                        "id": "1264363",
                        "type": "Reservation"
                    },
                    {
                        "id": "2042429",
                        "type": "Confirmation"
                    }
                ],
                "roomStay": {
                    "arrivalDate": "2024-07-07",
                    "departureDate": "2024-07-08",
                    "originalTimeSpan": {
                        "startDate": "2024-07-07",
                        "endDate": "2024-07-08"
                    },
                    "expectedTimes": {
                        "reservationExpectedDepartureTime": "2024-07-08"
                    },
                    "adultCount": 0,
                    "childCount": 0,
                    "roomClass": "ALL",
                    "roomType": "PM",
                    "numberOfRooms": 1,
                    "roomOwnershipType": "Regular",
                    "rateAmount": {
                        "amount": 0,
                        "currencyCode": "USD"
                    },
                    "rateSuppressed": false,
                    "reservationBlock": {
                        "blockIdList": [
                            {
                                "id": "918694",
                                "type": "Block"
                            },
                            {
```

```
                "id": "240707CREA",
                "type": "BlockCode"
            }
        ]
    },
    "fixedRate": false,
    "guarantee": {
        "guaranteeCode": "GNON",
        "shortDescription": "Group Non Deduct"
    },
    "marketCode": "CORI",
    "sourceCode": "GSALES",
    "sourceCodeDescription": "Group Sales",
    "balance": {
        "amount": 0,
        "currencyCode": "USD"
    },
    "compBalance": {
        "amount": 0,
        "currencyCode": "USD"
    },
    "roomTypeCharged": "PM",
    "roomNumberLocked": false,
    "pseudoRoom": true
},
"departureReservationInfo": {},
"reservationGuest": {
    "surname": "Create Posting Master",
    "language": "E",
    "vip": {},
    "address": {
        "streetAddress": "",
        "country": {
            "code": "US"
        }
    },
    "anonymization": {},
    "accompanyGuests": [],
    "guestRestricted": false,
    "nameType": "Group",
    "id": "1696531",
    "type": "Profile"
},
"sharedGuests": [],
"attachedProfiles": [
    {
        "name": "Create Posting Master",
        "profileIdList": [
            {
                "id": "1696531",
                "type": "Profile"
            }
        ],
        "reservationProfileType": "Group"
    }
],
                "reservationPaymentMethod": {},
                "reservationFolioWindows": [],
                "commissionsInfo": {},
                "reservationIndicators": [
                    {
                        "indicatorName": "NOPOST"
                    },
                    {
                        "indicatorName": "PACKAGEITEM",
                        "count": 2
                    },
                    {
                        "indicatorName": "HISTORYFUTURE",
                        "count": 1
                    },
                    {
                        "indicatorName": "COMMUNICATION",
                        "count": 1
                    },
                    {
                        "indicatorName": "ASSOCIATEDPROFILES",
                        "count": 2
                    },
                    {
                        "indicatorName": "DAILYDETAILS",
                        "count": 0
                    }
                ],
                "sourceOfSale": {
                    "sourceType": "PMS",
                    "sourceCode": "BHOTEL"
                },
                "waitlist": {},
                "advanceCheckIn": {
                    "advanceCheckedIn": false
                },
                "hotelId": "BHOTEL",
                "hotelName": "Baltimore Inner & Harbour East Hotel",
                "roomStayReservation": true,
                "createDateTime": "2024-02-21 10:49:30.0",
                "lastModifyDateTime": "2024-02-21 10:49:30.0",
                "reservationStatus": "Requested",
                "computedReservationStatus": "Requested",
                "walkInIndicator": false,
                "commissionPayoutTo": "None",
                "preRegistered": false,
                "openFolio": false,
                "allowMobileCheckout": false,
                "optedForCommunication": false
            }
        ],
        "totalPages": 1,
        "offset": 20,
        "limit": 20,
        "hasMore": false,
        "totalResults": 1
```

```
    },
    "links": [
        {
            "href": "https://XYZ/rsv/v1/hotels/BHOTEL/reservations",
            "rel": "other",
            "templated": false,
            "method": "POST",
            "operationId": "postReservation"
        },
        {
            "href": "https://XYZ/rsv/v1/hotels/BHOTEL/reservations/1264363",
            "rel": "self",
            "templated": false,
            "method": "GET",
            "operationId": "getReservation"
        },
        {
            "href": "https://XYZ/rsv/v1/hotels/BHOTEL/reservations/1264363",
            "rel": "self",
            "templated": false,
            "method": "PUT",
            "operationId": "putReservation"
        }
    ]
}
```

*Sample for* **postBlockPostingMaster**

**Post Block Posting Master**

POST**{{HostName}}**/blk/v1/hotels/**{{HotelId}}**/blocks/**{{BlockId}}**/status

```
{
    "criteria": {
        "hotelId": "{{HotelId}}",
        "blockId": {
            "type": "Block",
            "idContext": "OPERA",
            "id": "{{BlockId}}"
        },
        "responseInstruction": {
            "fetchFullReservation": false
        }
    }
}
```

**201 Response:** Created

Posting Master ID is returned in Location parameter in RS Headers.

### References

For more information, review the Best Practices in the OHIP user guide to learn more about functional and technical anti-patterns.

For more information, refer to the Creating Additional Posting Master Reservation in the OPERA Cloud Services user guide.

4

# Create Event with Function Space

**Business Context**

The Events feature is designed to manage any kind of catering activity. Events can be as simple as a one-hour reception or more complex, such as a three-day business meeting with meals, breaks, and specific meeting rooms with setup and resource requirements. Any group function can be an event.

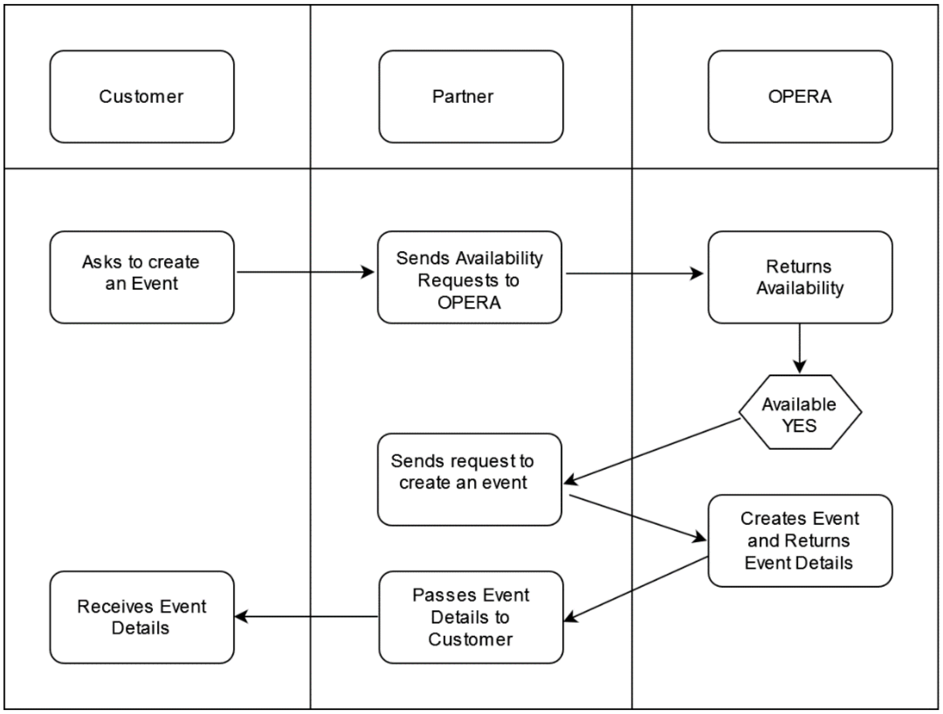
The below functional diagram explains the typical use cases for creating a new event.

Customer — The meeting manager is planning an off-site meeting for a group of VIP executives.

Partner — The partner is checking with the chosen property about meeting space availability for the selected dates.

OPERA Cloud — The application checks and returns availability request and response, creates the event, adds the details, and returns the final event detailed.

**Figure 4-1    Typical use cases for creating a new event**


Customer
Partner
OPERA
Asks to create an Event
Sends Availability Requests to OPERA
Returns Availability
Available YES
Sends request to create an event
Creates Event and Returns Event Details
Receives Event Details
Passes Event Details to Customer

### Prerequisites

**Table 4-1    Required Software Tools**

| Tool | Description | Links |
|---|---|---|
| Postman | Postman is an API Platform that allows you to design, create and test API's.<br>Use Postman to send API requests or to use the Postman collections provided. | Using the Oracle Hospitality APIs<br>Tool |

**Table 4-2    Current Versions Required**

| OPERA Cloud Platform Module | Description | Minimum Version |
|---|---|---|
| OPERA Cloud Services | The customer must have a subscription to OPERA Cloud Foundation | |
| OPERA Cloud Sales and Event Management | The customer must have a subscription to OPERA Cloud Sales and Event Management | |
| OPERA Cloud Sales and Event Management | For customers OHIP is included in the subscription to OPERA Cloud Foundation.<br>Partners need a subscription to Oracle Hospitality Integration Cloud Service | |

### Configuration of OPERA Controls

To activate the Create Block functionality, you need to ensure the following OPERA Controls are active in the target property. You can use the following API operations to validate the settings:

**Ensure that application function EVENTS - CATERING EVENTS is active**

RQ: ent/config/v1/settings?hotelId={{HotelId}}&parameterNameWildCard=CATERING EVENTS

Mandatory Fields - Available Values:

**Event Type**

RQ: evm/config/v1/cateringEventTypes?includeInactive=false&hotelIds={{hotelId}}

**Event Status Code**

RQ: blk/config/v1/blockStatusCodes

Dependent Field - Available Values:

**Function Space Configuration**

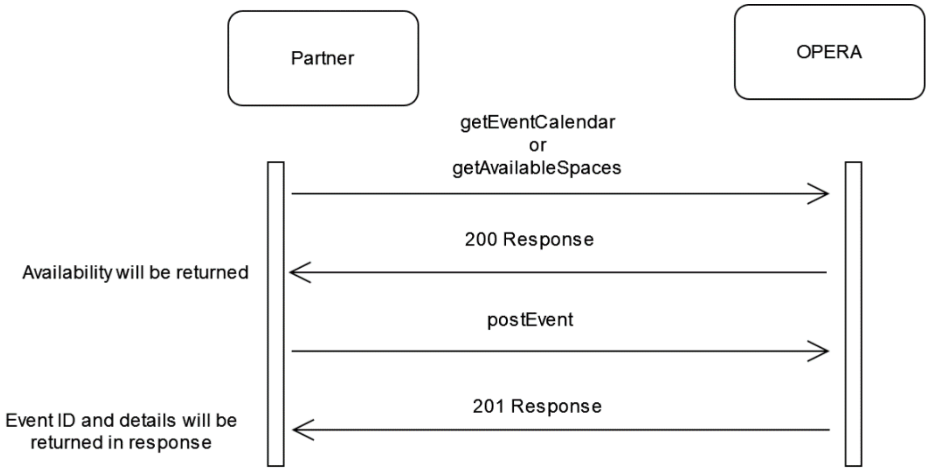
> **Note**
>
> This will return the codes available for Rental Code and Setup Styles which are required fields when creating an event with a function space.

RQ: evm/config/v1/hotels/{{HotelId}}/functionSpaces/{{FunctionSpaceCode}}/configurations

### Workflow

Step by step solution including:

**Figure 4-2 The sequence of API calls (in any direction)**



**Description of Steps**:

A typical process to create a new event with a function space is as follows:

1. Search for function space availability for the event date(s) using **getEventCalendar** or **getAvailableSpaces**. The getEventCalendar request returns function spaces available for a specific date while the getAvailableSpaces request provides function spaces available for the specified date and time period. If you are creating an event without function space, then omit this step
   a. If you use getAvailableSpaces, the default setup style and default rental code details are returned in the response for each space.
2. Search for the blocks by date and name. If you know the internal block ID, then omit this step. Ensure the block has an active Catering Status and Catering Attendees is populated.
   a. To search for a block, you can use the REST API call. {{HostDomain}}/operarest/{schemamap}/blk/v1/blocks?onlyOpportunities=false&offset=1&blockName={{BlockId}}&tourSeries=false&markAsRecentlyAccessed=false&leads=false&limit=20&hotelId={{HotelId}}&onlyPickupBlocks=false
3. Create the event using **postEvent**.
   a. Ensure that mandatory fields Event Type, Event Status Code, Start Time, End Time, and Expected Attendees are sent with valid values.
   b. If you are intending to assign a function space, ensure that the elements for Rental Code and Setup Styles are present and contain valid values.

**Sample calls**: Sample API calls for steps described in the diagram.

Sample for**geteventCalendar**

**Check Function Space Availability**

**getEventCalendar**

GET **{{HostName}}**/evm/v1/hotels/**{{HotelId}}**/eventcalendar?eventEndDate=2025-04-21&isSortDescending=false&showAvailableRooms=true&display=ALL&eventLabel=Booking&sortingBy=Name&eventStartDate=2025-04-21

**200 response:**

```
{
    "eventCalendarDetails": {
        "eventSpace": [
            {
                "description": "Anna",
                "eventSpaceAttributes": {
                    "spaceArea": {
                        "area": 960,
                        "unit": "SqFeet"
                    },
                    "shareable": false,
                    "comboSpace": false,
                    "alternate": false,
                    "maximumOccupancy": 100,
                    "comboElement": true,
                    "comboRoom": "AFCR"
                },
                "hotelId": "BHOTEL",
                "roomId": "ANNA",
                "name": "ANNA",
                "building": "MAIN",
                "roomOrder": 1
            }
        ],
        "eventCodes": [],
        "sellMessages": {},
        "totalRooms": 1
    },
    "links": [],
    "warnings": []
}
```

Sample for **getavailablespaceslov**

**Check Function Space Availability**

**getAvailableSpaces**

**GET {{HostName}}**/lov/v1/listOfValues/AvailableSpaces?parameterName=Event_Start&parameterName=Event_End&parameterName=HotelCode&includeInactiveFlag=false&parameterValue=2024-04-21+09:00:00.0&parameterValue=2025-04-21+17:00:00.0&parameterValue=**{{HotelId}}**

**200 response:**

```
{
    "listOfValues": {
        "items": [
            {
                "flexfields": [
                    {
                        "parameterName": "AvailableYn",
                        "parameterValue": "Y"
                    },
                    {
                        "parameterName": "SetupCode",
                        "parameterValue": "THE"
                    },
                    {
                        "parameterName": "OccMin",
                        "parameterValue": "10"
                    },
                    {
                        "parameterName": "OccMax",
                        "parameterValue": "100"
                    },
                    {
                        "parameterName": "SetupTime"
                    },
                    {
                        "parameterName": "SetdownTime"
                    },
                    {
                        "parameterName": "RateCode",
                        "parameterValue": "HD"
                    },
                    {
                        "parameterName": "Rate",
                        "parameterValue": "12"
                    },
                    {
                        "parameterName": "ShareableYn",
                        "parameterValue": "N"
                    },
                    {
                        "parameterName": "ForceAlternateYn",
                        "parameterValue": "N"
                    }
                ],
                "code": "001",
                "name": "Argus Conference Room - 001",
                "description": "Argus Conference Room - 001",
                "active": true
            },
```

Sample for **postEvent**

**Create Event with Space**

POST**{{HostName}}**/evm/v1/events

Body:

```
{
    "eventDetails": [
        {
            "eventPrimaryInfo": {
```

```
            "eventId": {
                "type": "EventId",
                "idContext": "OPERA",
                "id": "-1"
            },
            "hotelId": "BHOTEL"
        },
        "eventDetail": {
            "eventName": {
                "defaultText": "Meeting",
                "translations": [],
                "translatedTexts": []
            },
            "eventType": "MTG",
            "eventTimeSpan": {
                "startDateTime": "2025-04-21 09:00:00.0",
                "endDateTime": "2025-04-21 17:00:00.0"
            },
            "eventStatus": {
                "status": {
                    "code": "INQ"
                }
            },
            "attendees": {
                "expected": "30"
            },
            "doorcard": "Do Not Update KD API Block",
            "notMoveable": false,
            "displayDoorcard": true
        },
        "processInstructions": {
            "overrideCapacity": false,
            "overbook": false,
            "overrideMaxgroups": false,
            "warnFunctionSpace": true
        },
        "eventBlockInfo": {
            "blockIdList": [
                {
                    "type": "Block",
                    "idContext": "OPERA",
                    "id": "919698"
                }
            ],
            "hotelId": "BHOTEL",
            "markAsRecentlyAccessed": false
        },
        "functionSpaceInformation": {
            "functionSpaceCode": "BALT",
            "functionSpace": "Baltimore",
            "setupCode": "CLR",
            "setupTime": "30",
            "setdownTime": "30",
            "rateCode": "CUSTOM",
            "rentalCode": "CUSTOM",
            "rentAmount": {
                    "amount": "500"
                },
                "rentalAmount": {
                    "amount": "500"
                }
            }
        }
    ]
}
```

**201 Response**: Created

```
{
    "eventDetails": [
        {
            "eventPrimaryInfo": {
                "eventId": {
                    "type": "EventId",
                    "idContext": "OPERA",
                    "id": "642344"
                },
                "hotelId": "BHOTEL"
            },
            "eventDetail": {
                "eventName": {
                    "defaultText": "Meeting",
                    "translations": []
                },
                "eventType": "MTG",
                "eventTimeSpan": {
                    "startDateTime": "2025-04-21 09:00:00.0",
                    "endDateTime": "2025-04-21 17:00:00.0"
                },
                "eventStatus": {
                    "status": {
                        "code": "INQ"
                    }
                },
                "attendees": {
                    "expected": 30
                },
                "doorcard": "Do Not Update KD API Block",
                "masterEvent": false,
                "subEvent": false,
                "packageEvent": false,
                "alternateEvent": false,
                "notMoveable": false,
                "loudEvent": false,
                "hasPostings": false,
                "displayDoorcard": true,
                "cateringDeductInventory": false,
                "eventShared": false,
                "includeSpaceInPackage": false,
                "eventLevelRevenueActualization": false
            },
            "functionSpaceInformation": {
                "functionSpaceCode": "BALT",
                "setupCode": "CLR",
                "setupTime": 30,
                "setdownTime": 30,
                "rentalCode": "CUSTOM",
                "rentalAmount": {
                    "amount": 500
                }
            }
        }
    ]
}
```

### References

- For more information, review the Anti-Patterns chapter in the OHIP user guide to learn more about functional and technical anti-patterns.
- For more information, refer to the Events topic in the OPERA Cloud Services user guide.

5

# Change Block Status

## Business Case

All groups or blocks have a status that is attached to the block header. This status code reflects the group’s status in terms of inquiry, contract sent, contract pending, confirmed, open for pickup, canceled, and so on. With each status, different options can be included, and the next available status or a status cycle can be set.

During the life of a block, you might need to change block status several times. Each status change affects the inventory differently; for example, from INQUIRY to DEDUCT, from NON DEDUCT to DEDUCT, from DEDUCT to ACTUAL, or from any status to CANCEL. In most scenarios, a status change is also necessary to change the block to an open-for-pickup status (reservations can be made against the block) to close-for-pick-up (no reservations can be made against the block). Status configuration and status flow can be configured in OPERA, and it is necessary to understand the configured status and the available status sequence for your customer site to submit the correct status update for an existing block.

## Prerequisites

The following software tools are required to follow the steps listed below:

**Table 5-1 Required Software Tools**

| Tool | Description | Links |
|---|---|---|
| Postman | Postman is an API Platform that allows you to design, create and test API's.<br>Use Postman to send API requests or to use the Postman collections provided. | Using the Oracle Hospitality APIs<br>Tool |

**Table 5-2 Current Versions Required**

| OPERA Cloud Platform Module | Description | Minimum Version |
|---|---|---|
| OPERA Cloud Services | The customer must have a subscription to OPERA Cloud Foundation | |
| Oracle Hospitality Integration Platform (OHIP) | For custopmers OHIP is included in the subscription to OPERA Cloud Foundation.<br>Partners must subscribe to Oracle Hospitality Integration Cloud Service. | |

## Configuration of OPERA Controls

To activate the Create Block functionality, you must ensure the following OPERA Controls are active in the target property. You can use the following API operations to validate the settings:

**Ensure that application function BLOCKS - BUSINESS BLOCK is active**

RQ: ent/config/v1/settings?hotelId={{HotelId}}&parameterNameWildCard=BUSINESS BLOCKblk/config/v1/blockStatusCodes

### Check if the site is using USE SINGLE BLOCK STATUS

RQ: ent/config/v1/settings?hotelId={{HotelId}}&parameterNameWildCard=USE SINGLE BLOCK STATUS

Other Required Configuration Elements:

**Block Status**

RQ: blk/config/v1/blockStatusCodes

Block Status Flow

RQ: blk/config/v1/blockStatusCodes/{{scrBlockStatusCode}}/nextStatusCodes

**Reservation Type**

RQ: rsv/config/v1/guaranteeCodes?pageNumber=1&hotelIds={{hotelId}}

**Block Lost / Cancel / Refused Reasons**

RQ: blk/config/v1/blockLostBookingCodes

blk/config/v1/blockCancellationReasons

blk/config/v1/blockRefusedReasons

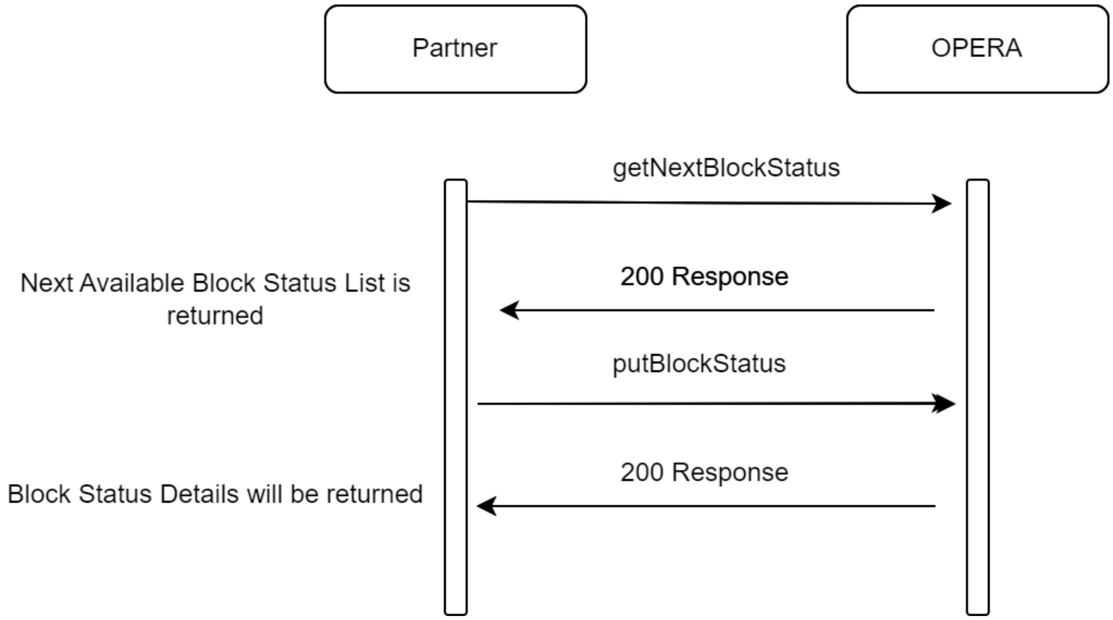
**Reservation Cancellation Reasons**

RQ: rsv/config/v1/cancellationCodes?fetchInactive=false

## Workflow

Step by step solution including:

**Figure 5-1 The sequence of API calls (in any direction)**


Partner
OPERA
getNextBlockStatus
Next Available Block Status List is returned
200 Response
putBlockStatus
Block Status Details will be returned
200 Response

### Description of Steps

A typical process to change a block status is as follows:

1. Check which next available statuses can be selected using the **getNextBlockStatus RQ** with parameter **currentStatus** set to the current Block Status of the block you would like to change.
2. Change the status of the block using the **putBlockStatus RQ** .
   a. Ensure that elements **currentBlockStatus**, **newBlockStatus**, and **reservationType** are sent with valid values.
   b. Two flags are available in the RQ to override rooms availability and function space availability warnings. These are **overbookAll** and **overrideEventsProcessingWarnings**.
   c. Ensure the reservation type sent is correct based on the new block status code. Blocks on INQUIRY or NON DEDUCT status require a non-deduct reservation type. Blocks on DEDUCT or ACTUAL status require a DEDUCT reservation type.
   d. If a catering status exists on the block and contains the same value as the rooms status, you can set the flag **applyChangesToCateringStatus** to true. This also applies the selected **newBlockStatus** to the catering status.
   e. If **BLOCKS - USE SINGLE BLOCK STATUS OPERA** Control is active, ensure that the block status and catering status are always changed to the same new value.
   f. When changing the status to a LOST / CANCEL / REFUSED status, the lost / cancellation / refused reason becomes mandatory. Therefore, if your new status is a CANCEL type status, verify that collection **cancellationDetails** contains valid values.
   g. When a Posting master already exists for the block and you want to cancel the Posting Master with the same RQ as the block cancellation, verify the following:
      i. The reservation cancellation for the Posting Master is contained within your RQ in section cancellationDetails.
      ii. Flag cancelAllPMReservations is set to true.
   h. Note that it is not possible to cancel a block when there are active reservations other than the Posting Master already linked to the block. It is necessary to first cancel or unlink all reservations before block cancellation is allowed.

**Sample calls**: Sample API calls for steps described in the diagram.

*Sample for* **getBlockNextStatus**

**Get List of Next Available Status Codes**

GET**{{HostName}}/blk/v1/blocks/status?currentStatus=TEN&hotelId={{HotelId}}&includeCateringStatus=true**

**No Body**

**200 Response: OK**

**Body:**

```
{
    "blockNextStatusList": [
        {
            "bookingStatus": {
                "status": {
                    "code": "DEF",
                    "description": "Definite"
                }
            },
            "parentBookingStatus": "TEN",
            "deductInventory": true,
            "catDeductInventory": true,
            "allowPickup": true,
            "returnToInventory": false,
            "defaultReservationType": "GDED"
        },
        {
            "bookingStatus": {
                "status": {
                    "code": "ACT",
                    "description": "Actual"
                }
            },
            "parentBookingStatus": "DEF",
            "deductInventory": true,
            "catDeductInventory": true,
            "allowPickup": true,
            "returnToInventory": false,
            "defaultReservationType": "GDED"
        },
        {
            "bookingStatus": {
                "status": {
                    "code": "TDN",
                    "description": "Turndown New"
                }
            },
            "parentBookingStatus": "DEF",
            "deductInventory": false,
            "catDeductInventory": false,
            "allowPickup": false,
            "returnToInventory": true
        },
        {
            "bookingStatus": {
                "status": {
                    "code": "REG",
                    "description": "Regret Type"
                }
            },
            "parentBookingStatus": "DEF",
            "deductInventory": false,
            "catDeductInventory": false,
            "allowPickup": false,
            "returnToInventory": true
        },
        {
            "bookingStatus": {
                "status": {
                    "code": "REF",
                    "description": "Refused"
                }
            },
            "parentBookingStatus": "DEF",
            "deductInventory": false,
            "catDeductInventory": false,
            "allowPickup": false,
            "returnToInventory": true,
            "defaultReservationType": "GNON"
        },
        {
            "bookingStatus": {
                "status": {
                    "code": "LOST",
                    "description": "Lost"
                }
            },
            "parentBookingStatus": "DEF",
            "deductInventory": false,
            "catDeductInventory": false,
            "allowPickup": false,
            "returnToInventory": true,
            "defaultReservationType": "GNON"
        },
        {
            "bookingStatus": {
                "status": {
                    "code": "LOS",
                    "description": "Lost"
                }
            },
            "parentBookingStatus": "DEF",
            "deductInventory": false,
            "catDeductInventory": false,
            "allowPickup": false,
            "returnToInventory": true
        },
        {
            "bookingStatus": {
                "status": {
                    "code": "CAN",
                    "description": "Cancel"
                }
            },
            "parentBookingStatus": "DEF",
            "deductInventory": false,
            "catDeductInventory": false,
            "allowPickup": false,
            "returnToInventory": true
        },
        {
            "bookingStatus": {
                "status": {
                    "code": "ALO",
                    "description": "Allotment...."
                }
            },
            "parentBookingStatus": "TEN",
            "deductInventory": false,
            "catDeductInventory": false,
            "allowPickup": true,
            "returnToInventory": false
        },
        {
            "bookingStatus": {
                "status": {
                    "code": "LOST",
                    "description": "Lost"
                }
            },
            "parentBookingStatus": "ALO",
            "deductInventory": false,
            "catDeductInventory": false,
            "allowPickup": false,
            "returnToInventory": true,
            "defaultReservationType": "GNON"
        },
        {
            "bookingStatus": {
                "status": {
                    "code": "TE1",
                    "description": "Tentative"
                }
            },
            "parentBookingStatus": "ALO",
            "deductInventory": true,
            "catDeductInventory": false,
            "allowPickup": false,
            "returnToInventory": false,
            "defaultReservationType": "9"
        },
        {
            "bookingStatus": {
                "status": {
                    "code": "REF",
                    "description": "Refused"
                }
            },
            "parentBookingStatus": "ALO",
            "deductInventory": false,
            "catDeductInventory": false,
            "allowPickup": false,
            "returnToInventory": true,
            "defaultReservationType": "GNON"
        },
        {
            "bookingStatus": {
                "status": {
                    "code": "TE1",
                    "description": "Tentative"
                }
            },
            "parentBookingStatus": "TEN",
            "deductInventory": true,
            "catDeductInventory": false,
            "allowPickup": false,
            "returnToInventory": false,
            "defaultReservationType": "9"
        },
        {
            "bookingStatus": {
                "status": {
                    "code": "LOS",
                    "description": "Lost"
                }
            },
            "parentBookingStatus": "TE1",
            "deductInventory": false,
            "catDeductInventory": false,
            "allowPickup": false,
            "returnToInventory": true
        },
        {
            "bookingStatus": {
                "status": {
                    "code": "REG",
                    "description": "Regret Type"
                }
            },
            "parentBookingStatus": "TE1",
            "deductInventory": false,
            "catDeductInventory": false,
            "allowPickup": false,
            "returnToInventory": true
        },
        {
            "bookingStatus": {
                "status": {
                    "code": "PEN",
                    "description": "Pending x"
                }
            },
            "parentBookingStatus": "TE1",
            "deductInventory": false,
            "catDeductInventory": true,
            "allowPickup": false,
            "returnToInventory": false,
            "defaultReservationType": "GNON"
        },
        {
            "bookingStatus": {
                "status": {
                    "code": "OPEN",
                    "description": "Open for pickup"
                }
            },
            "parentBookingStatus": "TE1",
            "deductInventory": false,
            "catDeductInventory": true,
            "allowPickup": true,
            "returnToInventory": false
        },
        {
```

```
            "bookingStatus": {
                "status": {
                    "code": "TOPN",
                    "description": "Open Pickup"
                }
            },
            "parentBookingStatus": "TEN",
            "deductInventory": false,
            "catDeductInventory": false,
            "allowPickup": true,
            "returnToInventory": false,
            "defaultReservationType": "CC"
        },
        {
            "bookingStatus": {
                "status": {
                    "code": "LOST",
                    "description": "Lost"
                }
            },
            "parentBookingStatus": "TOPN",
            "deductInventory": false,
            "catDeductInventory": false,
            "allowPickup": false,
            "returnToInventory": true,
            "defaultReservationType": "GNON"
        },
        {
            "bookingStatus": {
                "status": {
                    "code": "OPEN",
                    "description": "Open for pickup"
                }
            },
            "parentBookingStatus": "TEN",
            "deductInventory": false,
            "catDeductInventory": true,
            "allowPickup": true,
            "returnToInventory": false
        },
        {
            "bookingStatus": {
                "status": {
                    "code": "ACT",
                    "description": "Actual"
                }
            },
            "parentBookingStatus": "OPEN",
            "deductInventory": true,
            "catDeductInventory": true,
            "allowPickup": true,
            "returnToInventory": false,
            "defaultReservationType": "GDED"
        },
        {
            "bookingStatus": {
```

```
            "status": {
                "code": "CAN",
                "description": "Cancel"
            }
        },
        "parentBookingStatus": "OPEN",
        "deductInventory": false,
        "catDeductInventory": false,
        "allowPickup": false,
        "returnToInventory": true
    }
],
"cateringNextStatusList": [
    {
        "bookingStatus": {
            "status": {
                "code": "DEF",
                "description": "Definite"
            }
        },
        "parentBookingStatus": "TEN",
        "deductInventory": true,
        "catDeductInventory": true,
        "allowPickup": true,
        "returnToInventory": false,
        "defaultReservationType": "GDED"
    },
    {
        "bookingStatus": {
            "status": {
                "code": "ACT",
                "description": "Actual"
            }
        },
        "parentBookingStatus": "DEF",
        "deductInventory": true,
        "catDeductInventory": true,
        "allowPickup": true,
        "returnToInventory": false,
        "defaultReservationType": "GDED"
    },
    {
        "bookingStatus": {
            "status": {
                "code": "TDN",
                "description": "Turndown New"
            }
        },
        "parentBookingStatus": "DEF",
        "deductInventory": false,
        "catDeductInventory": false,
        "allowPickup": false,
        "returnToInventory": true
    },
    {
        "bookingStatus": {
            "status": {
                "code": "REG",
                "description": "Regret Type"
            }
        },
        "parentBookingStatus": "DEF",
        "deductInventory": false,
        "catDeductInventory": false,
        "allowPickup": false,
        "returnToInventory": true
    },
    {
        "bookingStatus": {
            "status": {
                "code": "REF",
                "description": "Refused"
            }
        },
        "parentBookingStatus": "DEF",
        "deductInventory": false,
        "catDeductInventory": false,
        "allowPickup": false,
        "returnToInventory": true,
        "defaultReservationType": "GNON"
    },
    {
        "bookingStatus": {
            "status": {
                "code": "LOST",
                "description": "Lost"
            }
        },
        "parentBookingStatus": "DEF",
        "deductInventory": false,
        "catDeductInventory": false,
        "allowPickup": false,
        "returnToInventory": true,
        "defaultReservationType": "GNON"
    },
    {
        "bookingStatus": {
            "status": {
                "code": "LOS",
                "description": "Lost"
            }
        },
        "parentBookingStatus": "DEF",
        "deductInventory": false,
        "catDeductInventory": false,
        "allowPickup": false,
        "returnToInventory": true
    },
    {
        "bookingStatus": {
            "status": {
                "code": "CAN",
                "description": "Cancel"
            }
        },
        "parentBookingStatus": "DEF",
        "deductInventory": false,
        "catDeductInventory": false,
        "allowPickup": false,
        "returnToInventory": true
    },
    {
        "bookingStatus": {
            "status": {
                "code": "ALO",
                "description": "Allotment...."
            }
        },
        "parentBookingStatus": "TEN",
        "deductInventory": false,
        "catDeductInventory": false,
        "allowPickup": true,
        "returnToInventory": false
    },
    {
        "bookingStatus": {
            "status": {
                "code": "LOST",
                "description": "Lost"
            }
        },
        "parentBookingStatus": "ALO",
        "deductInventory": false,
        "catDeductInventory": false,
        "allowPickup": false,
        "returnToInventory": true,
        "defaultReservationType": "GNON"
    },
    {
        "bookingStatus": {
            "status": {
                "code": "TE1",
                "description": "Tentative"
            }
        },
        "parentBookingStatus": "ALO",
        "deductInventory": true,
        "catDeductInventory": false,
        "allowPickup": false,
        "returnToInventory": false,
        "defaultReservationType": "9"
    },
    {
        "bookingStatus": {
            "status": {
                "code": "REF",
                "description": "Refused"
            }
```

```
    },
    "parentBookingStatus": "ALO",
    "deductInventory": false,
    "catDeductInventory": false,
    "allowPickup": false,
    "returnToInventory": true,
    "defaultReservationType": "GNON"
},
{
    "bookingStatus": {
        "status": {
            "code": "TE1",
            "description": "Tentative"
        }
    },
    "parentBookingStatus": "TEN",
    "deductInventory": true,
    "catDeductInventory": false,
    "allowPickup": false,
    "returnToInventory": false,
    "defaultReservationType": "9"
},
{
    "bookingStatus": {
        "status": {
            "code": "LOS",
            "description": "Lost"
        }
    },
    "parentBookingStatus": "TE1",
    "deductInventory": false,
    "catDeductInventory": false,
    "allowPickup": false,
    "returnToInventory": true
},
{
    "bookingStatus": {
        "status": {
            "code": "REG",
            "description": "Regret Type"
        }
    },
    "parentBookingStatus": "TE1",
    "deductInventory": false,
    "catDeductInventory": false,
    "allowPickup": false,
    "returnToInventory": true
},
{
    "bookingStatus": {
        "status": {
            "code": "PEN",
            "description": "Pending x"
        }
    },
    "parentBookingStatus": "TE1",
    "deductInventory": false,
    "catDeductInventory": true,
    "allowPickup": false,
    "returnToInventory": false,
    "defaultReservationType": "GNON"
},
{
    "bookingStatus": {
        "status": {
            "code": "OPEN",
            "description": "Open for pickup"
        }
    },
    "parentBookingStatus": "TE1",
    "deductInventory": false,
    "catDeductInventory": true,
    "allowPickup": true,
    "returnToInventory": false
},
{
    "bookingStatus": {
        "status": {
            "code": "TOPN",
            "description": "Open Pickup"
        }
    },
    "parentBookingStatus": "TEN",
    "deductInventory": false,
    "catDeductInventory": false,
    "allowPickup": true,
    "returnToInventory": false,
    "defaultReservationType": "CC"
},
{
    "bookingStatus": {
        "status": {
            "code": "LOST",
            "description": "Lost"
        }
    },
    "parentBookingStatus": "TOPN",
    "deductInventory": false,
    "catDeductInventory": false,
    "allowPickup": false,
    "returnToInventory": true,
    "defaultReservationType": "GNON"
},
{
    "bookingStatus": {
        "status": {
            "code": "OPEN",
            "description": "Open for pickup"
        }
    },
    "parentBookingStatus": "TEN",
    "deductInventory": false,
            "catDeductInventory": true,
            "allowPickup": true,
            "returnToInventory": false
        },
        {
            "bookingStatus": {
                "status": {
                    "code": "ACT",
                    "description": "Actual"
                }
            },
            "parentBookingStatus": "OPEN",
            "deductInventory": true,
            "catDeductInventory": true,
            "allowPickup": true,
            "returnToInventory": false,
            "defaultReservationType": "GDED"
        },
        {
            "bookingStatus": {
                "status": {
                    "code": "CAN",
                    "description": "Cancel"
                }
            },
            "parentBookingStatus": "OPEN",
            "deductInventory": false,
            "catDeductInventory": false,
            "allowPickup": false,
            "returnToInventory": true
        }
    ],
    "links": []
}
```

*Sample for* **putBlockStatus (Change from NON DEDUCT to DEDUCT)**

> **Note**
>
> Although there is a start and end date for each grid cell, in this call, the start and end date in each collection must be the same (and the collection must be repeated for any additional dates)/
>
> This call is for a block that is not open for pickup. Note the element: "allocation": "INITIAL." When running this RQ for an open-for-pickup call, the elements must be :"allocation": "ACTUAL."

**Change Block Status from NON DEDUCT to DEDUCT**

PUT**{{HostName}}**/blk/v1/hotels/**{{HotelId}}**/blocks/**{{BlockId}}**/status

Body:

```
{
    "verificationOnly": false,
    "changeBlockStatus": {
        "hotelId": "{{HotelId}}",
        "blockId": {
            "type": "Block",
            "idContext": "OPERA",
            "id": "{{BlockId}}"
        },
        "currentBlockStatus": "TEN",
        "newBlockStatus": "DEF",
        "reservationType": "GRPGTD",
        "overbookAll": false,
        "cancelAllPMReservations": false,
        "applyChangesToCateringSatus": true,
        "overrideEventsProcessingWarnings": true
    }
}
```

**200 Response**: OK

Body:

```
{
    "block": {
        "blockIdList": [
            {
                "id": "916877",
                "type": "Block",
                "idContext": "OPERA"
            }
        ],
        "blockDetails": {
            "blockStatus": {
                "bookingStatus": {
                    "status": {
                        "code": "DEF",
                        "description": "Definite"
                    }
                }
            },
            "reservationType": {
                "reservationType": "GDED"
            },
            "status": "O",
            "allowRateOverride": false,
            "manualCutOff": false,
            "wholesaleBlock": false
        },
        "markAsRecentlyAccessed": false
    },
    "cancellationDetails": {
        "cancellationCode": {}
    },
    "blockNextStatusList": [
        {
            "bookingStatus": {
                "status": {
                    "code": "LOST",
                    "description": "Lost"
                }
            },
            "parentBookingStatus": "DEF",
            "deductInventory": false,
            "allowPickup": false,
            "returnToInventory": true,
            "defaultReservationType": "GNON",
            "cancellationType": "Lost"
        },
        {
            "bookingStatus": {
                "status": {
                    "code": "ACT",
                    "description": "Actual"
                }
            },
            "parentBookingStatus": "DEF",
            "deductInventory": true,
            "allowPickup": true,
            "returnToInventory": false,
            "defaultReservationType": "GDED"
        },
        {
            "bookingStatus": {
                "status": {
                    "code": "LOS",
                    "description": "Lost"
                }
            },
            "parentBookingStatus": "DEF",
            "deductInventory": false,
            "allowPickup": false,
            "returnToInventory": true,
            "cancellationType": "Lost"
        },
        {
            "bookingStatus": {
                "status": {
                    "code": "CAN",
                    "description": "Cancel"
                }
            },
            "parentBookingStatus": "DEF",
            "deductInventory": false,
            "allowPickup": false,
            "returnToInventory": true,
            "cancellationType": "Cancel"
        },
        {
            "bookingStatus": {
```

```
                    "status": {
                        "code": "REG",
                        "description": "Regret Type"
                    }
                },
                "parentBookingStatus": "DEF",
                "deductInventory": false,
                "allowPickup": false,
                "returnToInventory": true,
                "cancellationType": "Refused"
            },
            {
                "bookingStatus": {
                    "status": {
                        "code": "REF",
                        "description": "Refused"
                    }
                },
                "parentBookingStatus": "DEF",
                "deductInventory": false,
                "allowPickup": false,
                "returnToInventory": true,
                "defaultReservationType": "GNON",
                "cancellationType": "Refused"
            },
            {
                "bookingStatus": {
                    "status": {
                        "code": "TDN",
                        "description": "Turndown New"
                    }
                },
                "parentBookingStatus": "DEF",
                "deductInventory": false,
                "allowPickup": false,
                "returnToInventory": true,
                "cancellationType": "Cancel"
            }
        ],
        "blockStatusChangeHistory": [
            {
                "sequence": 1,
                "bookingStatus": {
                    "code": "INQ",
                    "description": "Inquiry Non Deduct"
                },
                "modifyDateTime": "2024-02-14",
                "modifierId": "SPAVFIRST"
            },
            {
                "sequence": 2,
                "bookingStatus": {
                    "code": "TEN",
                    "description": "Tentative"
                },
                "modifyDateTime": "2024-02-14",
                "modifierId": "SPAVFIRST"
            },
            {
                "sequence": 3,
                "bookingStatus": {
                    "code": "DEF",
                    "description": "Definite"
                },
                "modifyDateTime": "2024-02-14",
                "modifierId": "SPAVFIRST"
            }
        ],
        "links": []
    }
```

*Sample for* **putBlockStatus (change from DEDUCT to CANCEL)**

> **Note**
>
> The start and end date in the collection define the date range for which the given values should be populated. Only one range can be defined per call.
>
> This call is for a block that is open for pickup. Note the element: "allocation": "Actual." When running this RQ for a not open-for-pickup call, the elements must be :"allocation": "Initial."

**Change Block Status from DEDUCT to CANCEL**

PUT **{{HostName}}**/blk/v1/hotels/**{{HotelId}}**/blocks/**{{BlockId}}**/status

Body:

```
{
    "verificationOnly": false,
    "changeBlockStatus": {
        "hotelId": "BHOTEL",
        "blockId": {
            "type": "Block",
            "idContext": "OPERA",
            "id": "916877"
        },
        "currentBlockStatus": "DEF",
        "newBlockStatus": "CAN",
        "reservationType": "GDED",
                "cancellationDetails": {
            "cancellationCode": {
                "code": "RATE"
            },
            "cancellationInfo": "Rate to High"
        },
        "pMReservationsCancellationDetails": {
            "cancellationCode": {
                "code": "6PM",
                "description": "6 PM Cancellation"
            }
        },
        "overbookAll": false,
        "applyChangesToCateringSatus": true,
        "overrideEventsProcessingWarnings": false
    }
}
```

**200 Response**: OK

Body:

```
{
    "block": {
        "blockIdList": [
            {
                "id": "916877",
                "type": "Block",
                "idContext": "OPERA"
            }
        ],
        "blockDetails": {
            "blockStatus": {
                "bookingStatus": {
                    "status": {
                        "code": "CAN",
                        "description": "Cancel"
                    }
                }
            },
            "reservationType": {
                "reservationType": "GDED"
            },
            "status": "C",
            "allowRateOverride": false,
            "manualCutOff": false,
            "wholesaleBlock": false
        },
        "markAsRecentlyAccessed": false
    },
    "cancellationDetails": {
        "cancellationNumber": {
            "id": "2040306",
            "type": "Cancellation",
            "idContext": "OPERA"
        },
        "cancellationCode": {
            "code": "RATE"
        },
        "cancellationDate": "2023-09-22",
        "cancellationInfo": "Rate to High"
    },
    "blockNextStatusList": [],
    "blockStatusChangeHistory": [
        {
            "sequence": 1,
            "bookingStatus": {
                "code": "INQ",
                "description": "Inquiry Non Deduct"
            },
            "modifyDateTime": "2024-02-14",
            "modifierId": "SPAVFIRST"
        },
        {
            "sequence": 2,
            "bookingStatus": {
                "code": "TEN",
                "description": "Tentative"
            },
            "modifyDateTime": "2024-02-14",
            "modifierId": "SPAVFIRST"
        },
        {
            "sequence": 3,
            "bookingStatus": {
                "code": "DEF",
                "description": "Definite"
            },
            "modifyDateTime": "2024-02-14",
            "modifierId": "SPAVFIRST"
        },
        {
            "sequence": 4,
            "bookingStatus": {
                "code": "CAN",
                "description": "Cancel"
            },
            "modifyDateTime": "2024-02-14",
            "modifierId": "SPAVFIRST"
        }
    ],
    "links": []
}
```

### References

- For more information, review the Anti-Patterns chapter in the OHIP user guide to learn more about functional and technical anti-patterns.
- Fore more information, refer to the Blocks introduction topic in the OPERA Cloud Services user guide.

# 6
# Update Room and Rate Grid

### Business Context

A block is an allocation of rooms for multiple room types, reserved for a specific date period, that does not require you to create individually named reservations. Blocks are used to reserve multiple rooms for events, such as weddings or conferences, inbound tour groups or airline crews, or for long-term allocations, such as airline allotments. To allocate guest rooms for a group of travellers, a **block header** must exist (for more information, refer to Create a Block with or without Room Grid). In this document, you can follow the steps to load a block room allocation using REST APIs. There are different operations available to use: two synchnronous and two asynchronous. It is recommended to use the asynchronous calls for longer blocks (that is, greater than 30 days).

### Prerequisites

**Table 6-1 Required Software Tools**

| Tool | Description | Links |
|---|---|---|
| Postman | Postman is an API Platform that allows you to design, create and test API's.<br>Use Postman to send API requests or to use the Postman collections provided. | Using the Oracle Hospitality APIs<br>Tool |

**Table 6-2 Current Versions Required**

| OPERA Cloud Platform Module | Description | Minimum Version |
|---|---|---|
| OPERA Cloud Services | The customer must have a subscription to OPERA Cloud Foundation | |
| Oracle Hospitality Integration Platform (OHIP) | For customers, OHIP is included in the subscription to OPERA Cloud Foundation.<br>Partners need a subscription to the Oracle Hospitality Integration Cloud Service. | |

### Configuration of OPERA Controls

To activate the Create Block functionality, you must ensure the following OPERA Controls are active in the target property. You can use the following API operations to validate the settings:

**Ensure that application function BLOCKS - BUSINESS BLOCK is active**

RQ: ent/config/v1/settings?hotelId={{HotelId}}&parameterNameWildCard=BUSINESS BLOCK

**Check if the site is using OCCUPANCY SPLIT BY ROOM TYPE**

RQ: ent/config/v1/settings?hotelId={{HotelId}}&parameterNameWildCard=OCCUPANCY SPLIT PER ROOM TYPE

**Check if the site is using RATE CODE MANDATORY BLOCK HEADER**

RQ: ent/config/v1/settings?hotelId={{HotelId}}&parameterNameWildCard=RATE CODE MANDATORY BLOCK HEADER

Critera dependencies:

**Block Status**

RQ: blk/config/v1/blockStatusCodes

**Rate Code (if mandatory)**

RQ: rtp/v1/ratePlans?includeRateInformation=false&includeInactive=true&offset=1&limit=200&sellDate=2024-02-08&hotelId={{hotelId}}

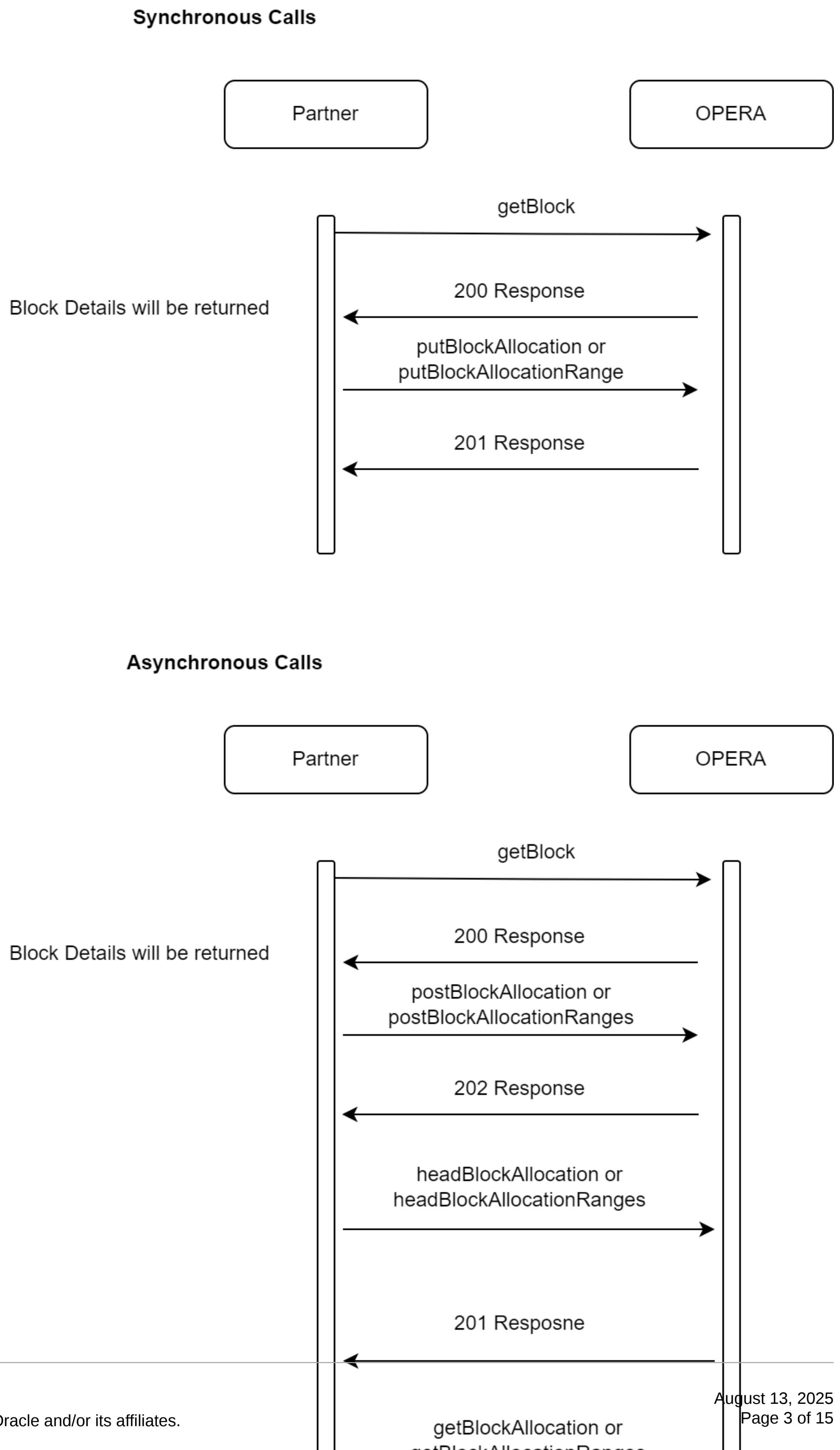
**Room Types**

RQ: rm/config/v1/hotels/{{hotelId}}/roomTypes?accessible=false&includeInactive=false&offset=1&hotelIds={{hoteldId}}&summaryInfo=true&limit=25&physical=true&pseudo=false

### Workflow

Step by step solution including:

**Figure 6-1 The sequence of API calls (in any direction)**



### Description of Steps

A typical process to update the room and rate grid is as follows:

1. Ensure that the block header you are trying to add Room and Rate grid exists using the **getBlock RQ**.
2. Add the room block using **putBlockAllocation**, **putBlockAllocationRange**, **postBlockAllocation (async)**, or **postBlockAllocationRange (async)**.
    a. When the **OCCUPANCY PER ROOM TYPE OPERA** control is not active, only submit inventory updates for the onePerson element. When the OPERA Control is active, inventory updates can be done for onePerson, twoPerson, threePerson, and FourPerson. Rates can be submitted for all occupancy levels, regardless of the OPERA Control setting.

    b. When the block is not in an open-for-pickup status, meaning reservations cannot be made against the block, the allocation element must carry the value INITIAL.

    c. When the block is open for pickup, any inventory updates must be done for allocation = ACTUAL.

    d. If a valid rate code has been submitted as part of the block creation process, it is not necessary to send rate amounts during the grid update process. Rate amounts in OPERA are populated automatically from the rate code on the block header.
3. Description of differences between the four different RQs.
    a. **putBlockAllocation**

        i. Contains one element per Date per Room Type (but elements can be repeated within the same call to loop through).

        ii. Start and End Date for each collection need to be the same

        iii. Recommended for shorter date ranges ( < 30 nights) where the inventory or rate values vary between different dates / room types.

    b. **putBlockAllocationRange**

        i. Contains one collection that applies the same inventory and rate values to a date range and list of room types.

        ii. Start and End Date for each collection define the date range.

        iii. Recommended for shorter date ranges ( < 30 nights) where the inventory and rate values are the same for all dates and room types within the range / collection.

        iv. Only one range can be contained within one call

    c. **postBlockAllocation (async)**

        > **Note**
        >
        > Only available from OPERA Cloud version 23.4 onwards.

        i. Contains one element per Date per Room Type (but elements can be repeated within the same call to loop through).

        ii. Start and End Date for each collection must be the same.

        iii. Recommended for longer date ranges ( > 30 nights) where the inventory or rate values vary between different dates / room types.

d. **postBlockAllocationRange (async)**

> **Note**
>
> Only available from OPERA Cloud version 23.5 onwards.

i. Can contain multiple collections where each collection applies the same inventory and rate values to the date range and list of room types for that collection.

ii. Start and End Date for each collection define the date range.

iii. Recommended for longer date ranges ( > 30 nights) where the inventory and rate values are the same for multiple dates and room types within the range / collection.

iv. Multiple ranges can be posted within one call in a loop.

4. Retrieve the udpated Block using **getBlock** with fetchInstructions for the room grid to ensure that the block was updated with the correct attributes (optional).

**Sample calls**: Sample API calls for steps described in the diagram.

*Sample for* **getBlock**

**Fetch Block with Grid Details**

GET**{{HostName}}**/blk/v1/hotels/{{HotelId}}/blocks/{{BlockId}}?fetchAllocatedRoomTypes=true&markAsRecentlyAccessed=false&roomTypes=DLXQ&roomTypes=STDK&roomTypes=STDQ&roomAllocationCriteria=Initial&fetchInstructions=RateGrid&numberOfDays=2&startDate=2024-09-06&roomAllocationCriteria=CutoffDate

**200 Response**: OK

```
{
    "blocks": {
        "blockInfo": [
            {
                "block": {
                    "blockIdList": [
                        {
                            "id": "{{BLockId}}",
                            "type": "Block",
                            "idContext": "OPERA"
                        }
                    ],
                    "roomAllocations": {
                        "roomAllocationType": [
                            {
                                "roomAllocationInfo": [
                                    {
                                        "roomGridInvSummary": [
                                            {}
                                        ],
                                        "availableRoomGridInvSummary": [
                                            {}
                                        ],
                                        "stayDate": "2024-09-06",
                                        "extendedStay": false
```

```
},
{
    "roomGridInvSummary": [
        {}
    ],
    "availableRoomGridInvSummary": [
        {}
    ],
    "stayDate": "2022-05-19",
    "extendedStay": false
},
{
    "roomGridInvSummary": [
        {}
    ],
    "availableRoomGridInvSummary": [
        {}
    ],
    "stayDate": "2022-05-20",
    "extendedStay": false
},
{
    "roomGridInvSummary": [
        {}
    ],
    "availableRoomGridInvSummary": [
        {}
    ],
    "stayDate": "2022-05-21",
    "extendedStay": false
},
{
    "roomGridInvSummary": [
        {}
    ],
    "availableRoomGridInvSummary": [
        {}
    ],
    "stayDate": "2022-05-22",
    "extendedStay": false
},
{
    "roomGridInvSummary": [
        {}
    ],
    "availableRoomGridInvSummary": [
        {}
    ],
    "stayDate": "2022-05-23",
    "extendedStay": false
},
{
    "roomGridInvSummary": [
        {}
    ],
    "availableRoomGridInvSummary": [
                {}
            ],
            "stayDate": "2022-05-24",
            "extendedStay": false
        }
    ],
    "allocation": "Initial"
},
{
    "roomAllocationInfo": [
        {
            "roomGridInvSummary": [
                {
                    "onePerson": 0,
                    "twoPerson": 0,
                    "threePerson": 0,
                    "fourPerson": 0
                }
            ],
            "availableRoomGridInvSummary": [
                {}
            ],
            "stayDate": "2022-05-18",
            "extendedStay": false
        },
        {
            "roomGridInvSummary": [
                {
                    "onePerson": 0,
                    "twoPerson": 0,
                    "threePerson": 0,
                    "fourPerson": 0
                }
            ],
            "availableRoomGridInvSummary": [
                {}
            ],
            "stayDate": "2022-05-19",
            "extendedStay": false
        },
        {
            "roomGridInvSummary": [
                {
                    "onePerson": 0,
                    "twoPerson": 0,
                    "threePerson": 0,
                    "fourPerson": 0
                }
            ],
            "availableRoomGridInvSummary": [
                {}
            ],
            "stayDate": "2022-05-20",
            "extendedStay": false
        },
        {
```

```
        "roomGridInvSummary": [
            {
                "onePerson": 0,
                "twoPerson": 0,
                "threePerson": 0,
                "fourPerson": 0
            }
        ],
        "availableRoomGridInvSummary": [
            {}
        ],
        "stayDate": "2022-05-21",
        "extendedStay": false
    },
    {
        "roomGridInvSummary": [
            {
                "onePerson": 0,
                "twoPerson": 0,
                "threePerson": 0,
                "fourPerson": 0
            }
        ],
        "availableRoomGridInvSummary": [
            {}
        ],
        "stayDate": "2022-05-22",
        "extendedStay": false
    },
    {
        "roomGridInvSummary": [
            {
                "onePerson": 0,
                "twoPerson": 0,
                "threePerson": 0,
                "fourPerson": 0
            }
        ],
        "availableRoomGridInvSummary": [
            {}
        ],
        "stayDate": "2022-05-23",
        "extendedStay": false
    },
    {
        "roomGridInvSummary": [
            {
                "onePerson": 0,
                "twoPerson": 0,
                "threePerson": 0,
                "fourPerson": 0
            }
        ],
        "availableRoomGridInvSummary": [
            {}
        ],
                                        "stayDate": "2022-05-24",
                                        "extendedStay": false
                                    }
                                ],
                                "allocation": "CutoffDate"
                            }
                        ],
                        "masterInfo": {
                            "roomTypes": []
                        },
                        "startDate": "2022-05-18",
                        "numberOfDays": 7
                    },
                    "hotelId": "BHOTEL",
                    "markAsRecentlyAccessed": false
                }
            }
        ]
    },
    "links": []
}
```

*Sample for* **putBlockAllocation**

> **Note**
>
> Although there is a start and end date for each grid cell, in this call, the start and end date in each collection must be the same (and the collection must be repeated for any additional dates).
>
> This call is for a block that is not open for pickup. Please note the element: "allocation": "INITIAL". When running this RQ for an open-for-pickup call, the elements must be :"allocation": "ACTUAL."

**Create / Update Room Grid (Cell by Cell)**

PUT**{{HostName}}**/blk/v1/hotels/**{{HotelId}}**/blocks/**{{BlockId}}**/allocation

Body:

```
{
    "criteria": {
        "hotelId": "{{HotelId}}",
        "blockId": {
            "type": "Block",
            "idContext": "OPERA",
            "id": "{{BlockId}}"
        },
        "allocationRoomTypes": [
            {
                "allocationGridDates": [
                    {
                        "roomAllocationInfo": [
                            {
```

```
                        "inventory": {
                            "forceOverbook": false
                        },
                        "rate": {
                            "onePerson": "240",
                            "twoPerson": "240",
                            "threePerson": "260"
                        },
                        "start": "2023-12-01",
                        "end": "2023-12-01"
                    },
                    {
                        "inventory": {
                            "forceOverbook": false
                        },
                        "rate": {
                            "onePerson": "200",
                            "twoPerson": "220"
                        },
                        "start": "2023-12-02",
                        "end": "2023-12-02"
                    }
                ],
                "allocation": "RATES"
            },
            {
                "roomAllocationInfo": [
                    {
                        "inventory": {
                            "forceOverbook": false,
                            "onePerson": "4",
                            "twoPerson": "4"
                        },
                        "rate": {},
                        "start": "2023-12-01",
                        "end": "2023-12-01"
                    },
                    {
                        "inventory": {
                            "forceOverbook": false,
                            "onePerson": "3",
                            "twoPerson": "3"
                        },
                        "rate": {},
                        "start": "2023-12-02",
                        "end": "2023-12-02"
                    }
                ],
                "allocation": "INITIAL"
            }
        ],
        "roomType": "EXEC"
    }
],
"genericRoomType": false
```

```
    }
}
```

**200 Response**: OK

*Sample for* **putBlockAllocationRange**

> **Note**
>
> The start and end date in the collection define the date range for which the given values should be populated. Only one range can be defined per call.
>
> This call is for a block that is open for pickup. Please note the element: "allocation": "Actual". When running this RQ for a not open-for-pickup call, the elements must be :"allocation": "Initial"

**Create / Update Room Grid (Range)**

PUT**{{HostName}}**/blk/v1/hotels/**{{HotelId}}**/blocks/**{{BlockId}}**/allocationRange

Body:

```
{
    "blockAllocationRange": {
        "blockId": {
            "type": "Block",
            "idContext": "OPERA",
            "id": "{{BlockId}}"
        },
        "hotelId": "{{HotelId}} ",
        "roomTypes": [
            "DLXQ"
        ],
        "beginDate": "2023-12-01",
        "endDate": "2023-12-02",
        "allocationType": "Actual",
        "incrementFlag": false,
        "blockInventory": {
            "onePerson": "8",
            "twoPerson": "3"
        },
        "blockRates": {
            "onePerson": "240",
            "twoPerson": "280"
        },
        "includedDays": "1111111",
        "rangeMode": "Core",
        "genericRoomType": false
    }
}
```

**200 Response**: OK

*Sample for* **postBlockAllocation (async)**

### Note

Even though there is a start and end date for each grid cell, in this call the start and end date in each collection must be the same (and the collection must be repeated for any additional dates)

This call is for a block that is not open for pickup. Please note the element: "allocation": "INITIAL". When running this RQ for an open-for-pickup call, the elements must be :"allocation": "ACTUAL"

**Post Block Allocation cell-by-cell (async)**

POST**{{HostName}}**/blk/async/v1/hotels/**{{HotelId}}**/blocks/**{{BlockId}}**/allocation

Body:

```
{
    "criteria": {
        "hotelId": "{{HotelId}}",
        "blockId": {
            "type": "Block",
            "idContext": "OPERA",
            "id": "{{BlockId}}"
        },
        "allocationRoomTypes": [
            {
                "allocationGridDates": [
                    {
                        "roomAllocationInfo": [
                            {
                                "inventory": {
                                    "forceOverbook": false
                                },
                                "rate": {
                                    "onePerson": "240",
                                    "twoPerson": "240",
                                    "threePerson": "260"
                                },
                                "start": "2023-12-01",
                                "end": "2023-12-01"
                            },
                            {
                                "inventory": {
                                    "forceOverbook": false
                                },
                                "rate": {
                                    "onePerson": "200",
                                    "twoPerson": "220"
                                },
                                "start": "2023-12-02",
                                "end": "2023-12-02"
                            }
                        ],
                        "allocation": "RATES"
                    },
                    {
                        "roomAllocationInfo": [
                            {
                                "inventory": {
                                    "forceOverbook": false,
                                    "onePerson": "4",
                                    "twoPerson": "4"
                                },
                                "rate": {},
                                "start": "2023-12-01",
                                "end": "2023-12-01"
                            },
                            {
                                "inventory": {
                                    "forceOverbook": false,
                                    "onePerson": "3",
                                    "twoPerson": "3"
                                },
                                "rate": {},
                                "start": "2023-12-02",
                                "end": "2023-12-02"
                            }
                        ],
                        "allocation": "INITIAL"
                    }
                ],
                "roomType": "EXEC"
            }
        ],
        "genericRoomType": false
    }
}
```

**202 Response**: Accepted

Retrieve Process ID from RS Headers - Location and append it into the next calls URL

HEAD**{{HostName}}**/blk/async/v1/hotels/**{{HotelId}}**/blocks/**{{BlockId}}**/allocation/1134ad3c-95fa-d7d8-e063-666a040a111d

**201 Response**: Created

Retrieve Process ID from RS Headers - Location and append it into the next calls URL

GET**{{HostName}}**/blk/async/v1/hotels/**{{HotelId}}**/blocks/**{{BlockId}}**/allocation/1134accb-f697-56b0-e063-666a040ae3b4

**200 Response**: OK

Body:

```
{
    "status": "Success"
}
```

*Sample for* **postBlockAllocationRanges (async)**

**Note**

Although there is a start and end date for each grid cell, in this call, the start and end date in each collection must be the same (and the collection must be repeated for any additional dates).

This call is for a block that is not open for pickup. Note the element: "allocation": "INITIAL." When running this RQ for an open-for-pickup call, the elements must be :"allocation": "ACTUAL."

**Post Block Allocation Ranges (async)**

POST**{{HostName}}**/blk/async/v1/hotels/**{{HotelId}}**/blocks/**{{BlockId}}**/allocationRanges

Body:

```
{
    "blockAllocationRange": {
        "blockId": {
            "type": "Block",
            "idContext": "OPERA",
            "id": "{{BlockId}}"
        },
        "hotelId": "{{HotelId}} ",
        "roomTypes": [
            "DLXQ"
        ],
        "beginDate": "2023-12-01",
        "endDate": "2023-12-02",
        "allocationType": "Actual",
        "incrementFlag": false,
        "blockInventory": {
            "onePerson": "8",
            "twoPerson": "3"
        },
        "blockRates": {
            "onePerson": "240",
            "twoPerson": "280"
        },
        "includedDays": "1111111",
        "rangeMode": "Core",
        "genericRoomType": false
    }
}
```

**202 Response**: Accepted

Retrieve Process ID from RS Headers - Location and append it into the next calls URL

HEAD**{{HostName}}**/blk/async/v1/hotels/**{{HotelId}}**/blocks/**{{BlockId}}**/allocationRanges/1134ad3c-95fa-d7d8-e063-666a040a111d

201 Response: Created

Retrieve Process ID from RS Headers - Location and append it into the next calls URL

GET**{{HostName}}**/blk/async/v1/hotels/**{{HotelId}}**/blocks/**{{BlockId}}**/allocationRanges/1134accb-f697-56b0-e063-666a040ae3b4

**200 Response**: OK

Body:

```
{
    "status": "Success"
}
```

### References

- For more information, review the Anti-Patterns chapter in the OHIP user guide to learn more about functional and technical anti-patterns.
- Fore more information, refer to the Blocks introduction topic in the OPERA Cloud Services user guide.

7

# Edit Event Number of Attendees or Event Description

### Business Context

The Events feature is designed to manage any kind of catering activity. Events can be as simple as a one-hour reception or more complex, such as a three-day business meeting with meals, breaks, and specific meeting rooms with setup and resource requirements. Any group function can be an event.

Events require many levels of organization including: booking a function space for the event, creating a food and beverage menu, and making sure all other resources needed for the event are available. Sometimes, you need to make changes to existing events; those changes could be updating the event description, changing the number of attendees, changing the price of an item, and so on.

### Prerequisites

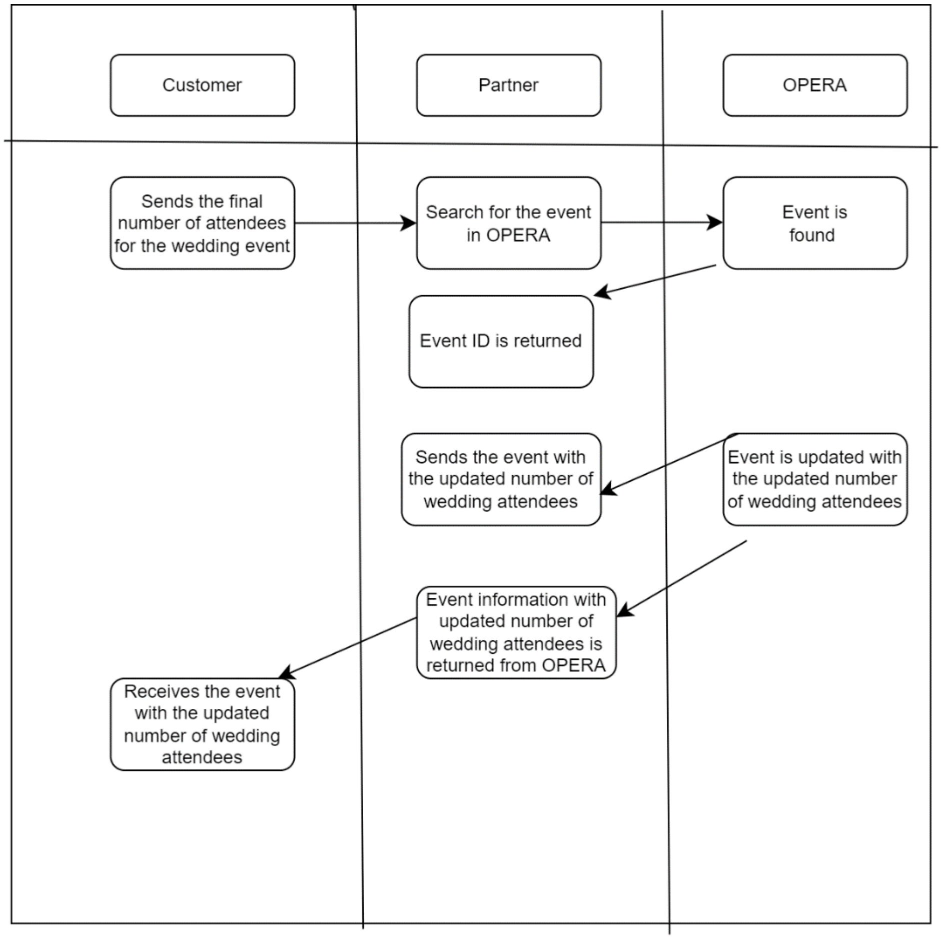
The below workflow diagram explains typical use cases for updating an existing event.

Customer — The groom wants to send the final number of attendees for his wedding this upcoming weekend.

Partner — The partner sends the request to update the number of wedding attendees so the customer (groom) can receive the response with the updated information.

OPERA Cloud — The application searches for the event ID, updates the current information with the correct number of attendees, and sends the updated event with the final number of wedding attendees.

### Figure 7-1 Workflow Diagram


Customer
Partner
OPERA
Sends the final number of attendees for the wedding event
Search for the event in OPERA
Event is found
Event ID is returned
Sends the event with the updated number of wedding attendees
Event is updated with the updated number of wedding attendees
Event information with updated number of wedding attendees is returned from OPERA
Receives the event with the updated number of wedding attendees

### Table 7-1 Required Software Tools

| Tool | Description | Links |
|---|---|---|
| Postman | Postman is an API Platform that allows you to design, create and test API's.<br>Use Postman to send API requests or to use the Postman collections provided. | Using the Oracle Hospitality APIs<br>Tool |

### Table 7-2 Current Versions Required

| OPERA Cloud Platform Module | Description | Minimum Version |
|---|---|---|
| OPERA Cloud Services | The customer must have a subscription to OPERA Cloud Foundation | |
| Oracle Hospitality Integration Platform (OHIP) | For customers, OHIP is included in the subscription to OPERA Cloud Foundation.<br>Partners need a subscription to the Oracle Hospitality Integration Cloud Service. | |

### Configuration of OPERA Controls

To activate the Create Block functionality, you must ensure the following OPERA Controls are active in the target property. You can use the following API operations to validate the settings:

**Ensure that application function EVENTS - CATERING EVENTS is active**

RQ: ent/config/v1/settings?hotelId={{HotelId}}&parameterNameWildCard=CATERING EVENTS

Mandatory Fields - Available Values:

**Function Space Configuration**

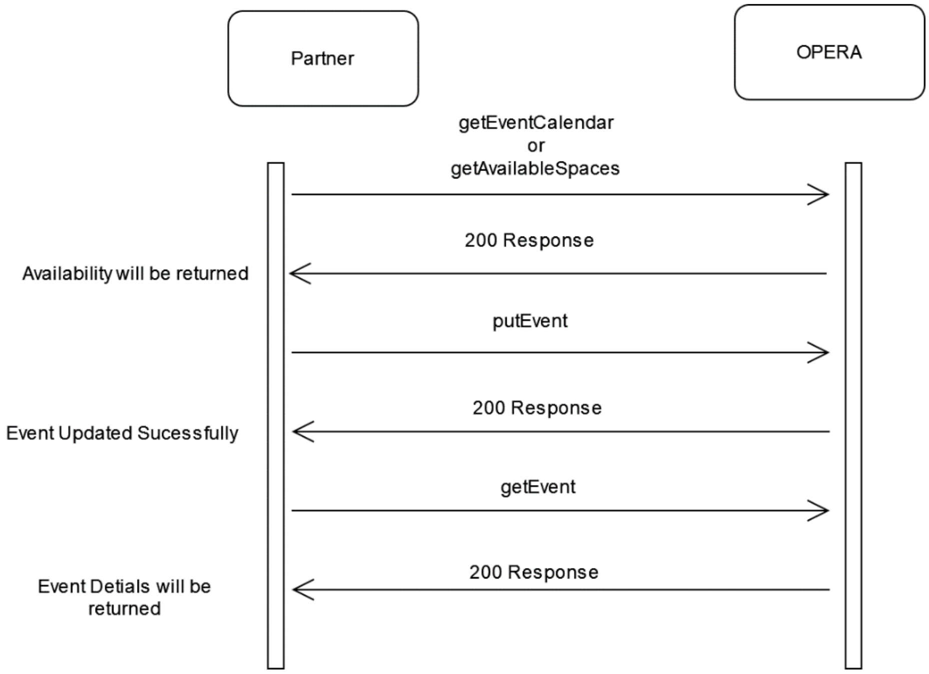
> **Note**
>
> This will return the codes available for Rental Code and Setup Styles which are required fields when creating an event with function space.

RQ: evm/config/v1/hotels/{{HotelId}}/functionSpaces/{{FunctionSpaceCode}}/configurations

### Workflow

Step by step solution including:

**Figure 7-2 The sequence of API calls (in any direction)**



### Description of Steps

A typical process to add or change a function space on an existing event is as follows:

1. Search for function space availability for the event date(s) using **getEventCalendar** or **getAvailableSpaces**.

2. Search for the event by block Id. If you know the internal event ID, then omit this step.
3. Add or Update the function space assigned to the event using **putEvent**.
    a. Ensure the mandatory fields Event Type, Event Status Code, Start Time, End Time, and Expected Attendees are sent with valid values.
    b. If you are intending to assign a function space, ensure that the elements for Rental Code and Setup Styles are present and contain valid values.

**Sample calls**: Sample API calls for steps described in the diagram.

*Sample for* **getEventCalendar**

**Check Function Space Availability**

**getEventCalendar**

GET **{{HostName}}**/evm/v1/hotels/**{{HotelId}}**/eventcalendar?eventEndDate=2025-04-20&isSortDescending=false&showAvailableRooms=true&display=ALL&eventLabel=Booking&sortingBy=Name&eventStartDate=2025-04-20

**200 response:**

```
{
    "eventCalendarDetails": {
        "eventSpace": [
            {
                "description": "Chesapeake A",
                "eventSpaceAttributes": {
                    "spaceArea": {
                        "area": 744,
                        "unit": "SqFeet"
                    },
                    "shareable": false,
                    "comboSpace": false,
                    "alternate": false,
                    "maximumOccupancy": 95,
                    "comboElement": true,
                    "comboRoom": "CHER"
                },
                "hotelId": "BHOTEL",
                "roomId": "CHEA",
                "name": "CHEA",
                "building": "MAIN",
                "roomOrder": 1
            }
        ],
        "eventCodes": [],
        "sellMessages": {},
        "totalRooms": 1
    },
    "links": [],
    "warnings": []
}
```

*Sample for* **getAvailableSpaces**

**Check Function Space Availability**

**getAvailableSpaces**

**GET {{HostName}}**/lov/v1/listOfValues/AvailableSpaces?parameterName=Event_Start&parameterName=Event_End&parameterName=HotelCode&includeInactiveFlag=false&parameterValue=2025-04-20+18:00:00.0&parameterValue=2025-04-20+19:00:00.0&parameterValue=**{{HotelId}}**

**200 response:**

```
{
    "listOfValues": {
        "items": [
            {
                "flexfields": [
                    {


            {
                "flexfields": [
                    {
                        "parameterName": "AvailableYn",
                        "parameterValue": "Y"
                    },
                    {
                        "parameterName": "SetupCode"
                    },
                    {
                        "parameterName": "OccMin"
                    },
                    {
                        "parameterName": "OccMax"
                    },
                    {
                        "parameterName": "SetupTime"
                    },
                    {
                        "parameterName": "SetdownTime"
                    },
                    {
                        "parameterName": "RateCode"
                    },
                    {
                        "parameterName": "Rate"
                    },
                    {
                        "parameterName": "ShareableYn",
                        "parameterValue": "N"
                    },
                    {
                        "parameterName": "ForceAlternateYn",
                        "parameterValue": "N"
                    }
                ],
                "code": "CHEA",
                "name": "Chesapeake A",
                "description": "Chesapeake A",
                "active": true
```

```
 }
```

*Sample for* **putEvent**

**Create Event with Space**

PUT**{{HostName}}**/evm/v1/events

Body:

```
{
    "eventDetails": [
        {
            "eventPrimaryInfo": {
                "eventId": {
                    "type": "EventId",
                    "idContext": "OPERA",
                    "id": "641634"
                },
                "hotelId": "BHOTEL"
            },
            "eventDetail": {
                "eventName": {
                    "defaultText": "Welcome Reception"
                },
                "eventType": "REC",
                "eventTimeSpan": {
                    "startDateTime": "2025-04-20 18:00:00.0",
                    "endDateTime": "2025-04-20 19:00:00.0"
                },
                "eventStatus": {
                    "status": {
                        "code": "INQ"
                    }
                },
                "attendees": {
                    "expected": "30"
                },
                "doorcard": "Do Not Update KD API Block",
                "notMoveable": false,
                "loudEvent": false,
                "displayDoorcard": true,
                "includeSpaceInPackage": false
            },
            "processInstructions": {
                "overrideCapacity": false,
                "overbook": false,
                "overrideMaxgroups": false,
                "warnFunctionSpace": true
            },
            "eventBlockInfo": {
                "blockIdList": [
                    {
                        "type": "Block",
                        "id": "919698"
                    }
                ],
                "hotelId": "BHOTEL",
                "markAsRecentlyAccessed": false
            },
            "functionSpaceInformation": {
                "functionSpaceCode": "CHEA",
                "functionSpaceDescription": "Chesapeake A",
                "setupCode": "BQ10",
                "setupTime": "30",
                "setdownTime": "30",
                "rentalCode": "CUSTOM",
                "rentalAmount": {
                    "amount": "500"
                }
            }
            }
        }
    ]
}
```

**200 Response**: Created

*Sample for* **getEvent**

**Fetch Event**

GET**{{HostName}}**/evm/v1/events?hotelId={{HotelId}}&eventId={{EventId}}

```
{
    "eventSummaries": {
        "eventSummary": [
            {
                "eventId": {
                    "type": "EventId",
                    "idContext": "OPERA",
                    "id": "641634"
                },
                "blockIdList": [
                    {
                        "id": "919698",
                        "type": "Block"
                    }
                ],
                "blockName": "Do Not Update KD API Block",
                "attendees": 30,
                "functionSpaceDetails": {
                    "code": "CHEA",
                    "description": "Chesapeake A"
                },
                "rentalAmount": 500,
                "functionSpaceSetup": {
                    "code": "BQ10",
                    "description": "Banquets 10 Chairs"
                },
                "eventStatus": {
                    "status": {
                        "code": "INQ",
                        "description": "Inquiry Non Deduct"
                    },
                    "color": "Blue"
                },
                "cateringStatusDeductInventory": false,
                "cateringStatusReturnToInventory": false,
                "setupTime": 30,
                "tearDownTime": 30,
                "masterEvent": false,
                "subEvent": false,
                "packageEvent": false,
                "alternateEvent": false,
                "displayDoorCard": true,
                "doorCard": "Do Not Update KD API Block",
                "excludeFromForecast": false,
                "useForecastRevenue": true,
                "minimumOccupancy": 1,
                "maximumOccupancy": 60,
                "loudEvent": false,
                "rentalCode": "CUSTOM",
                "eventShared": false,
                "shareable": false,
                "includeSpaceInPackage": false,
                "hotelId": "BHOTEL",
                "blockHotelCode": "BHOTEL",
                "hasPostings": false,
                "blockHasPostings": false,
                "hasPartialPostings": false,
                "allPosted": false,
                "nothingPosted": false,
                "eventStartDate": "2025-04-20 18:00:00.0",
                "eventEndDate": "2025-04-20 19:00:00.0",
                "type": "REC",
                "eventName": "Welcome Reception",
                "cateringCurrency": "USD",
                "cateringQuotedCurrency": "USD"
            }
        ],
        "totalPages": 1,
        "offset": 200,
        "limit": 200,
        "hasMore": false,
        "totalResults": 1
    },
    "links": []
}
```

**200 Response**: OK

### References

- For more information, review the Anti-Patterns chapter in the OHIP user guide to learn more about functional and technical anti-patterns.

- Fore more information, refer to the Events topic in the OPERA Cloud Services user guide.

8

# Add or Change a Function Space for an Existing Event

**Business Context**

Sales managers might create events and later assign the function space based on the property availability or the client's preferences and requests. Another instance might be a change in the guaranteed number of attendees that requires an update to either a larger or smaller space. Whatever the case, you need the ability to assign and/or change a function space using the REST APIs.

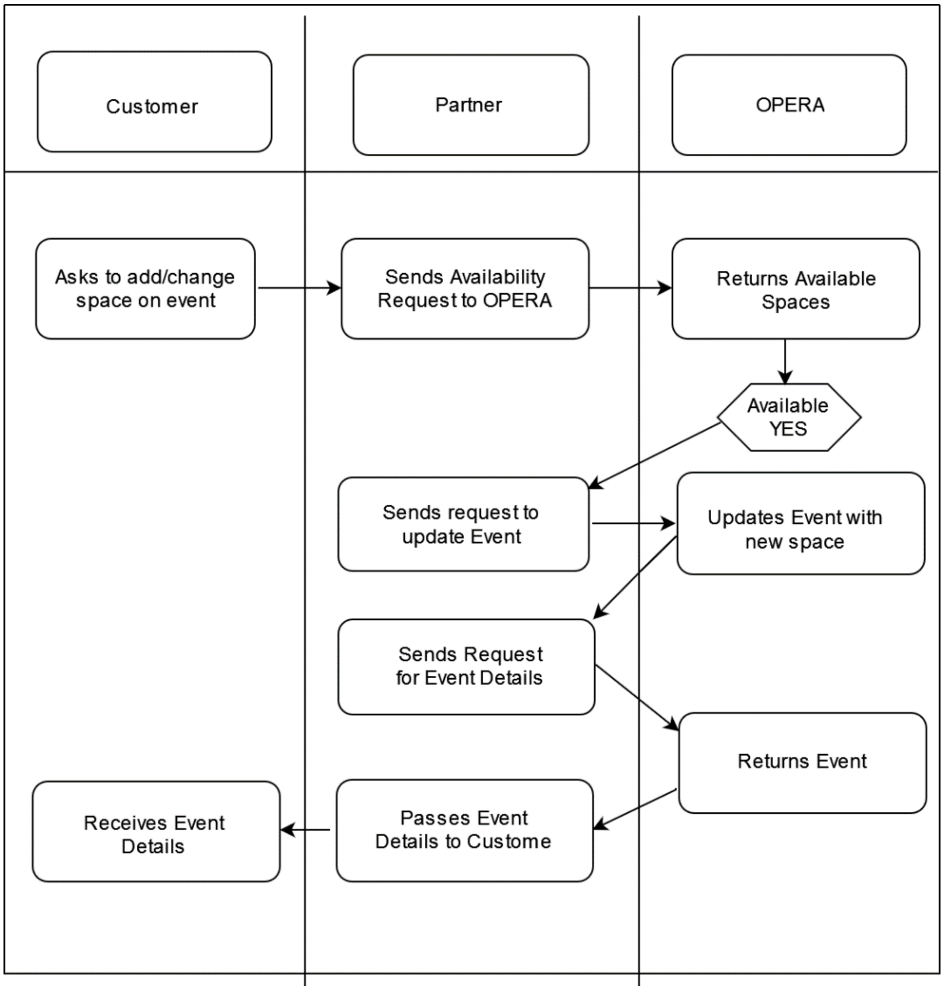
This below workflow diagram explains typical use cases for adding or changing a function space on an event.

Customer — The customer has chosen to have a larger space for his meeting based on the final number of attendees for his events.

Partner — The partner must check availability based on the response to book the larger space.

OPERA Cloud — The application checks for availability, returns the response, and possibly books the new space for the partner so the customer receives the new booking information.

**Figure 8-1 Workflow Diagram**


Customer
Partner
OPERA
Asks to add/change space on event
Sends Availability Request to OPERA
Returns Available Spaces
Available YES
Sends request to update Event
Updates Event with new space
Sends Request for Event Details
Returns Event
Receives Event Details
Passes Event Details to Custome

## Prerequisites

### Table 8-1 Required Software Tools

| Tool | Description | Links |
|---|---|---|
| Postman | Postman is an API Platform that allows you to design, create and test API's.<br>Use Postman to send API requests or to use the Postman collections provided. | Using the Oracle Hospitality APIs<br>Tool |

### Table 8-2 Current Versions Required

| OPERA Cloud Platform Module | Description | Minimum Version |
|---|---|---|
| OPERA Cloud Services | The customer must have a subscription to OPERA Cloud Foundation | |

Table 8-2 (Cont.) Current Versions Required

| OPERA Cloud Platform Module | Description | Minimum Version |
|---|---|---|
| Oracle Hospitality Integration Platform (OHIP) | For customers, OHIP is included in the subscription to OPERA Cloud Foundation.<br>Partners need a subscription to the Oracle Hospitality Integration Cloud Service. | |

### Configuration of OPERA Controls

To activate the Create Block functionality, you must ensure the following OPERA Controls are active in the target property. You can use the following API operations to validate the settings:

**Ensure that application function EVENTS - CATERING EVENTS is active**

RQ: ent/config/v1/settings?hotelId={{HotelId}}&parameterNameWildCard=CATERING EVENTS

Mandatory Fields - Available Values:

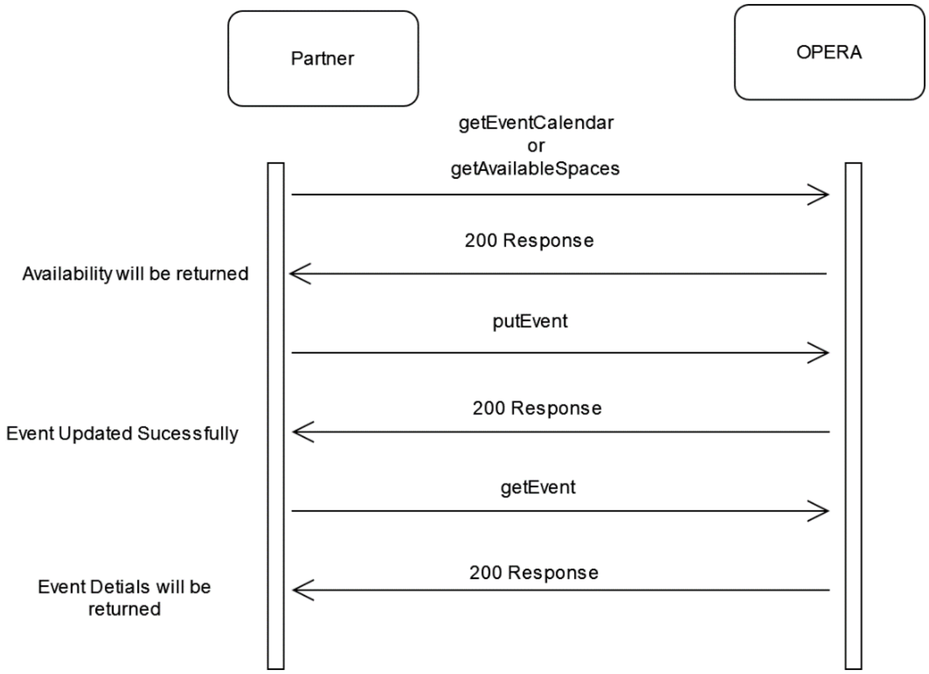
**Function Space Configuration**

> **Note**
>
> This returns the codes available for Rental Code and Setup Styles, which are required fields when creating an event with function space.
>
> RQ: evm/config/v1/hotels/{{HotelId}}/functionSpaces/{{FunctionSpaceCode}}/configurations

### Workflow

Step by step solution including:

**Figure 8-2 The sequence of API calls (in any direction)**



### Description of Steps

A typical process to add or change function space on an existing event is as follows:

1. Search for function space availability for the event date(s) using **getEventCalendar** or **getAvailableSpaces**.
2. Search for the event by block Id. If you know the internal event ID, then omit this step.
3. Add or Update the function space assigned to the event using **putEvent**.
   a. Ensure the mandatory fields Event Type, Event Status Code, Start Time, End Time, and Expected Attendees are sent with valid values.
   b. If you are intending to assign a function space, ensure that the elements for Rental Code and Setup Styles are present and contain valid values.

**Sample calls**: Sample API calls for the steps described in the diagram.

*Sample for* **getEventCalendar**

**Check Function Space Availability**

**getEventCalendar**

GET **{{HostName}}**/evm/v1/hotels/**{{HotelId}}**/eventcalendar?eventEndDate=2025-04-20&isSortDescending=false&showAvailableRooms=true&display=ALL&eventLabel=Booking&sortingBy=Name&eventStartDate=2025-04-20

**200 response:**

```
{
    "eventCalendarDetails": {
        "eventSpace": [
            {
                "description": "Chesapeake A",
                "eventSpaceAttributes": {
                    "spaceArea": {
                        "area": 744,
                        "unit": "SqFeet"
                    },
                    "shareable": false,
                    "comboSpace": false,
                    "alternate": false,
                    "maximumOccupancy": 95,
                    "comboElement": true,
                    "comboRoom": "CHER"
                },
                "hotelId": "BHOTEL",
                "roomId": "CHEA",
                "name": "CHEA",
                "building": "MAIN",
                "roomOrder": 1
            }
        ],
        "eventCodes": [],
        "sellMessages": {},
        "totalRooms": 1
    },
    "links": [],
    "warnings": []
}
```

*Sample for* **getAvailableSpaces**

**Check Function Space Availability**

**getAvailableSpaces**

**GET {{HostName}}**/lov/v1/listOfValues/AvailableSpaces?parameterName=Event_Start&parameterName=Event_End&parameterName=HotelCode&includeInactiveFlag=false&parameterValue=2025-04-20+18:00:00.0&parameterValue=2025-04-20+19:00:00.0&parameterValue=**{{HotelId}}**

**200 response:**

```
{
    "listOfValues": {
        "items": [
            {
                "flexfields": [
                    {


            {
                "flexfields": [
                    {
                        "parameterName": "AvailableYn",
                        "parameterValue": "Y"
                    },
                    {
                        "parameterName": "SetupCode"
                    },
                    {
                        "parameterName": "OccMin"
                    },
                    {
                        "parameterName": "OccMax"
                    },
                    {
                        "parameterName": "SetupTime"
                    },
                    {
                        "parameterName": "SetdownTime"
                    },
                    {
                        "parameterName": "RateCode"
                    },
                    {
                        "parameterName": "Rate"
                    },
                    {
                        "parameterName": "ShareableYn",
                        "parameterValue": "N"
                    },
                    {
                        "parameterName": "ForceAlternateYn",
                        "parameterValue": "N"
                    }
                ],
                "code": "CHEA",
                "name": "Chesapeake A",
                "description": "Chesapeake A",
                "active": true

 }
```

*Sample for* **putEvent**

**Create Event with Space**

PUT**{{HostName}}**/evm/v1/events

Body:

```
{
    "eventDetails": [
        {
            "eventPrimaryInfo": {
                "eventId": {
                    "type": "EventId",
                    "idContext": "OPERA",
                    "id": "641634"
                },
                "hotelId": "BHOTEL"
            },
            "eventDetail": {
                "eventName": {
                    "defaultText": "Welcome Reception"
                },
                "eventType": "REC",
                "eventTimeSpan": {
                    "startDateTime": "2025-04-20 18:00:00.0",
                    "endDateTime": "2025-04-20 19:00:00.0"
                },
                "eventStatus": {
                    "status": {
                        "code": "INQ"
                    }
                },
                "attendees": {
                    "expected": "30"
                },
                "doorcard": "Do Not Update KD API Block",
                "notMoveable": false,
                "loudEvent": false,
                "displayDoorcard": true,
                "includeSpaceInPackage": false
            },
            "processInstructions": {
                "overrideCapacity": false,
                "overbook": false,
                "overrideMaxgroups": false,
                "warnFunctionSpace": true
            },
            "eventBlockInfo": {
                "blockIdList": [
                    {
                        "type": "Block",
                        "id": "919698"
                    }
                ],
                "hotelId": "BHOTEL",
                "markAsRecentlyAccessed": false
            },
            "functionSpaceInformation": {
                "functionSpaceCode": "CHEA",
                "functionSpaceDescription": "Chesapeake A",
                "setupCode": "BQ10",
                "setupTime": "30",
                "setdownTime": "30",
                "rentalCode": "CUSTOM",
                "rentalAmount": {
                    "amount": "500"
                }
            }
            }
        }
    ]
}
```

**200 Response**: Created

*Sample for* **getEvent**

### Fetch Event

GET**{{HostName}}**/evm/v1/events?hotelId={{HotelId}}&eventId={{EventId}}

```
{
    "eventSummaries": {
        "eventSummary": [
            {
                "eventId": {
                    "type": "EventId",
                    "idContext": "OPERA",
                    "id": "641634"
                },
                "blockIdList": [
                    {
                        "id": "919698",
                        "type": "Block"
                    }
                ],
                "blockName": "Do Not Update KD API Block",
                "attendees": 30,
                "functionSpaceDetails": {
                    "code": "CHEA",
                    "description": "Chesapeake A"
                },
                "rentalAmount": 500,
                "functionSpaceSetup": {
                    "code": "BQ10",
                    "description": "Banquets 10 Chairs"
                },
                "eventStatus": {
                    "status": {
                        "code": "INQ",
                        "description": "Inquiry Non Deduct"
                    },
                    "color": "Blue"
                },
                "cateringStatusDeductInventory": false,
                "cateringStatusReturnToInventory": false,
                "setupTime": 30,
                "tearDownTime": 30,
                "masterEvent": false,
                "subEvent": false,
                "packageEvent": false,
                "alternateEvent": false,
                "displayDoorCard": true,
                "doorCard": "Do Not Update KD API Block",
                "excludeFromForecast": false,
                "useForecastRevenue": true,
                "minimumOccupancy": 1,
                "maximumOccupancy": 60,
                "loudEvent": false,
                "rentalCode": "CUSTOM",
                "eventShared": false,
                "shareable": false,
                "includeSpaceInPackage": false,
                "hotelId": "BHOTEL",
                "blockHotelCode": "BHOTEL",
                "hasPostings": false,
                "blockHasPostings": false,
                "hasPartialPostings": false,
                "allPosted": false,
                "nothingPosted": false,
                "eventStartDate": "2025-04-20 18:00:00.0",
                "eventEndDate": "2025-04-20 19:00:00.0",
                "type": "REC",
                "eventName": "Welcome Reception",
                "cateringCurrency": "USD",
                "cateringQuotedCurrency": "USD"
            }
        ],
        "totalPages": 1,
        "offset": 200,
        "limit": 200,
        "hasMore": false,
        "totalResults": 1
    },
    "links": []
}
```

**200 Response**: OK

### References

- For more information, review the Anti-Patterns chapter in the OHIP user guide to learn more about functional and technical anti-patterns.

9

# Add a Catering Menu to an Event

**Business Context**

Events require many levels of organization including: booking a function space for the event, creating a food and beverage menu, and making sure all other resources needed for the event are available. To add a Catering Menu to an event, you must search for an event first. This topic describes the steps to search for an event using the REST APIs, and describes how to add a Catering Menu.

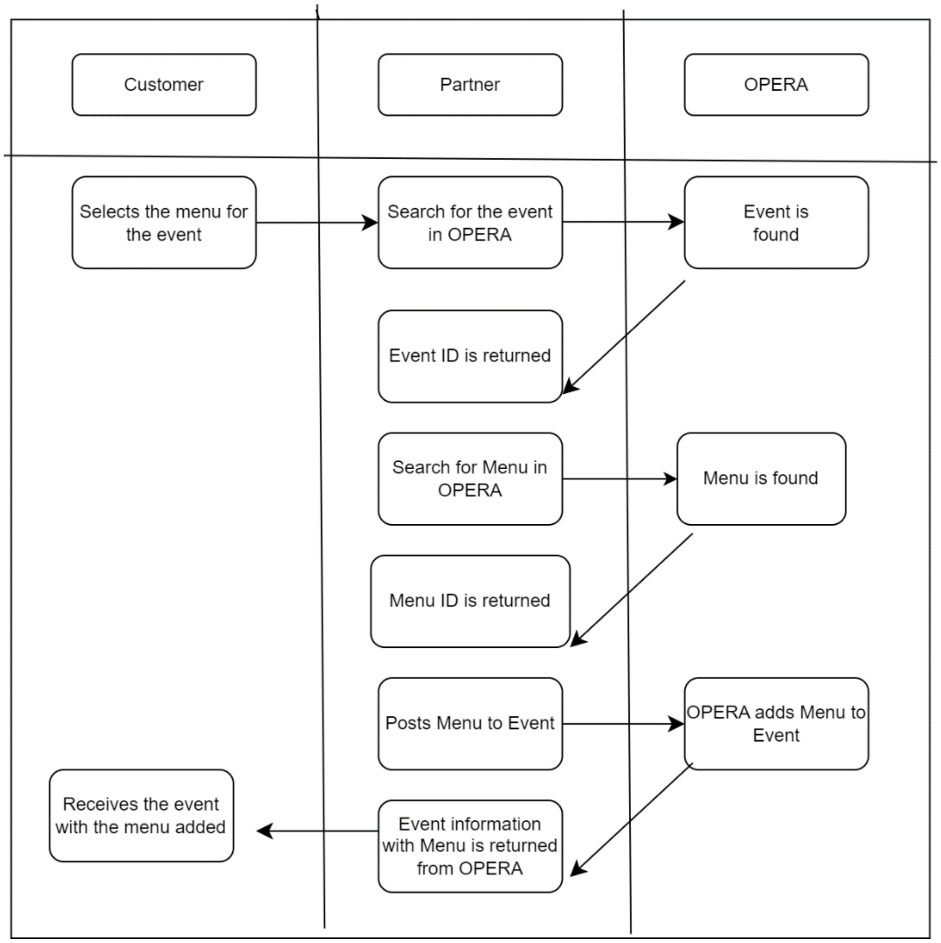
The below workflow diagram explains the typical use cases for adding a Catering Menu to an event.

Customer — The meeting planner who has selected a Catering Menu for the VIP group coming to the property for a one day meeting.

Partner — The partner communicates with the property the customer's choice so the Catering Menu can be added to the event.

OPERA Cloud — The application looks for the specific event, returns the event ID, adds the selected Catering Menu, and returns the event with the Catering Menu added.

### Figure 9-1 Workflow Diagram



## Prerequisites

### Table 9-1 Required Software Tools

| Tool | Description | Links |
|---|---|---|
| Postman | Postman is an API Platform that allows you to design, create and test API's.<br>Use Postman to send API requests or to use the Postman collections provided. | Using the Oracle Hospitality APIs<br>Tool |

### Table 9-2 Current Versions Required

| OPERA Cloud Platform Module | Description | Minimum Version |
|---|---|---|
| OPERA Cloud Services | The customer must have a subscription to OPERA Cloud Foundation | |
| Oracle Hospitality Integration Platform (OHIP) | For customers, OHIP is included in the subscription to OPERA Cloud Foundation.<br>Partners need a subscription to the Oracle Hospitality Integration Cloud Service. | |

### Configuration of OPERA Controls

To activate the Create Block functionality, you must ensure the following OPERA Controls are active in the target property. You can use the following API operations to validate the settings:

**Ensure that application function BLOCKS - BUSINESS BLOCK is active**

RQ: ent/config/v1/settings?hotelId={{HotelId}}&parameterNameWildCard=BUSINESS BLOCK

**Ensure that application function EVENTS - CATERING EVENTS is active**

RQ: ent/config/v1/settings?hotelId={{HotelId}}&parameterNameWildCard=CATERING EVENTS

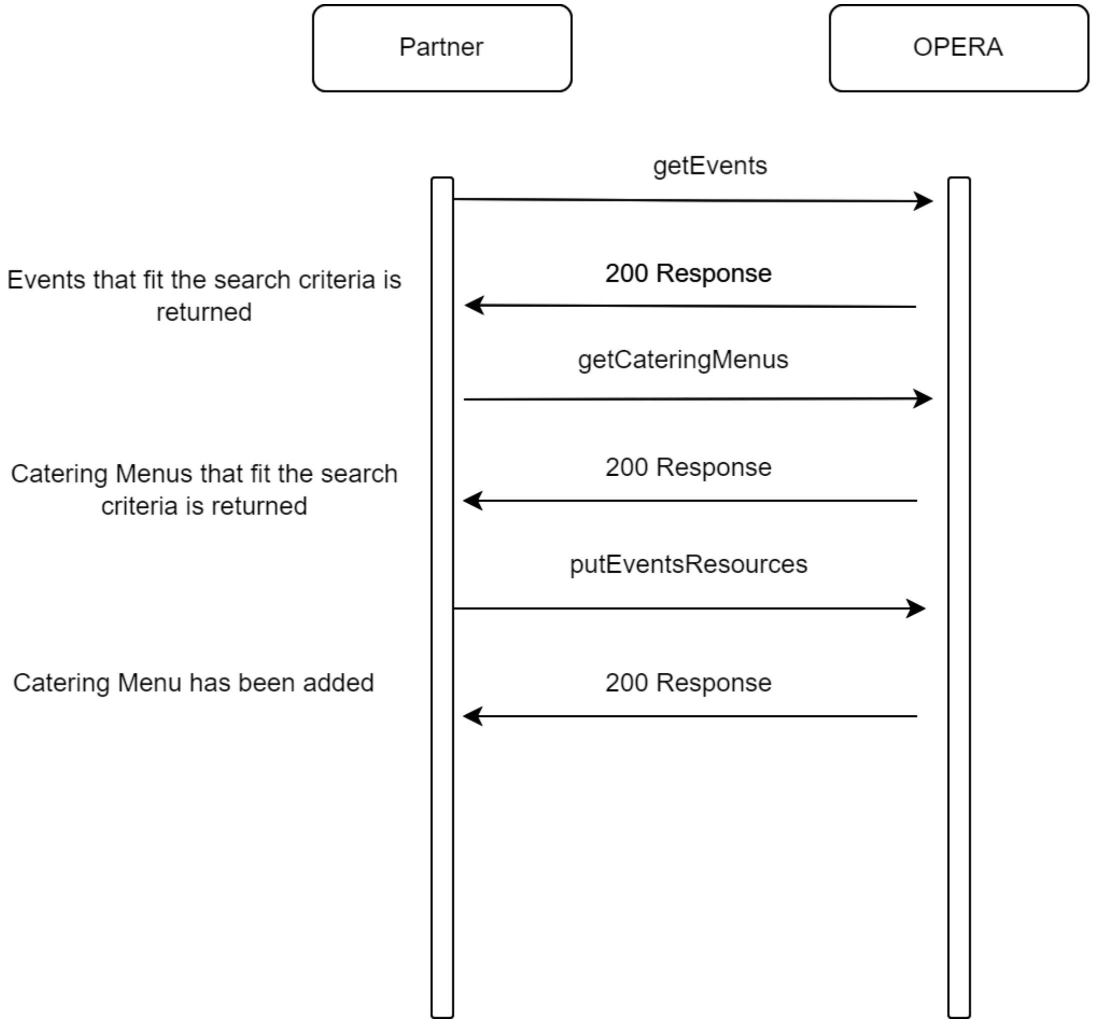
**Ensure that application function EVENTS - EVENT RESOURCES is active**

RQ: ent/config/v1/settings?hotelId={{HotelId}}&parameterNameWildCard=EVENT RESOURCES

### Workflow

Step by step solution including:

**Figure 9-2 The sequence of API calls (in any direction)**


Partner
OPERA
getEvents
200 Response
Events that fit the search criteria is returned
getCateringMenus
200 Response
Catering Menus that fit the search criteria is returned
putEventsResources
Catering Menu has been added
200 Response

### Description of Steps

A typical process to add a menu to an event is as follows:

1. Search for events using **getEvents**
    a. Multiple search parameters are available for example dates, status, event type, and other parameters.
2. Search for a menu using **getCateringMenus**
    a. In this specific scenario, the customer has provided you the menu name, so the search has been narrowed by name.
3. Add the Catering Menu to the event using **putEventsResources**
    a. The Catering Menu is added as configured, pricing, number of attendees, and other details cannot be modified during this operation. If changes are required, see the appropriate section in this user guide.

**Sample calls**: Sample API calls for steps described in the diagram.

*Sample for* **getEvents**

**Search for Event**

GET**{{HostName}}**/evm/v1/events?eventStartDate=2023-12-12&eventEndDate=2023-12-12&eventStatusCodes=DEF&hotelIds={{HotelId}}&eventTypeCodes=MTG

*Sample Response*

```
{
    "eventSummaries": {
        "eventSummary": [
            {
                "eventId": {
                    "type": "EventId",
                    "idContext": "OPERA",
                    "id": "610375"
                },
                "blockIdList": [
                    {
                        "id": "875435",
                        "type": "Block"
                    }
                ],
                "blockName": "STEST - Wait 1",
                "attendees": 20,
                "functionSpaceDetails": {
                    "code": "ANNA",
                    "description": "Annapolis Room"
                },
                "functionSpaceSetup": {
                    "code": "BQ10",
                    "description": "Banquets 10 Chairs"
                },
                "eventStatus": {
                    "status": {
                        "code": "DEF",
                        "description": "Definite"
                    },
                    "color": "Red"
                },
                "cateringStatusDeductInventory": true,
                "cateringStatusReturnToInventory": false,
                "setupTime": 30,
                "tearDownTime": 30,
                "masterEvent": false,
                "subEvent": false,
                "packageEvent": false,
                "alternateEvent": false,
                "displayDoorCard": true,
                "doorCard": "STEST - Wait",
                "excludeFromForecast": false,
                "useForecastRevenue": true,
                "minimumOccupancy": 5,
                "maximumOccupancy": 60,
                "loudEvent": false,
                "rentalCode": "CUSTOM",
                "eventShared": false,
                "shareable": false,
                "includeSpaceInPackage": false,
                "hotelId": "BHOTEL",
                "blockHotelCode": "BHOTEL",
                "hasPostings": false,
                "blockHasPostings": false,
                "hasPartialPostings": false,
                "allPosted": false,
                "nothingPosted": false,
                "eventStartDate": "2023-12-12 09:00:00.0",
                "eventEndDate": "2023-12-12 17:00:00.0",
                "type": "MTG",
                "eventName": "Meeting Wait 1",
                "cateringCurrency": "USD",
                "cateringQuotedCurrency": "USD"
            }
        ],
        "totalPages": 1,
        "offset": 200,
        "limit": 200,
        "hasMore": false,
        "totalResults": 1
    },
    "links": []
}
```

*Sample for* **getCateringMenus**

**Search for Menu**

GET**{{HostName}}**/evm/config/v1/hotels/{{HotelId}}/cateringMenus?name=briyani&eventTypesCodes=BRK

Sample Response:

```
{
    "menuList": [
        {
            "id": {
                "id": "111276"
            },
            "classId": "53812",
            "hotelId": "BHOTEL",
            "info": {
                "className": "Buffet Breakfast Menus",
                "name": {
                    "defaultText": "Briyani",
                    "translatedTexts": []
                },
                "dietaryList": [],
                "eventTypes": [],
                "type": "Food",
                "quickInsertCode": "111276",
                "servingType": "Pp",
                "servingSize": "",
                "includedInMenu": false,
                "consumptionBased": false,
                "menusSold": 0,
                "webBookable": false,
                "inactive": false,
                "salesPrice": {
                    "amount": 33
                },
                "multiChoice": false
            }
        }
    ],
    "totalPages": 1,
    "offset": 20,
    "limit": 20,
    "hasMore": false,
    "totalResults": 1,
    "links": []
}
```

*Sample for* **putEventsResources**

**Add the Catering Menu**

PUT**{{HostName}}**/operarest/{schemamap}/evm/v1/eventsResources

Body:

```
{
    "eventDetails": [
        {
            "eventPrimaryInfo": {
                "eventId": {
                    "type": "EventId",
                    "idContext": "OPERA",
                    "id": "610375"
                },
                "hotelId": "BHOTEL"
            },
            "eventBlockInfo": {
                "blockIdList": [
                    {
                        "type": "Block",
                        "idContext": "OPERA",
                        "id": "875435"
                    }
                ],
                "markAsRecentlyAccessed": false
            }
        }
    ],
    "eventMenus": {
        "eventMenuInfo": [
            {
                "eventMenu": {
                    "basicInfo": {
                        "menuId": {
                            "type": "MenuId",
                            "idContext": "OPERA",
                            "id": "111276"
                        },
                        "hotelId": "BHOTEL",
                        "multiChoice": false
                    }
                }
            }
        ],
        "includedInPackage": false,
        "critical": false
    }
}
```

### Limitations and Constraints

The customer has to provide an event ID in order for the partner to find the correct event to add the Catering Menu. Likewise, the Catering Menu name and/or menu ID has to be provided to the partner in order to add the correct one to the event.

### References

- For more information, refer to the Events topic in the OPERA Cloud Services user guide.

# 10 Add an Item to an Event

**Business Context**

Events require many levels of organization including: booking a function space for the event, creating a food and beverage menu, and making sure all other resources needed for the event are available. A customer usually decides the menu for an event after the booking process has already occurred and communicates the choice to the partner a few weeks prior to the event happening. This topic describes the steps to search for an event using the REST APIs, and describes how to add an item to an event that has already been created in the application. An item could be audio visual equipment, such as an LCD projector or teleprompters, or it could be an additional setup, such as a stage and a podium.

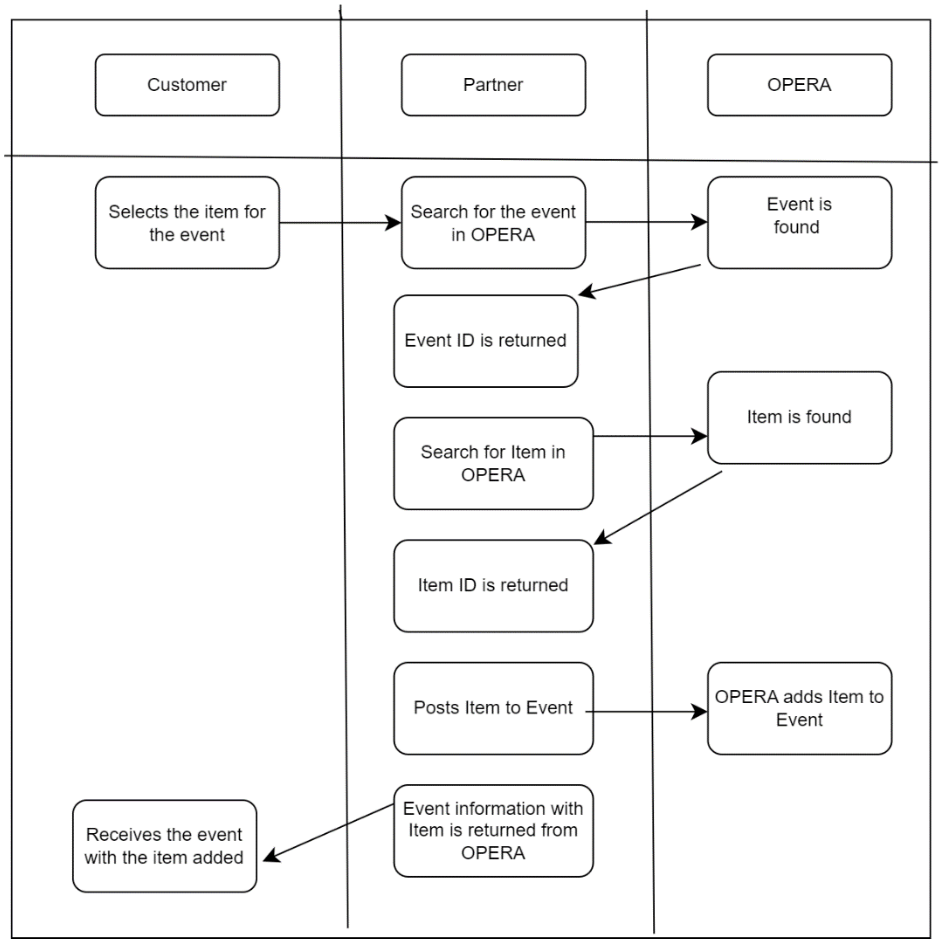
The below workflow diagram explains the typical use cases for adding a menu to an event.

Customer — The meeting planner has chosen the item to add to their meeting setup for next week.

Partner — The partner communicates the customer's selection to the property so the item can be added to the event.

OPERA Cloud — The application looks for the specific event, returns the event ID, adds the item selected, and returns the event with the item added.

**Figure 10-1    Workflow Diagram**



### Prerequisites

**Table 10-1    Required Software Tools**

| Tool | Description | Links |
|---|---|---|
| Postman | Postman is an API Platform that allows you to design, create and test API's.<br>Use Postman to send API requests or to use the Postman collections provided. | Using the Oracle Hospitality APIs<br>Tool |

**Table 10-2    Current Versions Required**

| OPERA Cloud Platform Module | Description | Minimum Version |
|---|---|---|
| OPERA Cloud Services | The customer must have a subscription to OPERA Cloud Foundation | |
| Oracle Hospitality Integration Platform (OHIP) | For customers, OHIP is included in the subscription to OPERA Cloud Foundation.<br>Partners need a subscription to the Oracle Hospitality Integration Cloud Service. | |

### Configuration of OPERA Controls

To activate the Create Block functionality, you must ensure the following OPERA Controls are active in the target property. You can use the following API operations to validate the settings:

**Ensure that application function BLOCKS - BUSINESS BLOCK is active**

RQ: ent/config/v1/settings?hotelId={{HotelId}}&parameterNameWildCard=BUSINESS BLOCK

**Ensure that application function EVENTS - CATERING EVENTS is active**

RQ: ent/config/v1/settings?hotelId={{HotelId}}&parameterNameWildCard=CATERING EVENTS

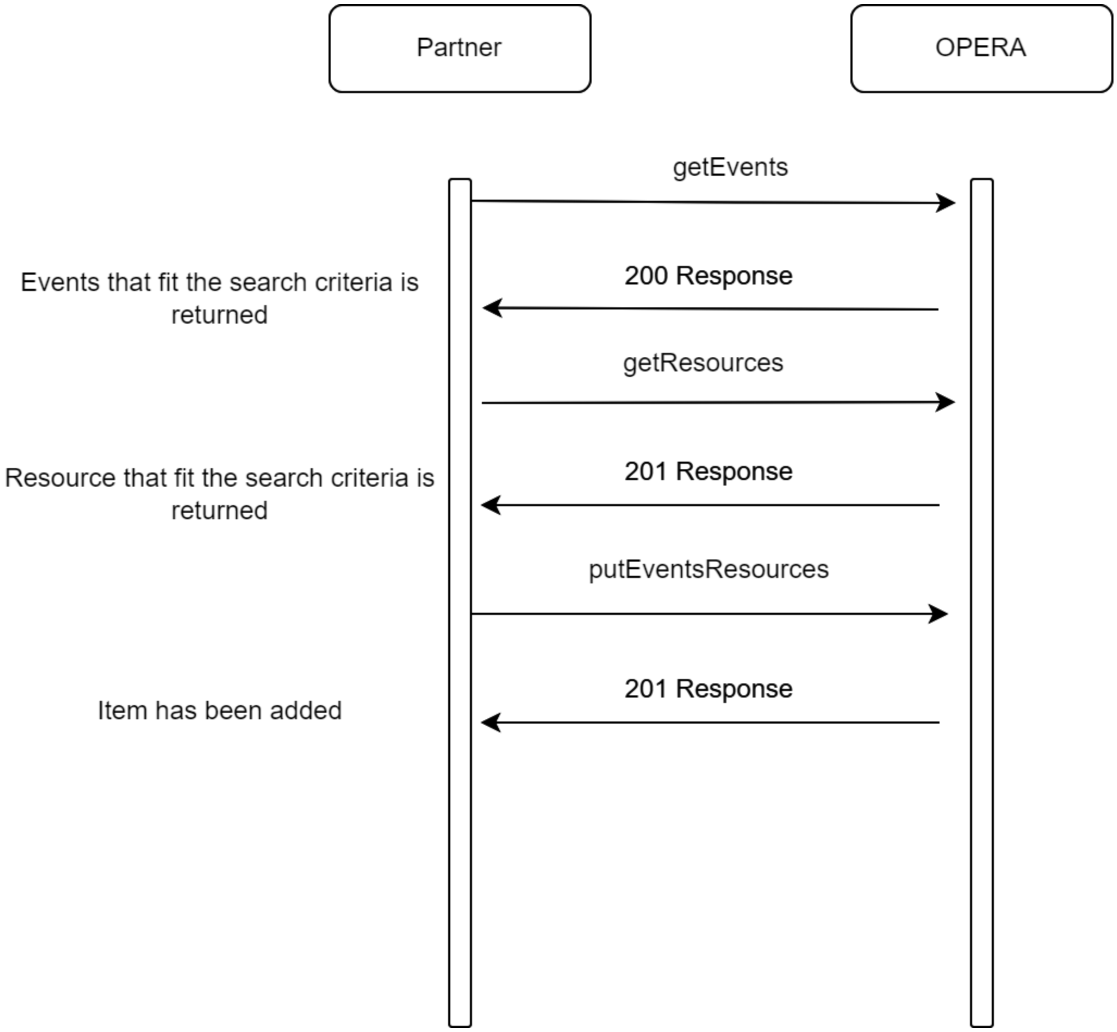
**Ensure that application function EVENTS - EVENT RESOURCES is active**

RQ: ent/config/v1/settings?hotelId={{HotelId}}&parameterNameWildCard=EVENT RESOURCES

### Workflow

Step by step solution including:

**Figure 10-2 The sequence of API calls (in any direction)**



### Description of Steps

A typical process to add a menu to an event is as follows:

1. Search for an event using **getEvents**.
    a. Multiple search parameters are available, for example, dates, status, event type, and other parameters.
2. Search for an Item using **getInventoryItem**.
    a. In this specific scenario, the customer has provided the name of the item to add to the event.
3. Add the Item to the event using **putEventsResources**.
    a. The Item is added as configured. Pricing, number of items, and other details cannot be modified during this operation. If changes are required, see the appropriate section in this user guide.

**Sample calls**: Sample API calls for steps described in the diagram.

*Sample for* **getEvents**

**Search for Event**

GET **{{HostName}}**/evm/v1/events?eventStartDate=2023-12-12&eventEndDate=2023-12-12&eventStatusCodes=DEF&hotelIds={{HotelId}}&eventTypeCodes=MTG

*Sample Response*

```
{
    "eventSummaries": {
        "eventSummary": [
            {
                "eventId": {
                    "type": "EventId",
                    "idContext": "OPERA",
                    "id": "610375"
                },
                "blockIdList": [
                    {
                        "id": "875435",
                        "type": "Block"
                    }
                ],
                "blockName": "STEST - Wait 1",
                "attendees": 20,
                "functionSpaceDetails": {
                    "code": "ANNA",
                    "description": "Annapolis Room"
                },
                "functionSpaceSetup": {
                    "code": "BQ10",
                    "description": "Banquets 10 Chairs"
                },
                "eventStatus": {
                    "status": {
                        "code": "DEF",
                        "description": "Definite"
                    },
                    "color": "Red"
                },
                "cateringStatusDeductInventory": true,
                "cateringStatusReturnToInventory": false,
                "setupTime": 30,
                "tearDownTime": 30,
                "masterEvent": false,
                "subEvent": false,
                "packageEvent": false,
                "alternateEvent": false,
                "displayDoorCard": true,
                "doorCard": "STEST - Wait",
                "excludeFromForecast": false,
                "useForecastRevenue": true,
                "minimumOccupancy": 5,
                "maximumOccupancy": 60,
                "loudEvent": false,
                "rentalCode": "CUSTOM",
                "eventShared": false,
                "shareable": false,
                "includeSpaceInPackage": false,
                "hotelId": "BHOTEL",
                "blockHotelCode": "BHOTEL",
                "hasPostings": false,
                "blockHasPostings": false,
                "hasPartialPostings": false,
                "allPosted": false,
                "nothingPosted": false,
                "eventStartDate": "2023-12-12 09:00:00.0",
                "eventEndDate": "2023-12-12 17:00:00.0",
                "type": "MTG",
                "eventName": "Meeting Wait 1",
                "cateringCurrency": "USD",
                "cateringQuotedCurrency": "USD",
                "eventIndicators": [
                    {
                        "indicatorName": "REVENUE",
                        "count": 3
                    }
                ]
            }
        ],
        "totalPages": 1,
        "offset": 100,
        "limit": 100,
        "hasMore": false,
        "totalResults": 1
    },
    "links": []
}
```

*Sample for* **getInventoryItem (slide projector)**

**Search for Item**

GET**{{HostName}}**/evm/config/v1/hotels/{{HotelId}}/inventoryItems?usedForEvents=true&itemName=slide+projector

Sample Response:

```
{
    "items": [
        {
            "hotelId": "BHOTEL",
            "itemId": {
                "id": "39059",
                "type": "ItemInventoryId"
            },
            "itemClassId": {
                "id": "24278",
                "type": "ItemClassId"
            },
            "itemClassCode": "81",
            "itemClassDescription": "Audio Visual Equipment",
            "itemName": {
                "defaultText": "Slide Projector",
                "translatedTexts": [
                    {
                        "value": "Dia-Projektor",
                        "language": "DE"
                    },
                    {
                        "value": "Rétro-projecteur",
                        "language": "F"
                    }
                ]
            },
            "itemDescription": "Slide Projector",
            "itemCode": "S39059",
            "revenueType": "AVEQ",
            "revenueTypeEditable": true,
            "externalOrder": false,
            "print": true,
            "sellInfo": {
                "available": {},
                "availableQty": 0,
                "qtyInStock": 3,
                "defaultQty": 1,
                "sellSeparate": false,
                "sellControl": false,
                "welcomeOffer": true,
                "webBookable": false,
                "itemClassWebBookable": false,
                "outsideStay": false,
                "fixedCharge": false,
                "discountable": true,
                "itemClassDiscountable": true,
                "criticalInventory": true,
                "price": {
                    "amount": 10
                }
```

```
},
"tracesPerDay": false,
"displaySequence": 15,
"eventTypes": [
    "AFT",
    "BKO",
    "BRK",
    "BRKFT",
    "BRKFTB",
    "BRKFTP",
    "CKT",
    "COF",
    "DD",
    "DIN",
    "DINB",
    "DINP",
    "EVT1",
    "EVT2",
    "EVT3",
    "KOF",
    "LIZ",
    "LIZ2",
    "LONG",
    "LUN",
    "LUNB",
    "LUNP",
    "MTG",
    "NOTMEMTG",
    "NYHTFDRG",
    "REC",
    "RECP",
    "SEM",
    "WED",
    "WEDC",
    "WEDDB",
    "WEDDP"
],
"departments": [],
"attributes": [],
"rates": [
    {
        "itemRateId": {
            "id": "31860",
            "type": "ItemRateId"
        },
        "itemRateCode": "DAY",
        "name": "Per Day",
        "price": {
            "amount": 200
        },
        "hourly": false,
        "default": true
    },
    {
        "itemRateId": {
            "id": "90597",
                        "type": "ItemRateId"
                    },
                    "itemRateCode": "COMP",
                    "name": "Complimentary",
                    "price": {},
                    "hourly": false,
                    "default": false
                }
            ],
            "vendors": [],
            "resourceNotes": [],
            "itemClassDisplaySequence": 8,
            "usedForEvents": true,
            "usedForReservation": false,
            "itemFromTemplate": false
        }
    ]
```

*Sample for* **putEventsResources**

**Add the Menu**

PUT**{{HostName}}**/evm/v1/eventsResources

Body:

```
{
    "eventItems": {
        "eventItemInfo": [
            {
                "eventItem": {
                    "quantity": "1",
                    "externalOrder": false,
                    "hotelId": "BHOTEL",
                    "itemId": {
                        "type": "ItemInventoryId",
                        "id": "134562"
                    }
                }
            }
        ],
        "includedInPackage": false,
        "critical": false
    },
    "eventDetails": [
        {
            "eventPrimaryInfo": {
                "eventId": {
                    "type": "EventId",
                    "id": "610375"
                },
                "hotelId": "BHOTEL"
            },
            "eventBlockInfo": {
```

```
        "blockIdList": [
          {
            "type": "Block",
            "id": "875435"
          }
        ],
        "markAsRecentlyAccessed": false
      }
    }
  ]
}
```

### References

- For more information, refer to the Events topic in the OPERA Cloud Services user guide.

# 11

# Add a Catering Package to a Block

### Business Context

Catering packages and templates are designed to make it easier for you to set up Events. They provide pre-configured resources typically used for Events. Templates and packages are similar except that templates do not include prices. To add a Catering Package to a block, you must first search for a block. This topic describes the steps to search for a block using the REST APIs, and describes how to add a Catering Package.

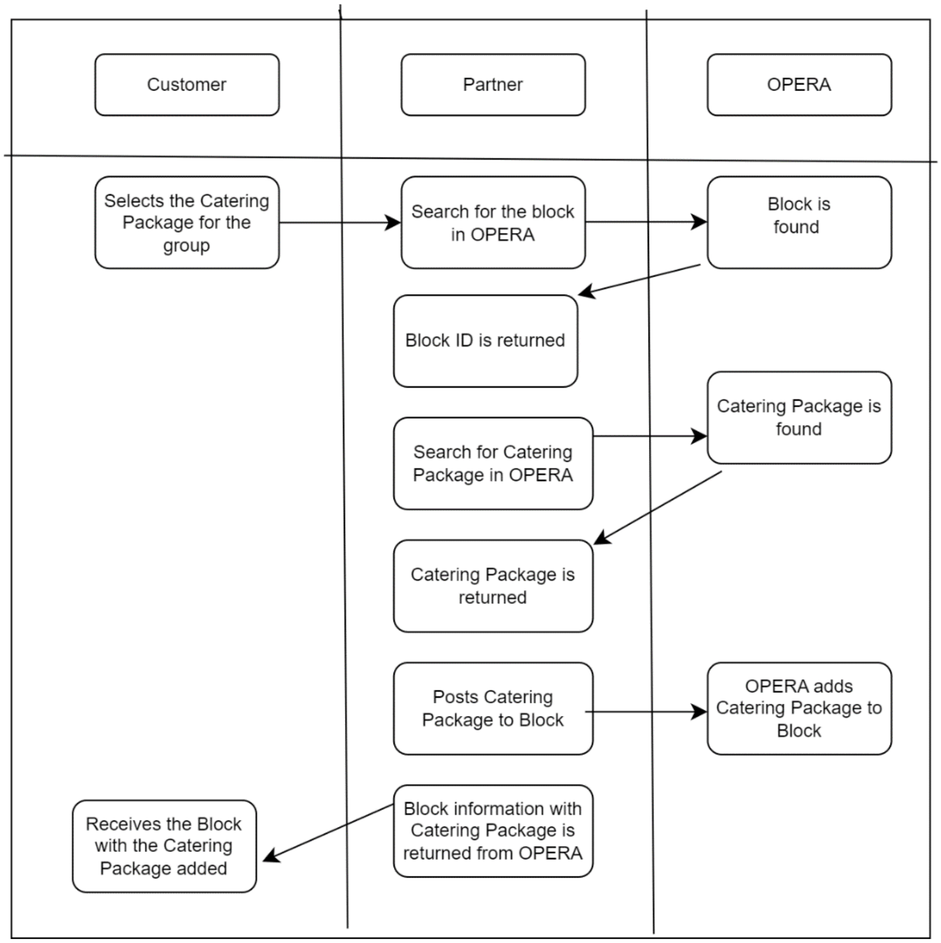
This below workflow diagram explains the typical use cases for adding a Catering Package to a block.

Customer — The meeting planner selects a Catering Package for the one day meeting for the Executive Committee off-site meeting.

Partner — The partner receives the requests and communicates to the property the customer selection.

OPERA Cloud — The application looks for the specific block, returns the block ID, adds the selected Catering Package, and returns the Block with the Catering Package added.

**Figure 11-1 Workflow Diagram**


Customer
Partner
OPERA
Selects the Catering Package for the group
Search for the block in OPERA
Block is found
Block ID is returned
Catering Package is found
Search for Catering Package in OPERA
Catering Package is returned
Posts Catering Package to Block
OPERA adds Catering Package to Block
Receives the Block with the Catering Package added
Block information with Catering Package is returned from OPERA

## Prerequisites

**Table 11-1 Required Software Tools**

| Tool | Description | Links |
|---|---|---|
| Postman | Postman is an API Platform that allows you to design, create and test API's.<br>Use Postman to send API requests or to use the Postman collections provided. | Using the Oracle Hospitality APIs<br>Tool |

**Table 11-2 Current Versions Required**

| OPERA Cloud Platform Module | Description | Minimum Version |
|---|---|---|
| OPERA Cloud Services | The customer must have a subscription to OPERA Cloud Foundation | |
| Oracle Hospitality Integration Platform (OHIP) | For customers, OHIP is included in the subscription to OPERA Cloud Foundation.<br>Partners need a subscription to the Oracle Hospitality Integration Cloud Service. | |

### Configuration of OPERA Controls

To activate the Create Block functionality, you must ensure the following OPERA Controls are active in the target property. You can use the following API operations to validate the settings:

**Ensure that application function BLOCKS - BUSINESS BLOCK is active**

RQ: ent/config/v1/settings?hotelId={{HotelId}}&parameterNameWildCard=BUSINESS BLOCK

**Ensure that application function EVENTS - CATERING EVENTS is active**

RQ: ent/config/v1/settings?hotelId={{HotelId}}&parameterNameWildCard=CATERING EVENTS

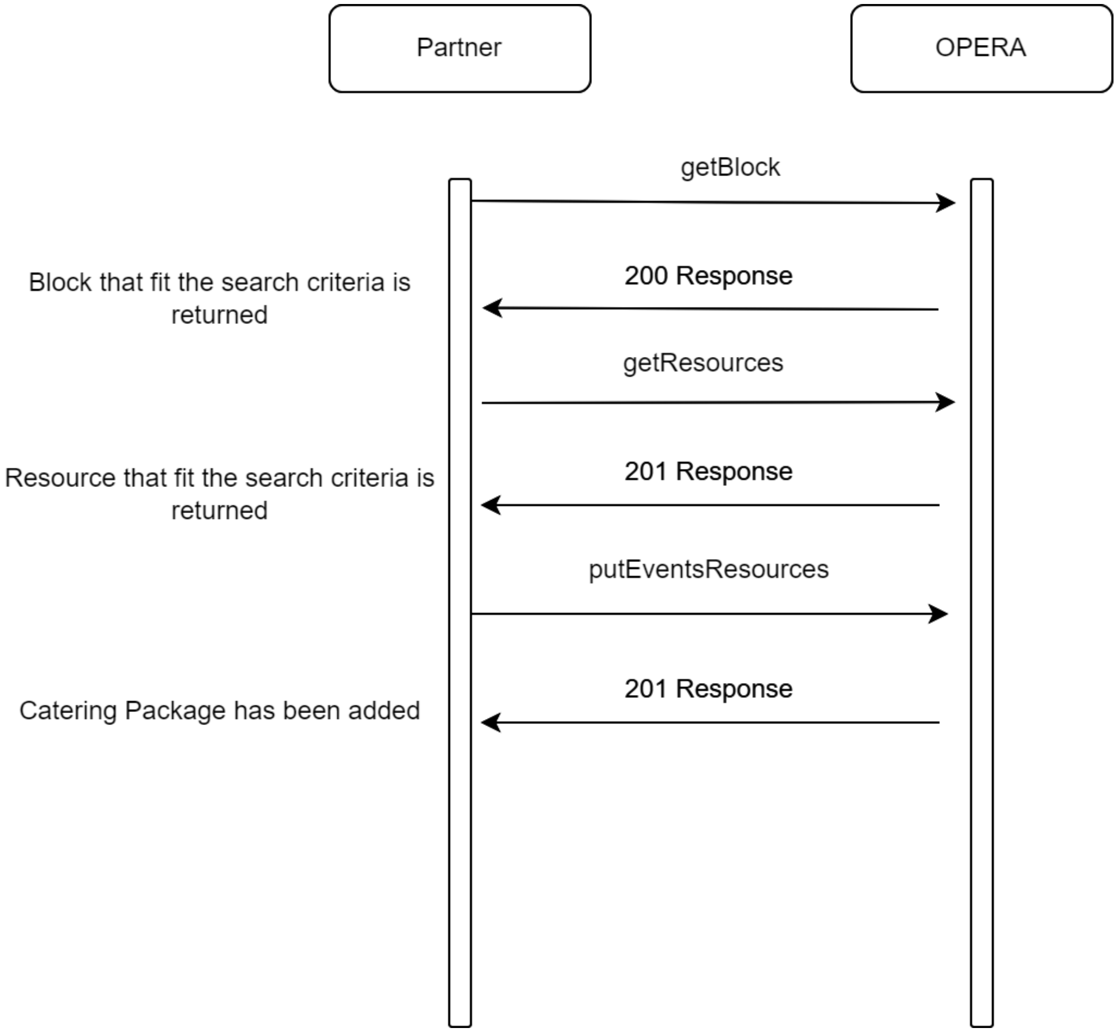
**Ensure that application function EVENTS - EVENT RESOURCES is active**

RQ: ent/config/v1/settings?hotelId={{HotelId}}&parameterNameWildCard=EVENT RESOURCES

### Workflow

Step by step solution including:

**Figure 11-2 The sequence of API calls (in any direction)**


Partner
OPERA
getBlock
Block that fit the search criteria is returned
200 Response
getResources
Resource that fit the search criteria is returned
201 Response
putEventsResources
Catering Package has been added
201 Response

### Description of Steps

A typical process to add a Catering Package to a block is as follows:

1. Search for a block using **getBlock**.
    a. Multiple search parameters are available, for example, dates, status, event type, and other parameters.
2. Search for a Catering Package using **getCateringPackages**.
    a. In this specific scenario, we are searching for all Catering Packages that are matching the status and length in days of the block.
3. Add the Catering Package to the block using **postCateringPackages**.
    a. The Catering Package is added as configured. Pricing, number of items, and other details cannot be modified during this operation. If changes are required, see the appropriate section in this user guide.

**Sample calls**: Sample API calls for the steps described in the diagram.

*Sample for* **getBlock**

In this example, the search is narrowed down to the Block ID.

**Search for Block**

GET**{{HostName}}**/operarest/CN/blk/v1/blocks?onlyOpportunities=false&offset=1&tourSeries=false&markAsRecentlyAccessed=false&leads=false&limit=20&blockIdList=1163926&blockIdType=Block&onlyPickupBlocks=false

*Sample Response*

```
{
    "blockSummaries": {
        "blockInfo": [
            {
                "blockIdList": [
                    {
                        "id": "1163926",
                        "type": "Block",
                        "idContext": "OPERA"
                    },
                    {
                        "id": "3329350",
                        "type": "GroupId",
                        "idContext": "OPERA"
                    }
                ],
                "block": {
                    "hotelId": "BHOTEL",
                    "blockCode": "12122023SANS",
                    "blockName": "Co. & Co.",
                    "blockStatus": {
                        "status": {
                            "code": "INQ",
                            "description": "Inquiry"
                        },
                        "color": "Green"
                    },
                    "status": "F",
                    "cateringOnlyBlock": false,
                    "startDate": "2023-12-12",
                    "endDate": "2023-12-13",
                    "inventoryControl": "NonElastic",
                    "blockIndicators": [
                        {
                            "indicatorName": "OWNERS",
                            "count": 3
                        },
                        {
                            "indicatorName": "ASSOCIATED_PROFILES",
                            "count": 3
                        }
                    ],
                    "rooms": 0,
                    "primaryOwners": {
                        "owner": [
                            {
                                "ownerCode": "CC",
                                "primary": true,
                                "ownership": "Block"
                            },
                            {
                                "ownerCode": "LBR",
                                "primary": true,
                                "ownership": "Rooms"
                            },
                            {
                                "ownerCode": "LBR",
                                "primary": true,
                                "ownership": "Catering"
                            }
                        ],
                        "lockBlockOwners": false,
                        "lockRoomsOwners": false,
                        "lockCateringOwners": false
                    },
                    "blockClassification": "RegularBooking",
                    "isCentralSalesLead": false,
                    "synchronized": false,
                    "groupName": "Sansone & Co.",
                    "tentativeBlock": true,
                    "actualRooms": 0,
                    "reservationType": "GNON",
                    "flexibleDates": false,
                    "deductInventory": false,
                    "cateringDeductInventory": false
                }
            }
        ],
        "totalPages": 1,
        "offset": 20,
        "limit": 20,
        "hasMore": false,
        "totalResults": 1
```

```
    },
    "masterInfoList": [],
    "links": []
}
```

*Sample for* **getCateringPackages**

In this example, the search is by Catering Package Name.

**Search for Item**

GET**{{HostName}}**/operarest/{schemamap}/evm/config/v1/cateringPackages?duration=2&startDateTime=2025-03-30&offset=1&hotelIds={{HotelId}}&minAttendees=10&limit=-1&description=CC+Meeting+Package+w/space&fetchInstructions=PackageInfo&fetchInstructions=BlockPackageInfo&sellDate=2024-03-30&endDateTime=2025-03-31&pkgTmpltType=Packages

Sample Response:

```
{
    "pkgTmplts": [
        {
            "pkgId": {
                "id": "77303"
            },
            "hotelId": "BHOTEL",
            "packageDetail": {
                "pkgCode": "1DAM",
                "description": {
                    "defaultText": "CC Meeting Package w/space",
                    "translatedTexts": [
                        {
                            "value": "Arabic",
                            "language": "A"
                        },
                        {
                            "value": "British English",
                            "language": "B"
                        },
                        {
                            "value": "D German",
                            "language": "D"
                        },
                        {
                            "value": "Croatian",
                            "language": "CRO"
                        }
                    ]
                },
                "duration": 1,
                "webBookable": false,
                "isTemplate": false
            },
            "pricingList": [
                {
                    "id": {
                        "id": "40524"
                    },
                    "priceCode": "IRMEET",
                    "description": "Meeting Day Package",
                    "totalPrice": {
                        "amount": 87
                    },
                    "flatRate": true
                },
                {
                    "id": {
                        "id": "64315"
                    },
                    "priceCode": "PRICE3",
                    "description": "PRICE3",
                    "totalPrice": {
                        "amount": 150
                    },
                    "flatRate": false
                }
            ]
        }
    ],
    "totalPages": 1,
    "offset": 0,
    "limit": -1,
    "hasMore": true,
    "totalResults": 1,
    "links": []
}
```

*Sample for* **postCateringPackages**

**Add the Menu**

POST {{HostName}}/operarest/CN/evm/v1/blocks/2200041036/cateringPackages

Body:

```
{
    "requestCount": "1",
    "criteria": {
        "hotelId": "BHOTEL",
        "blockId": {
            "type": "Block",
            "id": "2200041036"
        },
        "packageCode": "1DAM",
        "priceCode": "IRMEET",
        "status": "TEN",
        "rentalCode": "CUSTOM",
        "beginDate": "2025-03-30"
    }
```

### Limitations and Constraints

The customer must provide the partner with the correct Block Id to add the Catering Package to the block. Likewise, the customer must provide the Catering Package name and/or Id so the selected choice can be added to the Block.

### References

- Review the Anti-Patterns chapter in the OHIP user guide to learn more about functional and technical anti-patterns.
- Fore more information, refer to the Configuring Catering Packages / Templates topic in the OPERA Cloud Services user guide.