Oracle® Hospitality Integration Platform

User Guide

Release 24.1
F92950-01
April 2024

Oracle Hospitality Integration Platform User Guide, Release 24.1

F92950-01

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 explains how to use the Oracle Hospitality Integration Platform Developer Portal and how to explore the Hospitality REST APIs.

Audience

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

Customer Support

Use the Customer Support Portal to search the knowledgebase, access product documentation, contact Oracle Support, or to submit or view a support request to seek help resolving issues.

Access the Customer Support Portal at the following URL:

https://iccp.custhelp.com

When raising a ticket, it must be "**technical ticket**" and should be logged under "**Oracle Hospitality Integration Cloud Service – OHIP**."

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received

- Screenshots of each step you take

> **Note:**
>
> First-time users must register at the Customer Support Portal. For more information about the Customer Support Portal, click the **Getting Started With Support** tile on the Customer Support Portal's main page. You will need your CSI number from your Cloud administrator's welcome email.

For escalations, please refer to https://iccp.custhelp.com.

Support Request

Partners and customers will see the following prompts when logging a support request:

- Enter the Production URL.
- Enter the Production API Call Gateway.
- Enter the Production End Point URL.
- Enter the Production OPERA Instance URL.
- Enter the Chain.
- Enter the Property.
- Enter the User.

If the request relates to the Streaming API, also include the following:

- Confirm whether the environment card shows "Streaming Enabled."
- Enter the applicationId. This can be found by going to the OHIP Developer Portal, opening the application that is being used for the streaming API, and copying the last part of the URL, which is a number.
- Confirm that changes are occurring in the environment to which Business Events are subscribed.
- Enter the client being used to access the Streaming API:
  - Postman
  - GraphiQL
  - Oracle sample NodeJS client
  - Your own code
- Enter the error messages (if any) that are being received.

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
April 2024	Initial publication

1

Getting Started

Both Oracle Hospitality Integration Cloud Service users and OPERA Cloud Foundation users are authorized to access the Oracle Hospitality Integration Platform, the Oracle Hospitality Developer Portal, and Oracle Hospitality APIs.

Digital Learning Access

Oracle Hospitality Digital Learning content is available for Oracle Hospitality Integration Platform (OHIP). In Digital Learning under 'Learn about Oracle's Products,' you can choose from the following products:

- Oracle Hospitality OPERA Cloud
- Oracle Hospitality Reporting and Analytics
- NOR1 an Oracle Company
- Oracle Hospitality Integration Platform

After you select a product, different learning paths are available. For OHIP, there are two available Learning Paths:

- **Oracle Hospitality Integration Platform (OHIP) Learning Path** is designed for OPERA Cloud Foundation Customers who wish to access OHIP at their organization. The Oracle Hospitality Integration Platform (OHIP) Learning Path will provide an overview of the benefits, highlight features, and provide instruction on the fundamentals of OHIP.
- **Oracle Hospitality Integration Platform (OHIP) for Integrators Learning Path** is designed for Oracle Hospitality Integration Cloud Service Integrators (also known as Integration Partners) who wish to access OHIP. The Oracle Hospitality Integration Platform (OHIP) path will provide an overview of the benefits, highlight features, and provide instruction on the fundamentals of OHIP.

Getting Started for Partners (for Oracle Hospitality Integration Cloud Service Users)

There are two ways to onboard partners to the Oracle Hospitality Integration Cloud Service:

- Oracle Store — You can onboard by purchasing the Oracle Hospitality Integration Cloud Service through the Oracle Store. Follow the procedure below to onboard through the Oracle store.
- Partner Registration Cost Price Quote (CPQ) form— If you are unable to onboard through the Oracle Store, you can onboard by submitting the partner registration (CPQ) form. You can request this form through email by contacting hospitality-integrations_ww@oracle.com.

Oracle Store

To onboard through the Oracle Store:

> **Note:**
>
> You must have an Oracle account to proceed with the shopping cart checkout. To create a new Oracle account, go to the Oracle Store and click the **New user?** button and complete the form. After submitting the form, follow the instructions to check your email to verify your email address.

1. Go to the Hospitality Integration Platform and click **Shop now** and then click **Add to Cart**.

   Alternatively, go directly to the Oracle Shop and click **Add to Cart**.
2. Click your shopping cart to proceed to check out.
3. Click **Checkout**.
4. Follow the instructions to complete your account details, additional information, service information, billing information, and payment method and agree to the terms and conditions of the Cloud Service Agreement.

   > **Note:**
   >
   > Once you have onboarded with a credit card, you are also able to change your payment method to a PO or update your credit card details. To do this, contact Billing Support for assistance. Given the sensitivity of payment information, do not include that information in email communications.
5. Click **Place Order**.

To view and track your order, go to your Account Dashboard and click **Orders**.

After purchasing the Oracle Hospitality Integration Cloud Service, you will receive the following emails from either the Oracle Store or the CPQ process:

- The first email is a confirmation of your order and contains the order number for reference.
- A second email invites you to set up your Oracle Cloud Account. This email contains your order number.

  Follow the below steps to create your Oracle Cloud Account by clicking the **Create Cloud Account** link in the email and providing the following details.

  1. **Cloud Account Name**: A unique name for your Oracle Cloud Account.
  2. **Email Address**: Enter your email address. This is the email address to which the emails in steps 3 and 4 will be sent.
  3. Enter the email address of the cloud account admin and service admin for your services. This admin user can create other administrators or users.
- A third email prompts you to confirm your Oracle Cloud Account.
- A fourth email confirms that your setup is complete. This includes the Oracle Cloud Console URL to enable you to perform Cloud Admin tasks as well as a

Service Instance URL to the Oracle Hospitality Developer Portal. This URL conforms to the following format:

https://partner.hospitality-dev-portal.us-ashburn-1.ocs.oraclecloud.com/<Oracle Cloud Account Name>/ui/

> **Note:**
>
> If you have not received all the above emails within 24 hours of placing your order, please contact the Oracle customer service number referenced in the first email.

Digital Learning Access for Partners

Oracle Hospitality Integration Platform (OHIP) for Integrators Learning Path is designed for Integration Partners who wish to access OHIP. The Oracle Hospitality Integration Platform (OHIP) path provides an overview of the benefits and highlight features and provide instruction on the fundamentals of OHIP. To gain access, follow the instructions below:

For Partners who have purchased B92141 Oracle Hospitality Integration Cloud Service:

New Partners

- New partners as of 27-MAR-2023 will be sent a Digital Learning activation email once their order is received.
- The Digital Learning Activation email will be sent to the End User Contact that appears on the order.
- If an activation email is not received, email **operaenablement_ww@oracle.com** and include the following information:
  - **Company Name** (as it appears on the Order/Account)
  - **Primary Contact Email**

Existing Oracle Hospitality Integration Cloud Service Partners

If you have access to Oracle Hospitality Digital Learning, use the search filters on the Digital Learning Platform to find your desired OHIP content.

If you do not have access, follow these instructions:

- Email **operaenablement_ww@oracle.com** and include the following information:
  - **Company Name** (as it appears on the Order/Account)
  - **Address** (include Country)
  - **Primary Contact Name**
  - **Primary Contact Email**
  - **Approximate Purchase Date** (that is, the approximate date Oracle Hospitality Integration Cloud Service was purchased)

Once this information is received, the Digital Learning team will verify the information. Please allow five business days for responses.

Oracle Hospitality Open Forum

Oracle Hospitality Open Forum hosted within Slack is an inclusive platform designed to bring together individuals who share a passion for our industry. This vibrant community aims to foster collaboration, knowledge sharing, and networking opportunities, allowing members to connect, learn, and grow together.

The decision to create this open community was driven by several important reasons:

- **Collaboration and Knowledge Sharing**: By launching this open forum, we aim to encourage collaboration and knowledge sharing among Vendors, SIs, and Oracle Hospitality. It provides a dedicated space for members to ask questions, seek advice, and engage in thought-provoking discussions, leading to mutual growth and learning.
- **Networking and Relationship Building**: The Slack open community offers a platform for individuals to network and form valuable connections. This enables the establishment of relationships that can potentially lead to partnerships, mentorships, and career opportunities.
- **Inclusivity and Diversity**: The open community promotes inclusivity and diversity by bringing together individuals from various backgrounds and experiences. It fosters the exchange of diverse perspectives and creates an environment of understanding and empathy.
- **Maturity of the OHIP Platform** and a growing number of enabled platform components.

To facilitate effective communication and organization, we have structured the open community as follows:

Channels

1. **01_announcements**: This channel (managed by Oracle Hospitality) is used to share insightful updates on our products, maintenance, and important reminders with the partner community.
2. **02_open_forum**: The open forum channel enables partners, SIs, and other community members to interact directly with each other. Oracle will moderate this channel, and we kindly request all participants to adhere to etiquette rules to maintain a collaborative approach.

Additional Benefits

Note the following additional benefits:

- New partners who join OHIP will be granted access to a Private Channel during the first 30 days after joining OHIP. This channel will assist you with your initial development efforts.
- Existing partner private channels will be archived after 30 days from the open forum launch, with the exception outlined below.

Partners who meet the following criteria will remain eligible for private channel communication with Oracle Hospitality:

- Active OPN Membership
- Published Oracle Hospitality Marketplace Listing
- At least one customer in Production via OHIP

Forum Etiquette

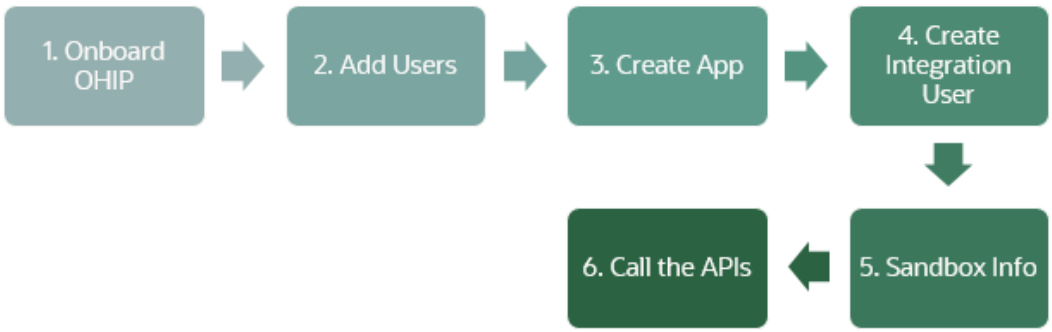
Here are some of the guidelines when interacting in the open community:

- Refrain from sharing personal or sensitive information, including usernames, passwords, and personally identifiable information (PII).
- Avoid the sharing of payment card information (PCI data) to prevent fraud or unauthorized access.
- Maintain respectful and considerate communication, avoiding harassment, discrimination, or hate speech.
- Actively participate and engage in the community by sharing expertise, asking questions, and contributing to discussions.
- Adhere to these guidelines to create a vibrant and secure space for collaboration and learning in the community.

Channel Access

You will automatically be granted access to the channels as you onboard OHIP. Please allow a few days after joining for your invitation to join.

Quick Start for Partners (Using the Partner Sandbox)



1. Onboard to OHIP via the Oracle Store.
2. Add users to the Developer Portal.
3. Sign in to the Developer Portal and register an application.
4. Create an integration user.
5. Obtain the credentials and gateway for the partner sandbox. The hotelId is SAND01.
6. Call the APIs.

   If you need additional consulting to get you started, you can purchase it in groups of 2 hours. Visit the Oracle Store for more information.

Adding Developer Portal Users

> **Note:**
>
> To add or manage Portal users, you must have the Cloud Account Administrator or Service Administrator user role, your activated Oracle Cloud account, and your Oracle Cloud Console URL. For more information on how to onboard or provision an Oracle Cloud account, please refer to Getting Started for Partners.
>
> The Oracle Cloud Console is customizable and can present different views to users depending on how it is set up.

The Account Administrator and Service Administrator users can add new users from the My Oracle Services screen in the Oracle Cloud Console.

> **Note:**
>
> Developer Portal users do not have access to call APIs.

Follow this process to create Developer Portal users:

1. Log in to your Oracle Cloud Account using your Oracle Cloud Console URL.
2. On the User Assignments screen, click **Users**.
3. On the Users screen, click **+ Add**.

   - On the Add User screen, enter the user details **First Name**, **Last Name**, **Email**, and **User Name**.

     The new user account is created.

   For details on how to manage users in OPERA Cloud Identity Management, refer to the OPERA Cloud Identity Management User Guide.

Assigning Users to Roles

After adding a new user, you can assign the **ApplicationDeveloper** role to the user, which gives them access to the Oracle Hospitality Developer Portal.

1. Select the **Application Developer Portal Role** (under Hospitality Developer Portal).
2. Click **Finish**.

> **Note:**
>
> The helper widget with the lifebelt icon might overlap the drop-down icon to the right of the Application roles table. If so, you must reposition it so the drop-down icon is visible and the ApplicationDeveloper role can be allocated to users.

Signing In to the Oracle Hospitality Developer Portal

Signing In to the Developer Portal

Sign in to the Oracle Hospitality Developer Portal to create applications, discover available Oracle Hospitality APIs, and register Oracle Hospitality APIs to your applications.

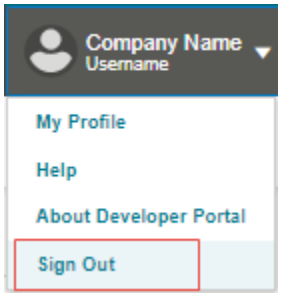
1. Open the Oracle Hospitality Developer Portal in a browser to the URL provided.
2. Enter your username and password.
3. Click **Sign In**.

> **Note:**
>
> Your account locks if you enter an incorrect username or password three times. For help getting back into your account, click **Can't sign in** and the follow instructions on the screen.

Signing Out of the Oracle Hospitality Developer Portal

You can sign out of the Oracle Hospitality Developer Portal using the User menu.

1. On the Oracle Hospitality Developer Portal, click the **User** menu drop-down.
2. Click **Sign Out.**



Changing Your Password

> **Note:**
>
> These instructions for changing your password apply to those who onboarded through the Oracle Hospitality Integration Cloud Service.
>
> If you onboarded through Oracle Hospitality OPERA Cloud Foundation/ OPERA Cloud Services, refer to Changing Your Password in the *Onboarding Customers* section.

To change your password:

1. Sign in to the Oracle Hospitality Developer Portal.
2. On the user drop down menu, click **My Profile**.
3. Click the **Change My Password** tab:
    a. Enter your **Old Password**.
    b. Enter the **New Password**.
    c. After all password criteria are met, enter the new password to confirm, and then click Submit.

Getting Started for Hoteliers (for Oracle Hospitality OPERA Cloud Foundation Users with Oracle Hospitality Shared Security Domain)

Oracle Hospitality OPERA Cloud Foundation includes access to the Oracle Hospitality Integration Platform.

Upon provisioning of or migration to Oracle Hospitality OPERA Cloud Foundation, you will receive an email advising you of the Oracle Hospitality Developer Portal URL.

For production OPERA environments, it will follow this URL format:

- https://customer.hospitality-dev-portal.us-ashburn-1.ocs.oraclecloud.com/ashburn/ui/

  or

- https://customer.hospitality-dev-portal.us-ashburn-1.ocs.oraclecloud.com/frankfurt/ui/

For non production OPERA environments, it will follow one of these URL formats:

- https://customer.hospitality-dev-portal.us-ashburn-1.ocs.oraclecloud.com/ashburnuat/ui/
- https://customer.hospitality-dev-portal.us-ashburn-1.ocs.oraclecloud.com/ashburnuat02/ui

- https://customer.hospitality-dev-portal.us-ashburn-1.ocs.oraclecloud.com/frankfurtuat/ui/

Digital Learning Access for Customers

Oracle Hospitality Integration Platform (OHIP) Learning Path is designed for OPERA Cloud Foundation Customers who want to access OHIP at their organization. The Oracle Hospitality Integration Platform (OHIP) Learning Path provides an overview of the benefits and highlight features and provides instruction on the fundamentals of OHIP.

New Customers

For Customers with Oracle Hospitality OPERA Cloud Foundation (Hotel/Resorts/Reseller), follow these instructions to access Digital Learning:

- The Digital Learning Activation email will be sent to the End User Contact that appears on the customer's order.
- If an activation email is not received, email **operaenablement_ww@oracle.com** and include the following information:
  - **Customer Name** (as it appears on their Order/Account)
  - **Primary Contact Email**

Existing Customers

If you have access to Oracle Hospitality Digital Learning, use the search filters on the Digital Learning Platform to find your desired OHIP content.

If you do not have access, but have OPERA Cloud Foundation, follow these instructions:

- Email **operaenablement_ww@oracle.com** and include the following information:
  - **Customer Name** (as it appears on the customer's Oracle account)
  - **Address** (include Country)
  - **Primary Contact Name**
  - **Primary Contact Email**

  Once this information is received, the Digital Learning team will verify the information. Please allow five business days for responses.

Quick Start for Hoteliers (in UAT Environment)

Prerequisites

OPERA Cloud Foundation SKU

- The hotel must purchase an OPERA Cloud Foundation SKU.

> **Note:**
>
> OHIP is not available for hotels using a legacy OPERA Cloud SKU.

Onboarded in OHIP

- The chain containing properties that are on OPERA Cloud Foundation must be onboarded with OHIP.

Oracle Cloud Infrastructure

- The onboarded chain and properties must be on Oracle Cloud Infrastructure (OCI) and NOT in a legacy data center.

Making API Calls

An OPERA Cloud chain administrator must create the DEVELOPERPORTALACCESS role via Oracle Identity Management (OIM) for both UAT and PRODUCTION.

1. Assign the DEVELOPERPORTALACCESS role to all OPERA Cloud users requiring access to the OHIP Developer Portal. This will support Single-Sign On (SSO) with OPERA Cloud. See Getting Started for more information.
   - **UAT**: Refer to your Welcome email for this URL.
   - **PROD** : Refer to your Welcome email for this URL.
2. Create a new integration user via Self Service using the UAT and PROD URLs. Refer to your Welcome email for these URLs. Once a request is submitted, the OPERA Cloud chain administrator must approve it. See Using the Oracle Hospitality APIs for more information.
3. Access the Developer Portal for UAT or PROD (using the URLs from your Welcome email) and click the **Environment** tab.
4. Click the environment card and to copy the client Id and secret. See Using the Oracle Hospitality APIs for more information.
5. Register a new application in the **Applications** tab of the OHIP Developer Portal. When creating a new application, subscribe the application to all the APIs that appear. See Register and Manage Applications for more information.
6. Enter the hotel ID from which you want to get data. This is not the same ID as the chain admin and should be the ID of a specific hotel.

   If you need to call early adopter (v0) APIs, send an email to hospitality-integrations_ww@oracle.com requesting to join the early adopter programme. You must reply to the email accepting the terms and conditions of the programme before access is provided.

Configuring Postman

1. Obtain the postman collections via the following:
   - Cloning our github repo and then importing the collections / environment under the postman-collections folder.
   - Forking our public postman workspace directly.
2. Set up a postman environment (one for UAT and one for PROD) with the previously obtained information plus the gateway URL:
   a. **HostName**: API gateway URLs for UAT and PROD. Refer to your Welcome email for these URLs.
   b. **Username**: The integration username previously obtained for UAT or PROD.
   c. **Password**: The integration password previously obtained for UAT or PROD.

d. **CLIENT_ID**: Client ID previously obtained from the Developer Portal for UAT or PROD.

e. **CLIENT_SECRET**: Client ID previously obtained from the Developer Portal for UAT or PRO.

f. **AppKey**: The application key previously obtained .

g. **HotelId**: Hotel ID against which you want to perform actions. For example, obtaining reservation data.

3. Once all environments have been configured, select the **Get OAuth Token** collection and make a call.

   If everything is set up correctly, OHIP responds with a HTTP 200 OK response, which includes the OAuth token.

   Now you can try out other collections as required.

Creating the Developer Portal Access Role

> **Note:**
>
> The below steps are required only for OPERA Cloud environments that are integrated with Oracle Hospitality Shared Security Domain. For environments with OPERA Cloud Identity Domain, the role should be available in the tenant domain.

This procedure is for a user with Chain Admin access in SSD-OIM to create an OHIP Developer Portal role for the corresponding OPERA instance. The role name is DEVELOPERPORTALACCESS.

When creating the role, ensure the role is assigned to the correct chain code.

Once the role is created, you can assign it to users so they can access the OHIP Developer Portal with the URLs specified in Getting Started for Hoteliers.

To add the DEVELOPERPORTALACCESS role

1. Navigate to and log in to the Oracle Identity Self Service portal.

   This step must be performed by a Chain Administrator because OHIP only supports CHAIN level access.

   > **Note:**
   >
   > The Oracle Identity Self Service URL is listed in the Welcome Letter received by your designated OPERA Property Administrator.

2. Click the **Manage** button.
3. Click the **Roles** tile and then click the **Create** link.
4. Enter the following required information into the General Role section:

a. **Name**. Enter the Chain Organization name followed by DEVELOPERPORTALACCESS. For example, PARCHN01-DEVELOPERPORTALACCESS.

b. **Display Name**. Reenter the same name as above.

> **Note:**
>
> The **Owned By** field will automatically populate based on who created the role.

5. The Hierarchy, Access Policy, and Members steps are not required. Click the **Next** to arrive on the Organization step.
6. Click the **Add Chain Organizations** button.
7. Search for the Chain Organization name and click the **Add Selected** button to add the role.
8. Click the **Select** button.
9. Verify the DEVELOPERPORTALACCESS and then click **Next**.
10. Verify the information you entered and then click the **Finish** button.

Adding Developer Portal Users

To access the Developer Portal, a hotel administrator assigns the DEVELOPERPORTALACCESS role to a chain org-level user in the Oracle Hospitality Shared Security Domain. For the steps to create this role, see Creating the Developer Portal Access Role.

> **Note:**
>
> Users assigned to this role must belong to the same organization specified in Creating the Developer Portal Access Role. Further, users must be chain org-level users to be granted the DEVELOPERPORTALACCESS role and to successfully access the Developer Portal.

> **Note:**
>
> Developer Portal users do not have access to call APIs.

For details on how to manage users in the Oracle Hospitality Shared Security Domain, refer to the Customer Support Portal document:

Oracle Hospitality Shared Security Domain (SSD) for OPERA

Signing In to the Oracle Hospitality Developer Portal

Signing In to the Developer Portal

Sign in to the Oracle Hospitality Developer Portal to create applications, discover available Oracle Hospitality APIs, and register Oracle Hospitality APIs to your applications.

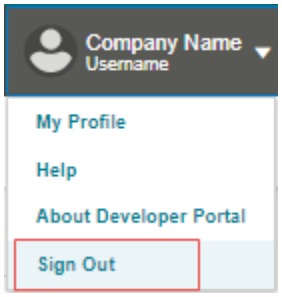
1. Open the Oracle Hospitality Developer Portal in a browser to the URL provided.
2. Enter your username and password.
3. Click **Sign In**.

> **Note:**
>
> Your account locks if you enter an incorrect username or password three times. For help getting back into your account, click **Can't sign in** and the follow instructions on the screen.

Signing Out of the Oracle Hospitality Developer Portal

You can sign out of the Oracle Hospitality Developer Portal using the User menu.

1. On the Oracle Hospitality Developer Portal, click the **User** menu drop-down.
2. Click **Sign Out.**



Changing Your Password

> **Note:**
>
> These instructions for changing your password apply to those who onboarded through the Oracle Hospitality OPERA Cloud Foundation.
>
> If you onboarded through the Oracle Hospitality Integration Cloud Service, refer to Changing Your Password in the *Onboarding Partners - Oracle Hospitality Integration Cloud Service* section.

To change your password:

1. Sign in to the Oracle Hospitality Developer Portal.
2. On the user drop down menu, click **My Profile**. A new browser tab opens for Oracle Identity Manager (OIM).
3. Click the **My Information** tile.
4. In the **Change Password** section, enter your **Old Password**, enter your **New Password**, and reenter your new password in the **Confirm New Password** field.
5. Click **Apply**. After a password change, a confirmation message shows that your password was successfully updated.

Getting Started for Hoteliers (for Oracle Hospitality OPERA Cloud Foundation Users with OPERA Cloud Identity Management)

Oracle Hospitality OPERA Cloud Foundation includes access to the Oracle Hospitality Integration Platform.

Upon provisioning of or migration to Oracle Hospitality OPERA Cloud Foundation, you will receive an email advising you of the Oracle Hospitality Developer Portal URL.

For production OPERA environments, it will follow this URL format:

- https://<ohip-domain>/<enterpriseID>prd/ui

For non production OPERA environments, it will follow one of these URL formats:

- https://<ohip-domain>/<enterpriseID>uat/ui

For example, the production URL for a customer portal with enterpriseID "testenterprise" would be the following: https://oracle-test.com/testenterpriseprd/ui.

Digital Learning Access for Customers

Oracle Hospitality Integration Platform (OHIP) Learning Path is designed for OPERA Cloud Foundation Customers who want to access OHIP at their organization. The Oracle Hospitality Integration Platform (OHIP) Learning Path provides an overview of the benefits and highlight features and provides instruction on the fundamentals of OHIP.

New Customers

For Customers with Oracle Hospitality OPERA Cloud Foundation (Hotel/Resorts/ Reseller), follow these instructions to access Digital Learning:

- The Digital Learning Activation email will be sent to the End User Contact that appears on the customer's order.
- If an activation email is not received, email **operaenablement_ww@oracle.com** and include the following information:
  - **Customer Name** (as it appears on their Order/Account)
  - **Primary Contact Email**

Existing Customers

If you have access to Oracle Hospitality Digital Learning, use the search filters on the Digital Learning Platform to find your desired OHIP content.

If you do not have access, but have OPERA Cloud Foundation, follow these instructions:

- Email **operaenablement_ww@oracle.com** and include the following information:
  - **Customer Name** (as it appears on the customer's Oracle account)
  - **Address** (include Country)
  - **Primary Contact Name**
  - **Primary Contact Email**

  Once this information is received, the Digital Learning team will verify the information. Please allow five business days for responses.

Quick Start for Hoteliers (in UAT Environment)

Prerequisites

OPERA Cloud Foundation SKU

- The hotel must purchase an OPERA Cloud Foundation SKU.

> **Note:**
>
> OHIP is not available for hotels using a legacy OPERA Cloud SKU.

Onboarded in OHIP

- The chain containing properties that are on OPERA Cloud Foundation must be onboarded with OHIP.

Oracle Cloud Infrastructure

- The onboarded chain and properties must be on Oracle Cloud Infrastructure (OCI) and NOT in a legacy data center.

Making API Calls

1. Assign the DEVELOPERPORTALACCESS role to all OPERA Cloud users requiring access to the OHIP Developer Portal. This will support Single-Sign On (SSO) with OPERA Cloud. See Getting Started for more information.
   - **UAT**: Refer to your Welcome email for this URL.
   - **PROD** : Refer to your Welcome email for this URL.
2. Access the Developer Portal for UAT or PROD (using the URLs from your Welcome email) and click the **Environment** tab.
3. Click the environment card and to copy the enterpriseID, scope, client Id, and secret. See Using the Oracle Hospitality APIs for more information.
4. Register a new application in the **Applications** tab of the OHIP Developer Portal. When creating a new application, subscribe the application to all the APIs that appear. See Register and Manage Applications for more information.

5. Enter the hotel ID from which you want to get data. This is not the same ID as the chain admin and should be the ID of a specific hotel.

   If you need to call early adopter (v0) Oracle Hospitality Distribution APIs, send an email to hospitality- integrations_ww@oracle.com requesting to join the early adopter programme. You must reply to the email accepting the terms and conditions of the programme before access is provided.

Configuring Postman

1. Obtain the postman collections via the following:
   - Cloning our github repo and then importing the collections / environment under the postman-collections folder.
   - Forking our public postman workspace directly.
2. Set up a postman environment (one for UAT and one for PROD) with the previously obtained information plus the gateway URL:
   a. **HostName**: API gateway URLs for UAT and PROD. Refer to your Welcome email for these URLs.
   b. **CLIENT_ID**: Client ID previously obtained from the Developer Portal for UAT or PROD.
   c. **CLIENT_SECRET**: Client ID previously obtained from the Developer Portal for UAT or PRO.
   d. **AppKey**: The application key previously obtained .
   e. **EnterpriseID**: The unique identifier for the enterprise.
   f. **Scope**: The scope variable representing the permissions of the client for authorization.
   g. **HotelId**: Hotel ID against which you want to perform actions. For example, obtaining reservation data.
3. Once all environments have been configured, select the **Get OAuth Token** collection and make a call.

   If everything is set up correctly, OHIP responds with a HTTP 200 OK response, which includes the OAuth token.

   Now you can try out other collections as required.

Adding Developer Portal Users

To access the Developer Portal, a hotel administrator assigns the DEVELOPERPORTALACCESS role to a chain org-level user in the Oracle Hospitality Cloud Identity Management.

> **Note:**
>
> Users must be chain org-level users to be granted the DEVELOPERPORTALACCESS role and to successfully access the Developer Portal.

> **Note:**
>
> Developer Portal users do not have access to call APIs.

For details on how to manage users in OPERA Cloud Identity Management, refer to Group Management in the OPERA Cloud Identity Management user guide.

Signing In to the Oracle Hospitality Developer Portal

Signing In to the Developer Portal

Sign in to the Oracle Hospitality Developer Portal to create applications, discover available Oracle Hospitality APIs, and register Oracle Hospitality APIs to your applications.

1. Open the Oracle Hospitality Developer Portal in a browser to the URL provided in the welcome letter.

   The customer portal URL can also be obtained using the EnterpriseID.

   - For production environments: https://<ohip-domain>/<enterpriseID>prd/ui
   - For non-production environments: https://<ohip-domain>/<enterpriseID>uat/ui
2. Enter your username and password.
3. Click **Sign In**.

> **Note:**
>
> Your account locks if you enter an incorrect username or password three times. For help getting back into your account, click **Can't sign in** and the follow instructions on the screen.

4. If your portal user has access to multiple chains of the enterprise, select the chain you want to administer once you log in.
   a. Search for and select the chain to which you want to login.

      You can also select this chain as the default chain for login by selecting the **Set this chain as default** option. This will avoid selecting the chain on each login and will directly log in to the default chain on each login.
5. Click **Select**.

Switching between Chains

If your portal user has access to multiple chains, you can switch between chains once logged in to the Oracle Hospitality Developer portal.

To switch between chains:

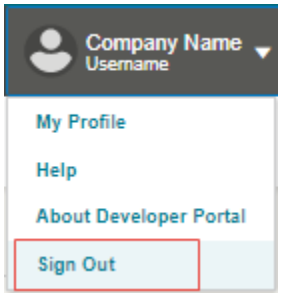
1. On the Oracle Hospitality Developer Portal, click the **User** menu drop-down.
2. Select **Switch Chains**.
3. Search for and select the chain to which you want to switch. You can also change the default chain for login to this chain by selecting the **Set this chain as default** option.

4. Click **Select**.

Signing Out of the Oracle Hospitality Developer Portal

You can sign out of the Oracle Hospitality Developer Portal using the User menu.

1. On the Oracle Hospitality Developer Portal, click the **User** menu drop-down.
2. Click **Sign Out.**



Changing Your Password

> **Note:**
>
> These instructions for changing your password apply to those who onboarded through the Oracle Hospitality OPERA Cloud Foundation.
>
> If you onboarded through the Oracle Hospitality Integration Cloud Service, refer to Changing Your Password in the *Onboarding Partners - Oracle Hospitality Integration Cloud Service* section.

To change your password:

1. Sign in to the Oracle Hospitality Developer Portal.
2. On the user drop down menu, click **My Profile**.
3. Click the **Change My Profile** tab.
   - Enter your **Old Password**.
   - Enter the **New Password**.
   - After all password criteria is met, enter the new password to confirm and then click **Submit**.

2

Discover and Subscribe to Oracle Hospitality APIs

Learn how to discover and subscribe to Oracle Hospitality APIs for your applications in the Oracle Hospitality Developer Portal.

API Search Engine

The API Search Engine enables you to filter by modules, workflows, lifecycle, and method. You can also search by free text to find an API or operation that meets your business case. Search results include operation level details from which you can access links to the API documentation and the Postman sample.

The table below lists all of the filters available on the left side of the page. Select from these filters to refine your API search results. The API search results appear in the center of the page in the form of API display cards.

Table 2-1 Side Filters

Filter	Description
Content	Search by API module or workflow. A workflow is a sample list of operations a user would perform for a specific scenario, such as creating a new block or performing a check-in.
API Lifecycle	APIs can be V0, V1, or Deprecated. V0 are described in the Discover and Subscribe to Oracle Hospitality APIs chapter. V1 are production ready and guaranteed for backward compatibility. Deprecated are no longer recommended for use. You should plan to cease using these and move to a V0/V1 equivalent operation.
API Category	The APIs are split into categories relevant to different Oracle Hospitality products. For example, Property are the APIs for OPERA Cloud, and Distribution are the APIs for the distribution platform.
API Module	Within a category, there are many APIs divided into modules and each module contains many operations. Filter by Module to narrow your search or leave this blank to view all the available modules.
API Workflow	API workflows are a collection of many different API call samples showing steps on how to perform functional workflows (for example, digital check-in and checkout).

Table 2-1 (Cont.) Side Filters

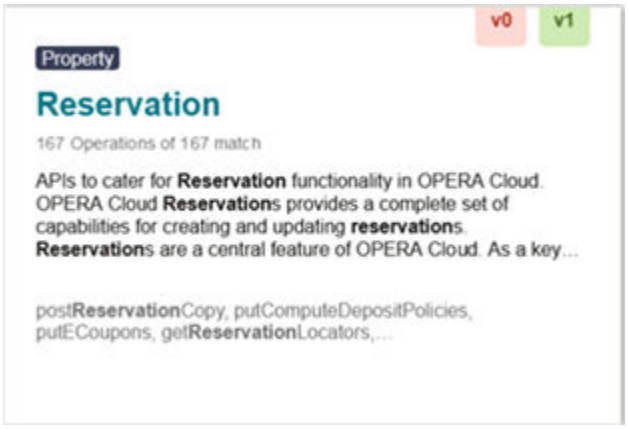
Filter	Description
Method	Filter by the following REST API Methods: **Get**: Retrieve information about the REST API resource. **Post**: Create a REST API resource. **Put**: Update a Rest API resource. **Delete**: Delete a REST API resource or related component. **Head**: Similar to a get, but the server does not return a response body. This method determines if a resource exists and is currently used for Property Asynchronous APIs.

> **Note:**
>
> The number in brackets next to an API indicates the number of operations within it.

API Display Cards

API cards appear based on the search criteria you enter. If the search field is left blank, all API cards appear in alphabetical order.

Figure 2-1 API Display Card Example



The API Display Card displays the below information.

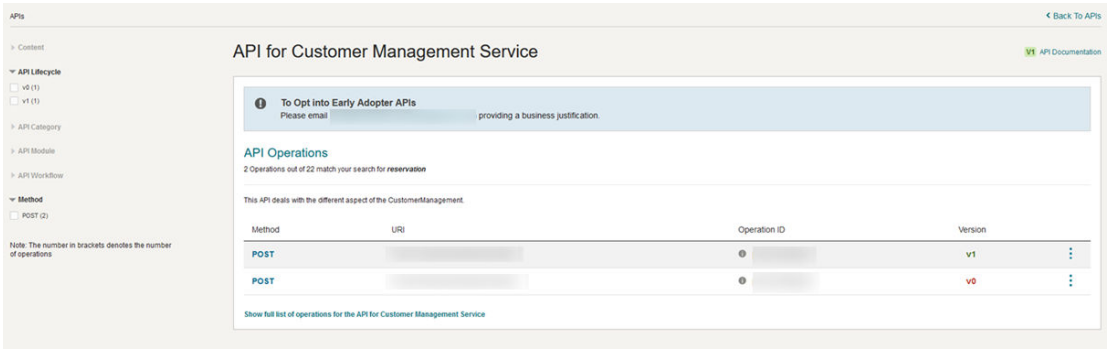
Table 2-2 API Display Card Fields

Field	Description
Property	This icon shows the category to which this API belongs. For example, Property are the APIs for OPERA Cloud, and Distribution are the APIs for Oracle Hospitality Distribution.
API Module Name	The name of the module. For example Reservation, Blocks, Cashiering, Reservations, and so on.
V0 / V1	V0 and V1 flags represent the operation level versioning in the API. If both flags are shown, it indicates that some operations are V1 and some are V0 within the module.
Number of Operations	Depending on your search, the API card will display the number of operations matching your search. If the search is left blank, the API card will show the total number of operations in that specific API.
Summary of the API	A short description of the API.
Operations that match your search	If you search a specific phrase or word and it matches the operations, the operations are highlighted on the card to provide a preview of the results displayed on the next page.

API Page

Once you click an API display card, the page for the API appears and shows more details about the operations within that API.

Figure 2-2 Example of an API Page — API for Customer Management Service



The API page shows the following details for the operations:

Table 2-3 API Page Details

Field	Description
API Documentation	Click this link to view the API documentation.
Information button	Click the  button to view a short description of this operation.
Method	The type of API method, such as POST, GET, PUT, DELETE, and so on.
URI	The URI path for this operation. The URI is depreciated if it has a strike through.
Operation ID	The ID for the API operation.
Version	The API version, which is either v0 or v1.
Vertical ellipsis	Click to access links to the API documentation and the Postman sample.
Show full list link	Click the link below the table to view a complete list of operations for the API.

Viewing Oracle Hospitality APIs

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

1. Click the **API Documentation** link to view the embedded documentation for the APIs.
2. On the left and middle panels, you can view all the available API resources and view an explanation for each field in both the request and the response. In the search field, you can enter the full name or partial name of a resource to further filter the results.
3. The right panel shows example requests and responses.

Early Adopter API Program

The Early Adopter API Program offers program members early access to newly released "version 0" Oracle Hospitality APIs. In your Developer Portal, these APIs are in a separate Early Adopter section, and you can discover and subscribe to these APIs within your registered application.

In this release of Oracle Hospitality Integration Cloud Service, Oracle is elevating the benefits of the Early Adopter program. Members of the program benefit from the following:

- Early access to version 0 Oracle Hospitality Distribution APIs.
- Best endeavors assistance provided by Oracle Hospitality Integration Cloud Service product management.
- The ability to provide feedback that can potentially influence the product roadmap.

> **Note:**
>
> Please keep to in mind that some of the (v.0) API capabilities offered as part of the early adopter program may be less stable than Production APIs and can experience some difficulties. Furthermore, our aim is to avoid breaking changes in v.0 APIs, however we cannot guarantee full backward compatibility as we roll out patches to improve functionality and/or usability of our early adopter APIs.
>
> Also note that the best endeavors assistance provided by product management as part of the program is not subject to service level agreements (SLAs); however, we do offer a premium (paid) service for partners and/or customers who desire dedicated support. Please contact hospitality-integrations_ww@oracle.com if this is of interest.

As an early API adopter, we look forward to your active insight and feedback as we refine and improve these API capabilities together.

Eligibility for the Early Adopter Program

Anyone who has access to the Oracle Hospitality Integration Platform is eligible to opt in to the Early Adopters Program. However, we have the following expectations from those that do opt in:

- Provide ongoing feedback that can help us improve the APIs.
- Share examples of API calls that worked or didn't work.
- When experiencing issues, provide context that can help us replicate the issue and also understand the business criticality. This information will help us prioritize accordingly.
- Be a bit patient as some of the APIs may be a little bumpy around the edges.

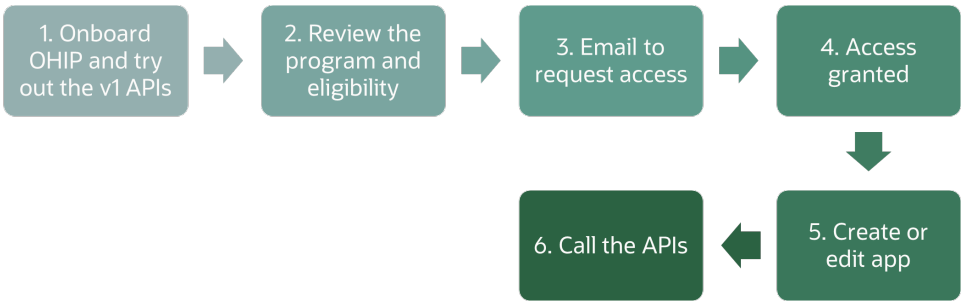
We're open to any feedback but ask that you please maintain a constructive tone. You can opt out at any time if this program is not for you.

How to Opt In to the Early Adopter Program

Opting in is easy. Just send an email to hospitality-integrations_ww@oracle.com and include the following information:

- use case(s)
- desired timelines
- why you think our early adopter APIs will help you be successful
- that you want to opt in to use the Distribution early adopter APIs

Joining the Early Adopter Program



1. Onboard OHIP and try out the v1 APIs. For partners, follow the steps in Quick Start for Partners (Using the Partner Sandbox). For customers, see Getting Started for Hoteliers.
2. Review the Early Adopter API Program, review the APIs in the program, and review the program eligibility and program expectations.
3. Email us at hospitality-integrations_ww@oracle.com.com to request access. In the email, state your use case(s), deadline timelines, and why you think our early adopter APIs will help you. For more information, see How to Opt In to the Early Adopter Program.

   If you meet the eligibility criteria, Oracle will grant you access to the Early Adopter program.
4. Sign in to the Developer Portal.
5. Register or edit an application. At step 10, on the subscription panel, select the **Hospitality APIs** tab and then select the check box. Next, select the **Early Adopter** tab and then select the check box.
6. Call the APIs.

   If you need additional consulting to get you started, you can purchase it in groups of 2 hours. Visit the Oracle Store for more information.

How to provide Early Adopter Program feedback

We anticipate our early adopters to be very active in the development of integrations using our APIs and providing feedback and insight that can help improve and refine our product.

Send an email to hospitality_apis_ww_grp@oracle.com with the following information:

- Ease of use.
- Insights as to what improvements or refinements to make.
- Successful or unsuccessful examples.

3

Register and Manage Applications

Learn how to register and manage applications using the Oracle Developer Portal.

Registering an Application

Register new applications from the Applications page.

1. In the Oracle Hospitality Developer Portal, click the **Applications** tab.
2. Click **Register Application**.
3. On the Register Details panel, enter the application name and an optional description.
4. Enter the **Application Name** that applies to your application.
5. Optionally, add a Description for the application.
6. Optionally, select the Application Type.
7. Select the environment: **Non Production** or **Production**. An application can be either a non production application or a production application.
   - **Non Production**: By default, an application has access only to non production environments. A non production application may not access production environments, so to move from calling non production to calling production environments, you must create a new production application. To access production environments, please email your Oracle Partner Network (OPN) reference number to hgbu_integrations_provisioning_grp@oracle.com.
   - **Production**: Production refers to Oracle Hospitality APIs that can access Oracle Hospitality systems such as a production Oracle Hospitality OPERA Cloud Services environment. A production application may not access non production environments.
8. In the **Contact Information** section, enter your **First Name**, **Last Name**, and **Email Address**. Entering the Email address, Phone Number, and Company Name is mandatory.

   In the event of a major incident with your production integration, Oracle Hospitality will reach out to you using these contact details to inform you of the incident. Oracle Hospitality will temporarily work with you to disable the integration and determine what actions must be taken before the integration can be reenabled. If you do not respond when Oracle Hospitality reaches out to you, Oracle Hospitality will disable your integration. Therefore, it is vital these contact details are monitored 24x7.
9. Click **Next**.
10. On the Subscriptions panel, select the group of Oracle Hospitality APIs to which you want to subscribe this application and then select the **Subscribe** check box.

> **Note:**
>
> ONLY subscribe to API Catalog if your application will be used for Oracle Integration Cloud (OIC) Hospitality Adaptor.

11. Click **View API documentation** to view the API documentation for the Oracle Hospitality APIs to which the application will be subscribed.
12. Click **Register**. The message *Application Registered Successfully* confirms the registration.
13. Your newly registered application appears at the top of the list on the **Applications** tab.

> **Note:**
>
> You are allowed a maximum of 100 applications for registration. You cannot register additional applications if you have reached the allowed limit. If you do so, the following error appears:
>
> **Error: Cannot create your application as you have reached the maximum number of applications allowed.**

Viewing Application Details

When viewing your application details, you can view the application key, the application contact details, application description, application type, and so on. The application key is masked by default.

To view Application Details:

1. On the Oracle Hospitality Developer Portal, click **Applications**.
2. On the All Applications page, select the registered application for which you want details.
3. Click **View details**.

Viewing the Application Key

When in your application details, you can view and copy the application key. The application key is masked by default.

To view and copy your Application key:

1. On the Oracle Hospitality Developer Portal, click **Applications**.
2. On the All Applications page, select the application for which you need the application key.
3. Click **View details**, and then under the Application Key, click **Show** or click **Copy**.

Editing Application Details

Edit an Application from the Application details page:

1. On the Oracle Hospitality Developer Portal, click the **Applications** tab.
2. On the Applications page, click **View details**.
3. On the **Overview** tab, click the **pencil icon**.
4. In the **Edit Application Details** dialog, edit the details, and then click **Save**.

Deleting an Application

> **Note:**
>
> There are some important points to understand when you want to delete an application:
>
> - You cannot delete an application if you have outstanding billing charges.
> - You cannot delete an application that is still subscribed to consume events using the streaming API. Unsubscribe from these events first, then delete the application.
> - Deletion cannot be undone, which means you will permanently remove any historical data and the app key will no longer work. Once you delete your app, you cannot restore it.

If you want delete your application, follow these steps:

1. On the Oracle Hospitality Developer Portal, click the **Applications** tab.
2. On the Applications screen, click **View details**.
3. On the Application Details page, in the **Overview** tab, click the **Delete icon** and click **Delete** again to confirm.

   The following message appears when you click the **Delete icon**:

   “Are you sure you want to delete the [App_Name] application?

   Deleting an application will permanently remove any historical data, and the app key will no longer work. You won't be able to restore it.”

Editing an Application Subscription

To change the APIs to which an application is subscribed:

1. On the Oracle Hospitality Developer Portal, click the **Applications** tab.
2. On the Applications page, click **View details** for the application.
3. Click the **Subscriptions** tab.
4. On the **Subscriptions** tab, click the **pencil icon** to edit.
5. Make your desired edits and click **Save**.

Get a Portable Export of My Application Details

If you need a Portable Export of your Application Details, create a ticket in the Customer Support Portal.

Reissuing an Application Key

A unique application key is created for each application, which must be sent in the header of every request to Oracle Hospitality APIs. You can reissue a key for an application from its Overview page.

> **Important:**
>
> When you reissue a key, requests to the Oracle Hospitality APIs that an application is subscribed to will fail until you send them the new key.

1. On the Oracle Hospitality Developer Portal, click **Applications**.
2. On the Applications screen, select the application for which you want to reissue the key.
3. Click **View Details**.
4. Click the **Show** link to see the unique application key that is created for your application.
5. Click the **Reissue** link.
6. Click **Reissue** at the confirmation message prompt.
7. The reissued application key appears instantly. Click **Hide** to hide the key.

Suspending an Application

If you wish to temporarily suspend your Application, create a ticket in the Customer Support Portal.

After the application is suspended, you can no longer make Oracle Hospitality API calls using that application.

4

Call Usage Alerts

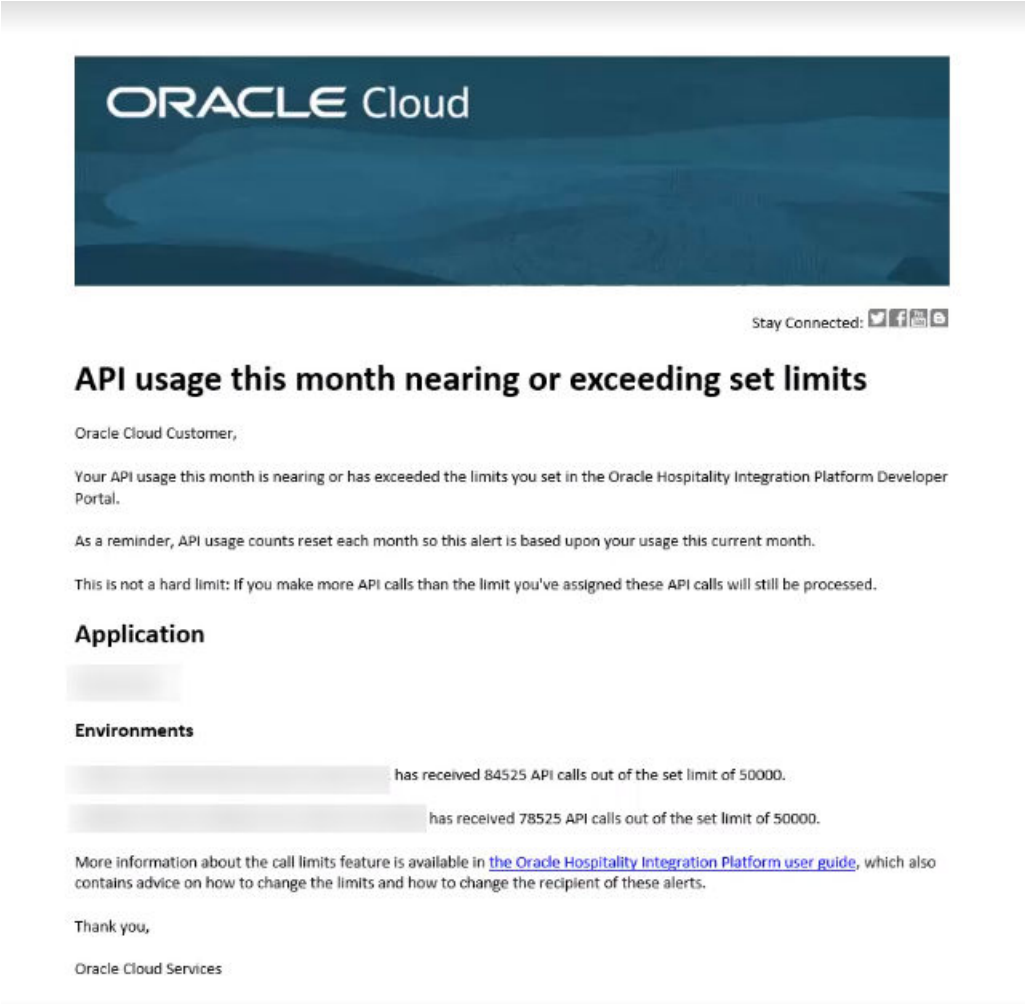
Call usage alerts trigger an email when API usage is near or exceeds the limit assigned for API calls to an environment. On the **Alerts** tab under **Applications**, you can configure the call limit for each application and its environment(s). With call usage alerts, you can do the following:

- Create a new usage alert setting
- Adjust a usage alert setting
- Remove a usage alert setting

Call usage alerts are beneficial to developer portal users concerned about unexpected spikes in usage. To avoid this potential problem you can set call limits, enabling you to better budget and monitor your API calls.

When you approach or exceed your call limit, an email is sent. Only one usage alert email is sent per day per application listing all the environments being called by the application that approach or exceed the assigned usage limit.

Figure 4-1 Usage Alert Email Example



> **Note:**
>
> Usage alerts do not prevent API calls from exceeding the chosen limit and only provide information in an email about the environments and applications that are near or exceed the call limit.

If you are using multiple applications or calling multiple environments, it is recommended to set a low limit for each environment on each application. For calls to non-production environments, a default of 10,000 calls is set for the limit, but no default limit is set for calls to production environments.

You can view your current usage in the Analytics tab.

Configuring Call Usage Alerts

You set call usage limits at the application level. Since an environment can be used for multiple applications, you can set different call usage limits for the same environment.

Setting a Call Usage Limit

1. On the Developer Portal, click **Applications** and then select an application by clicking it.
2. Click the **Alerts** tab.
3. Choose an environment and click the pencil icon for it.
4. Click the up or down arrow for the **Usage Alert Setting** to increase or decrease the usage limit.

   Selecting the up arrow increases the limit and selecting the down arrow decreases it.
5. Click **Save**.

Removing a Call Usage Limit

If you no longer wish to receive usage alert emails, set the **Usage Alert Setting** to zero on each environment for each application.

1. On the Developer Portal, click **Applications** and then select an application by clicking it.
2. Click the **Alerts** tab.
3. Choose an environment and click the pencil icon for it.
4. Click the down arrow for the **Usage Alert Setting** until the call limit is zero.
5. Click **Save**.

Changing the Recipient for Usage Alerts

The email address on the Overview tab of an application is the email address to which usage alerts are sent. This can be changed at any time and takes effect within 24 hours.

1. On the Developer Portal, click **Applications** and then select an application by clicking it.
2. On the **Overview** tab, under **Contact Details**, click the **Question Mark** button.
3. Enter a valid email address for the new recipient.
4. Click **Save**.

5

Environments (Gateways and Credentials) with Resource Owner Group Authentication Scheme (SSD)

From the Environments page in the Developer Portal, partners can view the information (that is, clientId, clientSecret, and gateway URL) required for calling the partner sandbox. Partners who have purchased non-production OPERA Cloud environments can also obtain this information. Once partners are ready to call a hotel's environment, they can add a new environment and obtain the clientId, clientSecret, and gateway URL details.

Customers can also view their environment's clientId, clientSecret, and gateway URL on the Environments page.

Viewing Partner Sandbox Details

1. Open the Developer Portal and click **Environments** at the top of the page. Within the Partner Sandbox section, you can view and copy your Client ID, Client Secret, and Gateway URL.
2. Click the **Generate your integration username and password** link to create your integration user with the specified Tenant ID.

Adding an Environment

Prerequisites for Adding an Environment

- Only partners can add an environment. Customers can by default view their environments. For more information, see Viewing Environment Details.
- The hotel must be using OPERA Cloud Foundation before you can add an environment.
- Partners must create an integration user and have the user approved by the hotel before adding an environment. See Authenticating to Oracle Hospitality Property APIs for steps to create an integration user.

To Add an Environment

1. Open the Developer Portal and click **Environments** at the top of the page.
2. Click **Add Environment**.
3. Select **integration username**.
4. Enter the **Integration Username** for your integration user.
5. Select your **Region**.
6. Select whether the environment you are adding is a **Non Production** or **Production** environment.

7. Click **Add**.

Error Messages

If you encounter an error message when adding an environment, it could be for a variety of reasons, such as the environment owner not yet approving the integration user. The Add Environment page lists the environment errors and recommends the next course of action. If the recommendation is to raise a support request with Oracle Customer Support at the Customer Support Portal, include the exact error message and error code in your support request as this will shorten the resolution time.

If you add back a removed environment, you must use the same integration username that was previously used to add the environment. The error message will inform you which integration username was previously used to add the environment.

Viewing Environment Details

1. From the Developer Portal, click **Environments** at the top of the page. All your environments appear on this page.
2. Click the **View Details** link for the environment you want to view.
3. The following details appear based on the authentication scheme of the customer's environment.
    a. **Gateway URL** for the environment.
    b. **Client ID** and **Secret**. If the Client ID and Client Secret do not appear, you can create these by clicking the **Create Client Credentials** button.
    c. **Authentication scheme** supported by the environment. This would be the Resource Owner Group.
    d. Integration username of the API user.

Viewing the Client Secret

1. From the Developer Portal, click **Environments** at the top of the page.
2. Click the **View Details** link for the environment you want to view.
3. Click the **Show** link for the Client Secret.

If the Client ID and Client Secret do not appear, these can be created by clicking the **Create Client Credentials** button.

Changing Your Client Secret

To change/reissue your Client Secret:

1. From the Developer Portal, click **Environments** at the top of the page.
2. Click the **View Details** link for the environment.
3. Click the **Reissue** link for the Client Secret and click **Reissue** again to confirm.

The following message appears when you click **Reissue**:

“Are you sure you want to reissue the Client Secret? This will affect the following environments: [A list of environments appears]

The current Client Secret will no longer work for the above environment(s).”

Removing an Environment

> **Note:**
>
> - When you remove an environment, it will cease sending any events to which you have subscribed via the streaming API.
> - You can add back the environment, but it will be subject to the customer's approval.
> - If you add back an environment, you must use the same integration username that was previously used to add the environment.

1. From the Developer Portal, click **Environments** at the top of the page. All your environments appear on this page.
2. Click the **View Details** link for the environment you want to remove.
3. Click **Remove Environment** and click **Remove** to confirm.

   The following message appears when you click **Remove Environment**:

   “Are you sure you want to remove this [Environment_Name] environment?

   You can add the environment again, and it will be subject to customer's approval.”

Managing Partner Connections

For customer environments supporting a Resource Owner Group authentication scheme (SSD), partner connections are approved by the chain administrator. This approval grants the WSACCESS role to the integration user.

Approving Partner Connections

When a partner submits a new request to add an integration user via the vendor registration portal, the chain administrator will approve the partner request in SSD after receiving an email from SSD.

Rejecting Partner Connections

To revoke access for a partner, the chain administrator must remove the WSACCESS role from the integration user within SSD.

6

Environments (Gateways and Credentials) with Client Credentials Authentication Scheme (OCIM)

Partners who have purchased non-production OPERA Cloud environments can view the information (that is, clientId, clientSecret, EnterpriseID, Scope and gateway URL) required for calling the environment from the Environments page in the Developer Portal. Once partners are ready to call a hotel's environment, they can add a new environment and obtain the ClientId, ClientSecret, EnterpriseID, Scope, and gateway URL details.

Customers can also view their environment's clientId, clientSecret, EnterpriseID, Scope, and gateway URL on the Environments page.

Sandbox Availability

Partners wanting to integrate with customer environments supporting client credentials-based authentication scheme (OPERA Cloud Identity Management) must work with the respective customer's non production environment.

Adding an Environment

Prerequisites for Adding an Environment

- Only partners can add an environment. Customers can by default view their environments. For more information, see Viewing Environment Details.
- The hotel must be using OPERA Cloud Foundation before you can add an environment.
- Partners must check with customers if they can provide the Enterprise ID.
- For Enterprise ID, you can find it in the environment details of the customer in the customer's OHIP developer portal. See Viewing Environment Details for the steps.

To Add an Environment

1. Open the Developer Portal and click **Environments** at the top of the page.
2. Click **Add Environment**.
3. Select **Enterprise ID**.
4. Enter the **Enterprise ID** and **chain code** received from the customer. An information message is rendered to confirm that the access request is sent for all the gateways of the chain.
5. Select your **Region**.
6. Select whether the environment you are adding is a **Non Production** or **Production** environment.
7. Click **Add**.

The environment request is sent to the customer for approval within the customer developer portal and the status of the environment is "Pending Approval." Once approved, you can view the environment details. For more information, see Viewing Environment Details.

Error Messages

If you encounter an error message when adding an environment, it could be for a variety of reasons, such as the environment owner not yet approving the integration user. The Add Environment page lists the environment errors and recommends the next course of action. If the recommendation is to raise a support request with Oracle Customer Support at the Customer Support Portal, include the exact error message and error code in your support request as this will shorten the resolution time.

If you add back a removed environment, you must use the same Enterprise ID and chain code that was previously used to add the environment. The error message informs you of the Enterprise ID previously used to add the environment.

Viewing Environment Details

1. From the Developer Portal, click **Environments** at the top of the page. All your environments appear on this page.
2. Click the **View Details** link for the environment you want to view.
3. The following details appear based on the authentication scheme of the customer's environment.
    a. **Gateway URL** for the environment.
    b. **Client ID** and **Secret**. If the Client ID and Client Secret do not appear, you can create these by clicking the **Create Client Credentials** button.
    c. **Authentication scheme** supported by the environment. This would be **Client Credentials**.
    d. **EnterpriseID**: This is a unique identifier for the enterprise and must be shared with the partners along with the chain code, so they can add the customer environment for integration.
    e. **Scope**: This scope variable represents the access permissions assigned to the client for the environment.

Issuing the Client Secret

For enhanced security for environments supporting a Client Credentials-based authentication scheme (OPERA Cloud Identity Management), you can only view the client secret once when you issue the secret. Therefore, it is required for customers and partners to note the client secret once it is issued. In case of loss of client secret, a new secret must be issued, and all integrations must be updated with the new client secret.

To issue a new client secret:

1. From the Developer Portal, click **Environments** at the top of the page.
2. Click the **View Details** link for the environment you want to view.
3. Click the **Issue** link for the Client Secret.

4. Click the **Issue** button on the confirmation message, which indicates the environments that will be affected once the new client secret is issued.
5. Click the **Copy** button to copy the new client secret.

If the Client ID and Client Secret do not appear, you can create them by clicking the **Create Client Credentials** button.

Changing Your Client Secret

To change/reissue your Client Secret:

1. From the Developer Portal, click **Environments** at the top of the page.
2. Click the **View Details** link for the environment.
3. Click the **Issue** link for the Client Secret.
4. Click the **Issue** button on the confirmation message, which indicates the environments affected once the new client secret is issued.
5. Click the **Copy** button to copy the new client secret.

Removing an Environment

> **Note:**
>
> - When you remove an environment, it will cease sending any events to which you have subscribed via the streaming API.
> - You can add back the environment, but it will be subject to the customer's approval.
> - If you add back an environment, you must use the same enterprise ID and chain code that was previously used to add the environment.

1. From the Developer Portal, click **Environments** at the top of the page. All your environments appear on this page.
2. Click the **View Details** link for the environment you want to remove.
3. Click **Remove Environment** and click **Remove** to confirm.

   The following message appears when you click **Remove Environment**:

   “Are you sure you want to remove this [Environment_Name] environment?

   You can add the environment again, and it will be subject to customer's approval.”

Managing Partner Connections

For customer environments supporting a Client Credentials authentication scheme (OCIM), the partner connection requests take place in the OHIP (Partner) Developer Portal and approvals take place within the OHIP (customer) developer portal.

Approving Partner Connections

When a partner submits a new request to add a customer environment, follow the below steps in the OHIP customer portal to approve the partner's connection request.

1. From the Developer Portal, click **Environments** at the top of the page.
2. Click **Partner Connections**. All the partner connections appear in this section.
3. Search for the partner connection that is pending approval.
4. Click **Approve** to approve the partner connection request. The status of the partner connection request should change to "Approved" in both the customer and partner portal.

Rejecting Partner Connections

When a partner submits a new request to add a customer environment, follow the below steps in the OHIP customer portal to reject the partner's connection request.

1. From the Developer Portal, click **Environments** at the top of the page.
2. Click **Partner Connections**. All the partner connections appear in this section.
3. Search for the partner connection that is pending approval.
4. Click **Reject** to reject the partner connection request. The status of the partner connection request should change to "Rejected" in the Environment section of both the customer and partner portal.

7

Migrating to Client Credentials Authentication Scheme (OCIM)

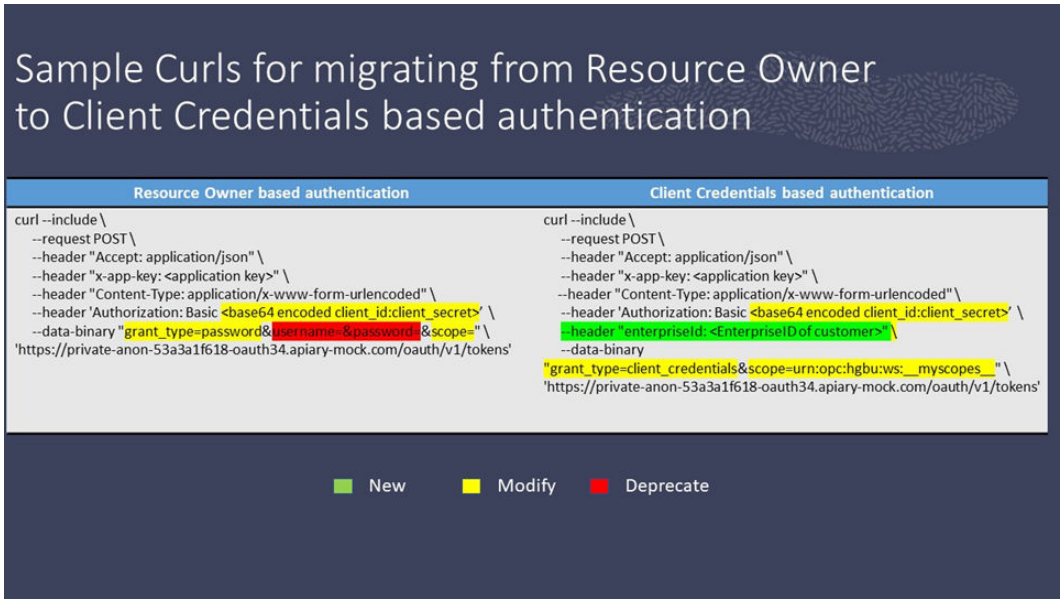
Follow the below steps only for environments that are migrated from a resource owner group (SSD) authentication scheme to client credentials authentication scheme (OPERA Cloud Identity Management).

Stage 1: Preparing for Migration (Pre-Migration)

In this stage, the customer and partner environments are marked with a label ‘Migrating Soon.’ This implies that customers and partners can now prepare for the upcoming migration, so they have a seamless transition with minimal downtime.

Customers and their partners are advised to take the following actions:

1. Reissue their client secret so it also works with a client credentials authentication scheme (OPERA Cloud Identity Management). For more information, see Changing Your Client Secret.
2. The below image depicts the changes you must make in the OAuth token API. These changes enable you to use the client credentials authentication scheme (OPERA Cloud Identity Management) once your environment completes the migration. For more information, see Authenticating to Oracle Hospitality Property APIs.



> **Note:**
>
> Customers and partners must use the resource owner group authentication (SSD) to get OAuth tokens for their integration and should NOT switch to the new authentication code until the migration is complete. The pre-migration stage is only to give sufficient time to prepare the authentication code for client credentials and to reissue the client secrets.

Stage 2: Migration

Customers and their partners are informed well in advance of the time for the actual migration by the Customer Success manager. During the migration phase, customers and partners will temporarily be unable to access their integrations for a short period of time.

Stage 3: Post Migration Validation

The customer success manager informs the customer and their partners once the migration has been completed. Customers and their partners are advised to follow the below instructions to ensure a successful migration.

1. **OHIP Developer Portal**: Customers must check if users can still access the OHIP Developer Portal. If not, customers must grant access to the user by adding the developerportalaccess role in OPERA Cloud Identity Management. For more information, see Adding Developer Portal Users.

   > **Note:**
   >
   > Developer Portal URL will change after migration to OPERA Cloud Identity Management. Customers need to login using the new URL provided by CSM via customer communication.

2. **Client Credentials**: The clientId for existing OHIP integrations will not be migrated. If not regenerated during the pre-migration step, customers and partners can still regenerate client credentials post migration. For more information, see Changing Your Client Secret.
3. **Integration Users**: Integration users are not needed for Client Credentials authentication.
4. **Authentication Scheme**: The authentication scheme in all the migrated customer and partner environments should be changed to Client Credentials. For more information, see Viewing Environment Details.
5. **Environment Details**: All environment details should now be reflecting two new fields — ‘EnterpriseID’ and ‘Scope’ — which are required when authenticating with Client Credentials-based authentication. For more information, see Viewing Environment Details.
6. **Existing Partner Connections**: All partner connections should automatically be migrated to a Client Credentials authentication scheme and should be in "Approved" status. This can be validated in the OHIP developer portal.

7. Customers and their partners must validate if the below existing data is visible post migration “as is” in the developer portal.
    a. Applications
    b. Streaming applications and their configuration
    c. Restricted APIs access
    d. Analytics
8. **Switch Authentication**:
   Customers and their partners must regenerate the OAuth token using the client credentials workflow and replace all their old tokens with the new token for existing integrations. For more information, see Authenticating to Oracle Hospitality Property APIs.

8

Using the Oracle Hospitality APIs

Having created an application and obtained the gateway URL from the portal, calling APIs is a four-step process:

1. Obtain details from the hotel. In the case of the partner sandbox, the hotel code is SAND01.
2. Add an environment. See Environments Gateways and Credentials for details.
3. **Authenticate**: Obtain an oAuth token using the oAuth API in a call sent to the gateway URL.
4. **Call APIs**: Send your API calls to the gateway URL following the API documentation displayed in the portal.

Oracle Hospitality Property APIs with Resource Owner Group Authentication (SSD)

This section contains the following topics:

Obtaining Details from the Hotel

OPERA Cloud resources are available only via the Oracle Hospitality Integration Platform if the hotel company is using OPERA Cloud Foundation. You will need the following pieces of information from the hotel whose data you are accessing:

- **Production or Non Production**: Many hotels run non production OPERA Cloud environments as well as their production environment(s). Determine whether the OPERA Cloud environment you are calling is a production or non production instance.
- **Region**: The region in which the hotel's OPERA Cloud environment is running.
- **Tenant / Chain Code**: The hotel's tenant code used to target the right tenant especially in OPERA Cloud environments that host multiple tenants.
- **Hotel ID**: The OPERA internal ID of a hotel in the customer's tenant.

  For environments supporting the Resource Owner authentication scheme, the below details must be obtained:

- **SSD URL**: The URL on which to create an integration user.

Follow these steps to obtain this URL:

1. Ask the customer for their OPERA login page URL.
2. Go to this customer provided URL.
3. Click the **Register new account** button on the OPERA login page to arrive at the OPERA Shared Security Domain (SSD), which has a URL like the following:

https://rp15-prod2-ssd-ohs.oracleindustry.com/identity/faces/register?_afrLoop=4294503438550013&_afrWindowMode=0&_afrWindowId=null&_adf.ctrl-state=segh9ko67_1

4. Change this URL to a format similar to the following example by replacing the section after the question mark with "apiuser=y". This is the URL for creating an integration user.

   https://rp15-ssd-ohs.oracleindustry.com/identity/faces/register?apiuser=y

Authenticating to Oracle Hospitality Property APIs

You can call Oracle Hospitality APIs using different authentication schemes depending on the scheme supported by your environment. The authentication scheme supported by your environment can be found on the environments card under the Environments tab. For more information, see Viewing Environment Details.

To call Oracle Hospitality APIs using the Resource Owner-based authentication scheme, you need the following pieces of information:

1. **Integration username and password**: A username and password for an integration account in Oracle Hospitality Shared Security Domain granted by a hotel administrator to permit access to their data. This is one of the pieces of information needed to obtain the oAuth token. You can request this through the Oracle Hospitality Shared Security Domain (SSD) partner self-service registration portal. Completing and submitting this form sends a partner registration request to the OPERA customer for approval.

   > **Note:**
   >
   > Every partner must have a unique integration user. This integration user is unique to one OPERA Tenant/Chain. Therefore, if you are accessing multiple OPERA Tenants/Chains, you need to request multiple integration users.
   >
   > Please ensure you are signed out of the Oracle Hospitality Shared Security Domain (SSD) before requesting your integration user. If you are signed out, ‘Anonymous’ appears at the top right of the screen.

   - Go to the SSD URL you obtained from the customer. For more information, see Obtaining Details from the Hotel.
   - Specify the **Tenant** /Chain code as advised by the hotel, or else enter hotel code(s) advised by the hotel.
   - Avoid spaces in the **Vendor** name field and keep the name to a maximum of 10 characters.
   - Once the hotel approves your integration user, you will receive an email.
   - For more information on this step, refer to the *Partner Self-Registration Guide* attached to the following Customer Support Portal document:

     Oracle Hospitality Shared Security Domain (SSD) for OPERA

> **Note:**
>
> Please note and keep secure your generated interface ID and key. The interface ID is your integration user's username, and the interface key is your integration user password.

2. **OAuth Client ID and Client Secret**: These grant you access to the identity servers (Oracle Hospitality Shared Security Domain). See Environments (Gateways and Credentials) for steps on how to view or add these.
3. **Application key**: A unique application key is created for each application upon registering an application in the Oracle Hospitality Developer Portal. This key must be passed in the "x-app-key" header of every request to the Oracle Hospitality APIs. To obtain your application key, refer to Viewing the Application Key.

> **Note:**
>
> A new application and its corresponding application key are required when you move to Production.

Only follow the below steps if this authentication scheme appears on the **Environments** tab.

1. To obtain your oAuth token, use the above details in the 'Authenticate with the Identity Server' call in the oAuth API. You can find this description in the oAuth API on the APIs page of the Developer Portal.
2. Send this oAuth request to the gateway URL listed in the Developer Portal.

> **Note:**
>
> oAuth tokens expire after 1 hour (60 minutes). Currently, we do not support refresh tokens.

Calling Oracle Hospitality Property APIs

All Oracle Hospitality APIs expect the following mandatory headers:

1. **Token**: A header "Authorization" that includes the oAuth token obtained through Authenticating to Oracle Hospitality APIs.
2. **x-hotelid**: Supply a hotel (or property) code from the chain to which you have access in a header "x-hotelid." This is the Hotel ID you obtained from the hotel. If you do not know your x-hotelid, contact Oracle Customer Support at https://iccp.custhelp.com.
3. **Application key**: An unique application key is created for each application upon registering an application in the Oracle Hospitality Developer Portal. This key must be passed in the "x-app-key" header of every request to the Oracle Hospitality APIs. To obtain your application key, refer to Viewing the Application Key

   .

Required Headers:

- x-app-key:<ApplicationKey>
- x-hotelid:<HotelId>
- Authorization: Bearer <oAuth Token>

Optional Headers:

- x-hubid — When you have hub-level access, you can supply the x-hubid to retrieve multiple properties.
- x-externalSystem — If you have an external system in an OPERA environment (for example, for use with streaming or polling), then specifying that external system in the header of inbound API calls will avoid echoing those changes back to the streaming and polling APIs. For the streaming APIs, the external system code is shown on the application's Events Subscribed tab.

Example

GET https://<Gateway Domain>/lov/v1/listOfValues/Titles

The below table lists the environment variables you can add to the Postman Environment.

Table 8-1 Postman Environment Variables

Variable	Value
AppKey	This value is accessed from the Developer Portal.
ClientId	This value is accessed from the Developer Portal.
ClientSecret	This value is accessed from the Developer Portal.
HotelId	This value is supplied by the hotel.
Password	This value is accessed from the "Interface Key" in the vendor self-registration portal.
Username	This value is accessed from the "Interface Id" in the vendor self-registration portal.

API Throttling

The Oracle Hospitality Integration Platform (OHIP) APIs are throttled. Throttling limits the number of concurrent calls to OPERA Cloud to avoid impacting day-to-day hotel operations. If more than 50 requests per second are received for a single OPERA Cloud environment, throttling helps maintain the day-to-day running of the hotel while still enabling API traffic. The 51st and subsequent requests received in a single second will be delayed by 1000 milliseconds. This throttling limit may be changed to protect day-to-day hotel operations. We provide advance warning of changes when possible. If you are frequently hitting the throttling limit, consider the following:

- For Shop & Book related calls, consider Distribution APIs (see Oracle Hospitality Distribution APIs for more information).
- Retrieve more coarse grained data in a single call rather than performing filtering in the client.

- Reduce the frequency of calls.
- Cache data that changes infrequently, such as Lists of Values.
- Review the API release notes for new operations that could reduce the frequency of API calls (for example, asynchronous (jobbed) operations).
- Consider Connecting to the Streaming API as an alternative to the Polling API.

Oracle Hospitality Property APIs with Client Credentials Authentication (OCIM)

This section contains the following topics:

Obtaining Details from the Hotel

OPERA Cloud resources are available only via the Oracle Hospitality Integration Platform if the hotel company is using OPERA Cloud Foundation. You will need the following pieces of information from the hotel whose data you are accessing:

- **Production or Non Production**: Many hotels run non production OPERA Cloud environments as well as their production environment(s). Determine whether the OPERA Cloud environment you are calling is a production or non production instance.
- **Region**: The region in which the hotel's OPERA Cloud environment is running.
- **Tenant / Chain Code**: The hotel's tenant code used to target the right tenant especially in OPERA Cloud environments that host multiple tenants.
- **Hotel ID**: The OPERA internal ID of a hotel in the customer's tenant.
- **Enterprise ID**: The unique identifier for the enterprise. This can be found in the environment details of the customer in the customer's OPERA Cloud Identity Management portal. For more information, see Viewing Environment Details.

Authenticating to Oracle Hospitality Property APIs

You can call Oracle Hospitality APIs using different authentication schemes depending on the scheme supported by your environment. The authentication scheme supported by your environment can be found on the environments card under the Environments tab. For more information, see Viewing Environment Details.

To call Oracle Hospitality APIs using the Client Credential based authentication scheme, you need the following pieces of information:

1. **OAuth Client ID and Client Secret**: These grant you access to the identity servers (OPERA Cloud Identity Management). See Environments (Gateways and Credentials) for steps on how to view or add these.
2. **Application Key**: A unique application key is created for each application upon registering an application in the Oracle Hospitality Developer Portal. This key must be passed in the "x-app-key" header of every request to the Oracle Hospitality APIs. To obtain your application key, refer to the Viewing the Application Key topic.

> **Note:**
>
> A new application and its corresponding application key are required when you move to Production.

3. **Scope**: A static variable that represents the authorization scope and permissions granted to the client for accessing the APIs from OHIP.
4. **Enterprise ID**: A unique ID for the enterprise for which the client is created.

To obtain your oAuth token, use the above details in the 'Authenticate with the Identity Server' call in the oAuth API. You can find this description in the oAuth API on the APIs page of the Developer Portal.

Send this oAuth request to the gateway URL listed in the Developer Portal.

> **Note:**
>
> oAuth tokens expire after 1 hour (60 minutes). Currently, we do not support refresh tokens.

Calling Oracle Hospitality Property APIs

All Oracle Hospitality APIs expect the following mandatory headers:

1. **Token**: A header "Authorization" that includes the oAuth token obtained through Authenticating to Oracle Hospitality APIs.
2. **x-hotelid**: Supply a hotel (or property) code from the chain to which you have access in a header "x-hotelid." This is the Hotel ID you obtained from the hotel. If you do not know your x-hotelid, contact Oracle Customer Support at https://iccp.custhelp.com.
3. **Application key**: An unique application key is created for each application upon registering an application in the Oracle Hospitality Developer Portal. This key must be passed in the "x-app-key" header of every request to the Oracle Hospitality APIs. To obtain your application key, refer to Viewing the Application Key

   .

Required Headers:

- x-app-key:<ApplicationKey>
- x-hotelid:<HotelId>
- Authorization: Bearer <oAuth Token>

Optional Headers:

- x-hubid — When you have hub-level access, you can supply the x-hubid to retrieve multiple properties.
- x-externalSystem — If you have an external system in an OPERA environment (for example, for use with streaming or polling), then specifying that external system in the header of inbound API calls will avoid echoing those changes back

to the streaming and polling APIs. For the streaming APIs, the external system code is shown on the application's Events Subscribed tab.

Example

GET https://<Gateway Domain>/lov/v1/listOfValues/Titles

The below table lists the environment variables you can add to the Postman Environment.

Table 8-2 Postman Environment Variables

Variable	Value
AppKey	This value is accessed from the Developer Portal.
ClientId	This value is accessed from the Developer Portal.
ClientSecret	This value is accessed from the Developer Portal.
HotelId	This value is supplied by the hotel.
Password	This value is accessed from the "Interface Key" in the vendor self-registration portal.
Username	This value is accessed from the "Interface Id" in the vendor self-registration portal.

API Throttling

The Oracle Hospitality Integration Platform (OHIP) APIs are throttled. Throttling limits the number of concurrent calls to OPERA Cloud to avoid impacting day-to-day hotel operations. If more than 50 requests per second are received for a single OPERA Cloud environment, throttling helps maintain the day-to-day running of the hotel while still enabling API traffic. The 51st and subsequent requests received in a single second will be delayed by 1000 milliseconds. This throttling limit may be changed to protect day-to-day hotel operations. We provide advance warning of changes when possible. If you are frequently hitting the throttling limit, consider the following:

- For Shop & Book related calls, consider Distribution APIs (see Oracle Hospitality Distribution APIs for more information).
- Retrieve more coarse grained data in a single call rather than performing filtering in the client.
- Reduce the frequency of calls.
- Cache data that changes infrequently, such as Lists of Values.
- Review the API release notes for new operations that could reduce the frequency of API calls (for example, asynchronous (jobbed) operations).
- Consider Connecting to the Streaming API as an alternative to the Polling API.

Oracle Hospitality Asynchronous APIs

For long-running operations such as adding, updating, or retrieving large amounts of data, there is a series of Oracle Hospitality Property APIs called asynchronous ("async") APIs. These all work the same way with specific operation details available in the Developer Portal API documentation. Simply search for "Async" in the APIs tab.

Certain business use cases also require Asynchronous APIs, such as the Revenue Management System business use case, so refer to those specific business use cases for details.

Business Context

The adoption of Revenue Management has become more prevalent among properties aiming to optimize the value of each room. The asynchronous Property APIs offer a means to collect and update bulk data efficiently. There are two distinct approaches to retrieve data from OPERA Cloud through asynchronous Property APIs and business event-driven Property APIs. The asynchronous Property APIs are specifically designed for revenue management systems, allowing smooth updates of bulk data like inventory, restrictions, and room rates in OPERA Cloud. These APIs are fully compatible with OPERA Cloud versions 23.1.x and above.

Creating the External System

All asynchronous operations require you to create an external system. This is so Oracle Hospitality OPERA Cloud knows for whom to process the request. The external system code is in the path of every Asynchronous API.

If you already have an external system created by the OPERA environment owner, ask them to check whether step 4–d is complete. If complete, you can use this for the Asynchronous APIs.

Only the (Customer) OPERA environment owner can create an external system. They should follow this process:

1. Verify a chain-level user has the below tasks assigned. To assign these tasks, follow the steps in the Assigning Tasks to a Role topic in the OPERA Cloud User Guide.
    a. Navigate to **Interfaces Admin**, **Property Interfaces** and **External Systems**.
    b. Select **New/Edit External Systems**.
2. Log in at the Hub level (not the property level).
3. From the Administration menu, select **Interfaces**, select **Business Events**, and then select **External Systems**.
4. Click **New** and enter the following details:
    a. **Code**: Enter a code for the external system.
    b. **Description**: Enter a name of the external system.
    c. **Sequence**: Specify sequence in which the external system will display in the External Systems list.
    d. For each property in the chain, enter or select the **Property** and click **Active** to activate it.
5. Click **Save**.

    Make sure you remember the external system code you created.
6. Communicate to the integrator the external system code created at step 4.

    If customers have any challenges with creating the external system, they should either contact Oracle Customer Support at the Customer Support Portal (in case of a technical error) or contact Oracle Hospitality Consulting Services if additional help is needed.

Prerequisites

Tools

- Postman for analyzing / testing of the APIs

Environment

- OPERA Cloud version 23.1.x and above

Configurations

All asynchronous operations require you to create an external system. This ensures that Oracle Hospitality OPERA Cloud knows the intended recipient of the request. The external system code is in the path of every Asynchronous API. For details on the how to configure external systems, refer to the Creating the External System topic.

Workflow

Using the Asynchronous APIs

The Asynchronous APIs involve a three-step data flow process, and it is imperative to execute all three steps without skipping any steps. Refer to Best Practices for additional guidelines.

1. The **POST** request is the first step from an external system to OPERA Cloud, which can do one of the following:
   - **POST Data to OPERA Cloud**: This starts a process to accept the data into OPERA Cloud.
   - **Fetch Data from OPERA Cloud**: This starts a process to retrieve data from OPERA Cloud.

   Once you have sent this post request to OPERA Cloud, you should receive an HTTP 202 Accepted response if the request is valid. The response header parameter 'location' provides a URL that contains a request ID. This ID is required in step 2.

   > **Note:**
   >
   > Refer to Types and Recommendations to learn more about the associated API limits. If the bulk data to be posted to or retrieved from OPERA Cloud does not align with the API specifications, a validation error is generated.

2. The **HEAD** request is the second step from an external system to OPERA Cloud to check the status of the process started with the POST request in the first step. Use the header parameter '**location**' from the POST response in this HEAD request. Once the process is completed, the HEAD response returns a header parameter 'status' with value 'COMPLETED.' Similar to step 1, the header parameter 'location' returns a URL containing a request ID that is required in step 3.
3. The **GET** request is the third step from an external system to OPERA Cloud to either collect the bulk data or confirm the post of data was successful. Use the ID in the location

header URL returned by the HEAD response in step 2. The GET response provides the requested data or log specifics, particularly if you have added data, and indicates any potential failures.

Figure 8-1 Scenario 1: Customer/Partner obtains the intended result

Figure 8-2 Scenario 2: Invalid request

Sample Workflow

For a sample workflow, this example uses the operation startRevenueInventoryStatistics.

1. **POST Request:** startRevenueInventoryStatisticsProcess

```
curl --location '{{HostName}}/inv/async/v1/externalSystems/
{{ExtSystemCode}}/hotels/{{HotelId}}/revenueInventoryStatistics' \
--header 'Content-Type: application/json' \
--header 'x-app-key: {{AppKey}}' \
--header 'x-hotelId: {{HotelId}}' \
--header 'Authorization: Bearer {{Token}}' \
--data '{
    "dateRangeStart": "YYYY-MM-DD",
    "dateRangeEnd": "YYYY-MM-DD",
    "roomTypes": [
        ""
    ]
}'
```

POST Response Headers:

HTTP Status: 202 Accepted

Key	Value
Date	Sat, 23 Dec 2023 03:59:40 GMT
Content-Length	0
Connection	keep-alive
X-Powered-By	Express
cache-control	no-cache, no-store
location	{{HostName}}/inv/async/v1/externalSystems/{{ExtSystemCode}}/hotels/{{HotelId}}/revenueInventoryStatistics/<**ID1**

Key	Value
path	startRevenueInventoryStatisticsProcess
vary	origin
Strict-Transport-Security	max-age=31536000; includeSubDomains
X-Content-Type-Options	nosniff
X-XSS-Protection	1; mode=block
X-Frame-Options	SAMEORIGIN

2. **HEAD request**: getRevenueInventoryStatisticsProcessStatus

The ID (**ID1** in Step 1) from the location URL of the POST Response Header parameter is fed to the HEAD request.

```
curl --location --head '{{HostName}}/inv/async/v1/externalSystems/
{{ExtSystemCode}}/hotels/{{HotelId}}/revenueInventoryStatistics/{{ID1}}' \
--header 'Content-Type: application/json' \
--header 'x-app-key: {{AppKey}}' \
--header 'x-hotelId: {{HotelId}}' \
--header 'Authorization: Bearer {{Token}}' \
--data ''
```

HEAD Response Headers:

HTTP Status: 201 Created

Key	Value
Date	Sat, 23 Dec 2023 03:59:40 GMT
Content-Length	0
Connection	keep-alive
X-Powered-By	Express
cache-control	no-cache, no-store
location	{{HostName}}/inv/async/v1/externalSystems/{{ExtSystemCode}}/hotels/{{HotelId}}/revenueInventoryStatistics/<**ID2**
path	getRevenueInventoryStatisticsProcessStatus
status	COMPLETED
vary	origin
Strict-Transport-Security	max-age=31536000; includeSubDomains
X-Content-Type-Options	nosniff
X-XSS-Protection	1; mode=block
X-Frame-Options	SAMEORIGIN
Referrer-Policy	no-referrer
Pragma	no-cache
Access-Control-Expose-Headers	Set-Cookie
Set-Cookie	operaEntId=<EntId>;Path=/;Secure;SameSite=None

Key	Value
Set-Cookie	tenant=<tenantId>Path=/;Secure;SameSite=None

3. GET request: getRevenueInventoryStatistics

Similar to step 1, the ID (**ID2** in step 2) from the location URL of the HEAD Response Header parameter is fed to the GET request.

```
curl --location '{{HostName}}/inv/async/v1/externalSystems/
{{ExtSystemCode}}/hotels/{{HotelId}}/revenueInventoryStatistics/
{{ID2}}' \
--header 'Content-Type: application/json' \
--header 'x-app-key: {{AppKey}}' \
--header 'x-hotelId: {{HotelId}}' \
--header 'Authorization: Bearer {{Token}}' \
--data ''
```

GET Response Body:

HTTP Status: 200 OK

A sample successful response body for the operation getRevenueInventoryStatistics is shown below:

```
 "revInvStats": [
        {
            "property": "<HotelID>",
            "occupancyDate": "YYYY-MM-DD",
            "physicalRooms": "0",
            "noShowRooms": "0"
        },
        {
            "property": "<HotelID>",
            "occupancyDate": "YYYY-MM-DD",
            "physicalRooms": "0",
            "noShowRooms": "0"
        },
        {
            "property": "<HotelID>",
            "occupancyDate": "YYYY-MM-DD",
            "physicalRooms": "0"
        }
    ]
}
```

Best Practices

- Asynchronous APIs allow a maximum of 150 requests per minute for each application, encompassing POST, HEAD, and GET requests. Take this into account when deciding the polling frequency for HEAD requests.
- After sending a POST request, allow a waiting period of at least 1-2 minutes before initiating the HEAD request.

- When sending a HEAD request, wait for the **HTTP status 201 Created** response before proceeding with the GET request.
- Remember that multiple retries of the HEAD call might be necessary, depending on the volume of data being returned.

Types and Usage Recommendations

The currently supported Async API operations are listed below.

The Asynchronous APIs can be classified into two types according to how they interact with OPERA Cloud:

1. Operations that fetch Data from OPERA Cloud. This is applicable to the following operations along with the limits specified below:

Operation ID	Functionality	Limit	OPERA Cloud Version Dependencies (if any)	Recommendations
Module: Inventory				

Operation ID	Functionality	Limit	OPERA Cloud Version Dependencies (if any)	Recommendations
startRevenueInventoryStatisticsProcess (POST) getRevenueInventoryStatisticsProcessStatus (HEAD) getRevenueInventoryStatistics (GET)	• Enables you to retrieve Revenue Inventory Statistics • You can use this API to fetch revenue inventory statistics for past, present, and future reservations from OPERA Cloud. You will be able to filter using stay date (with a start and end date) to fetch inventory data.	• Every identical request (employing the same query parameters) necessitates a mandatory 30-minute interval between submissions. Please note that this limit applies regardless of whether the POST, HEAD, and GET cycle has been completed. • Current date range limit: 400 days (Refer to the **OPERA Cloud Version Dependencies** column for more details.)	• With the OPERA Cloud versions 22.1 and higher, the getRevenueInventoryStatistics API response includes values for roomSold, projected roomRevenue, roomArrival, roomDeparture, and projected totalRevenue for reservations with arrival date same as the property business date. • With the OPERA Cloud versions 22.5 and higher, every identical request (employing the same query parameters) necessitates a mandatory 30-minute interval between submissions. Please note that this limit applies regardless of whether the POST, HEAD, and GET cycle has been completed.	• This API is suitable for retrieving future inventory data and calculating net revenue. You can group the data either by Room Type, Market Code, or Guarantee Code (Reservation Type) or you can gather data per date for the entire hotel.

Operation ID	Functionality	Limit	OPERA Cloud Version Dependencies (if any)	Recommendations
			• With the OPERA Cloud versions 23.1 and higher, the getRevenueInventoryStatistics API response expiration time has been reduced from 24 to 6 hours. • With the OPERA Cloud versions 23.2 and higher, the maximum date range allowed per request is extended from 94 to 400 days.	
Module: Blocks				

Operation ID	Functionality	Limit	OPERA Cloud Version Dependencies (if any)	Recommendations
startBlockAllocationSummaryProcess (POST) getBlockAllocationSummaryProcessStatus (HEAD) getBlockAllocationSummary (GET)	• Enables you to retrieve Block Allocation Summary • You can use this API to fetch Block allocation information for a hotel and specified date range. The block allocated inventory, rates, and room type statistics, including revenue, are returned as part of the response.	• Every identical request (employing the same query parameters) necessitates a mandatory 30-minute interval between submissions. Please note that this limit applies regardless of whether the POST, HEAD, and GET cycle has been completed. • The maximum date range allowed per request is 94 days. • When requesting data with startLastModifiedDate/ endLastModifiedDate, subsequent identical requests (with the same query parameters) must be spaced 3 hours apart. Note that this limit only applies to identical POST requests made without completing the entire POST, HEAD,	• With the OPERA Cloud versions 21.5 and higher, the getBlockAllocationSummary API response will include values for ownerCode, cutOffDays, cutOffDate, blockType, all status blocks, cancellation Date, and extraPerson values. • With the OPERA Cloud versions 22.1 and higher, the getBlockAllocationSummary API response has been enhanced to include block daily and total revenue statistics. • With the OPERA Cloud versions 22.4 and higher, a new search filter lastModifiedDate has been added to the startBlockAllocationSummaryProcess API operation to get the data modified	

Operation ID	Functionality	Limit	OPERA Cloud Version Dependencies (if any)	Recommendations
		and GET cycle.	after the previous fetch request. • With the OPERA Cloud versions 22.4 and higher, when requesting data with startLastModifiedDate/ endLastModifiedDate, subsequent identical requests (with the same query parameters) must be spaced 3 hours apart. Note that this limit only applies to identical POST requests made without completing the entire POST, HEAD, and GET cycle. • With the OPERA Cloud versions 22.5 and higher, every identical request (employing the same query parameters) necessitates a mandatory 30-minute interval between submissions.	

Operation ID	Functionality	Limit	OPERA Cloud Version Dependencies (if any)	Recommendations
			Please note that this limit applies regardless of whether the POST, HEAD, and GET cycle has been completed. • With the OPERA Cloud versions 23.1 and higher, the getblockAllocationSummary API response expiration time has been reduced from 24 to 6 hours. • With OPERA Cloud versions 23.3 and higher, the getblockAllocationSummary API response will include origin code.	
Module: Reservations				

Operation ID	Functionality	Limit	OPERA Cloud Version Dependencies (if any)	Recommendations
startReservationsDailySummaryProcess (POST) getReservationsProcessStatus (HEAD) getReservationsDailySummary (GET)	• Enables you to retrieve Reservation Daily Summary • This API allows external systems to retrieve a summary of reservations for a specified hotel and date range.	• Every identical request (employing the same query parameters) necessitates a mandatory 30-minute interval between submissions. Please note that this limit applies regardless of whether the POST, HEAD, and GET cycle has been completed. • The maximum date range allowed per request is 94 days (whether it is in the past or the future). • When requesting data with startLastModifiedDate/ endLastModifiedDate, subsequent identical requests (with same query parameters) will have to be spaced 3 hours apart. Note that this limit only applies to identical POST requests made	• With the OPERA Cloud versions 22.4 and higher, every identical request (employing the same query parameters) necessitates a mandatory 30-minute interval between submissions. Please note that this limit applies regardless of whether the POST, HEAD, and GET cycle has been completed. • With the OPERA Cloud versions 22.4 and higher, a new search filter lastModifiedDate has been added to the startReservationsDailySummaryProcess operation to get the data modified after the previous fetch request. • With the OPERA Cloud versions 22.4 and higher, when	• This API is suitable to get a summary of all reservations at a hotel when onboarding a new property with your integration. • Data can be requested using either startDate/ endDate or startLastModifiedDate/ endLastModifiedDate. It is recommended to use the startLastModifiedDate and endLastModifiedDate parameters to retrieve incremental updates since the last call. • When requesting data with startLastModifiedDate/ endLastModifiedDate, it is recommended to complete all sets of calls (POST, HEAD, and GET) before making new data requests. • Note that if there are no reservations

Operation ID	Functionality	Limit	OPERA Cloud Version Dependencies (if any)	Recommendations
		without completing the entire POST, HEAD, and GET cycle.	requesting data with startLastModifiedDate/ endLastModifiedDate, subsequent identical requests (with same query parameters) will have to be spaced 3 hours apart. Note that this limit only applies to identical POST requests made without completing the entire POST, HEAD, and GET cycle. • With the OPERA Cloud versions 22.4 and higher, the getReservationdailySummary API response will include a reservation's net room rate amount. • With the OPERA Cloud versions 23.1 and higher, the getReservationsDailySummary API response expiration time has been	that match the specified criteria for the date range (fixed dates or last modified), it is possible to receive an empty response, indicating no reservations found.

Operation ID	Functionality	Limit	OPERA Cloud Version Dependencies (if any)	Recommendations
			reduced from 24 to 6 hours.	

2. Operations that POST Data to OPERA Cloud. This is applicable to the following operations along with the limits specified below:

Operation ID	Functionality	Limit	OPERA Cloud Version Dependencies (if any)	Recommendations
Module: Rate Plan				

Operation ID	Functionality	Limit	OPERA Cloud Version Dependencies (if any)	Recommendations
startSetDailyRatePlanSchedulesProcess (POST) getDailyRatePlanSchedulesProcessStatus (HEAD) getDailyRatePlanSchedules (GET)	• Enables you to create Daily Rate Plan Schedules • You can use this API to add and/or update the rate price schedule to existing OPERA Daily Rate plans.	• Every identical request (employing the same query parameters) necessitates a mandatory 30-minute interval between submissions. Note that this limit applies regardless of whether the POST, HEAD, and GET cycle has been completed. • While you can update multiple Daily Rate Plan schedules for a given property, you can only update rates for only one property at a time. Refer to About Rate Codes in the OPERA Cloud Services User Guide for more information on rate structures. • You can update rates up to 180 days into the future. • The primary and only limitation in this process	• With the OPERA Cloud versions 22.5 and higher, every identical request (employing the same query parameters) necessitates a mandatory 30-minute interval between submissions. Note that this limit applies regardless of whether the POST, HEAD, and GET cycle has been completed. • With the OPERA Cloud versions 23.1 and higher, the getDailyRatePlanSchedules API response expiration time has been reduced from 24 to 6 hours.	• Operate on a first-come, first-served basis. Data submissions are processed in the order they are received, and the only constraint lies in ensuring that the payload adheres to the specified size guidelines.

Operation ID	Functionality	Limit	OPERA Cloud Version Dependencies (if any)	Recommendations
		is related to the payload size that the API user intends to upload. • Optimal payload size: 2 MB		
startSetBestAvailableRatePlansProcess (POST) getBestAvailableRatePlansProcessStatus (HEAD) getBestAvailableRatePlans (GET)	• Enables you to create Best Available Rate Plans by Length of Stay or by Day. • You can use this API to post new or update existing Best Available Rate by Length Of Stay or by DAY to OPERA Cloud.	• Every identical request (employing the same query parameters) necessitates a mandatory 30-minute interval between submissions. Note that this limit applies regardless of whether the POST, HEAD, and GET cycle has been completed. • The primary and only limitation in this process is related to the payload size that the API user intends to upload. • Optimal payload size: 100 KB	• With the OPERA Cloud versions 22.5 and higher, every identical request (employing the same query parameters) necessitates a mandatory 30-minute interval between submissions. Please note that this limit applies regardless of whether the POST, HEAD, and GET cycle has been completed. • With the OPERA Cloud versions 23.1 and higher, the getBestAvailableRatePlans API response expiration time has been reduced from 24 to 6 hours.	Operate on a first-come, first-served basis. Data submissions are processed in the order they are received, and the only constraint lies in ensuring that the payload adheres to the specified size guidelines.

Operation ID	Functionality	Limit	OPERA Cloud Version Dependencies (if any)	Recommendations
startHurdleRatesProcess (POST) getHurdleRatesProcessStatus (HEAD) getHurdleRates (GET)	• Enables you to create Hurdle Rates • You can use this API to create sell limits in OPERA Cloud by date. Refer to Sell Limits in the OPERA Cloud Services User Guide for more information.	• Every identical request (employing the same query parameters) necessitates a mandatory 30-minute interval between submissions. Please note that this limit applies regardless of whether the POST, HEAD, and GET cycle has been completed. • The primary and only limitation in this process is related to the payload size that the API user intends to upload. • Optimal payload size: 100 KB	• With the OPERA Cloud versions 22.5 and higher, every identical request (employing the same query parameters) necessitates a mandatory 30-minute interval between submissions. Note that this limit applies regardless of whether the POST, HEAD, and GET cycle has been completed. • With the OPERA Cloud versions 23.1 and higher, the getHurdleRates API response expiration time has been reduced from 24 to 6 hours.	Operate on a first-come, first-served basis. Data submissions are processed in the order they are received, and the only constraint lies in ensuring that the payload adheres to the specified size guidelines.

Module: Availability

Operation ID	Functionality	Limit	OPERA Cloud Version Dependencies (if any)	Recommendations
postRestrictionsProcess (POST) getRestrictionsProcessStatus (HEAD) getRestrictions (GET)	• Enables you to create Restrictions. • A user can send various restrictions to OPERA Cloud by specifying restriction details in the request. You can set restrictions for a whole year and have multiple restrictions on a given day. However, there can be a hierarchy of restrictions. • Refer to Managing Restrictions in the OPERA Cloud Services User Guide for more information.	• Every identical request (employing the same query parameters) necessitates a mandatory 30-minute interval between submissions. Note that this limit applies regardless of whether the POST, HEAD, and GET cycle has been completed. • The primary and only limitation in this process is related to the payload size that the API user intends to upload. • Optimal payload size: 2 MB	• With the OPERA Cloud versions 22.5 and higher, every identical request (employing the same query parameters) necessitates a mandatory 30-minute interval between submissions. Note that this limit applies regardless of whether the POST, HEAD, and GET cycle has been completed. • With the OPERA Cloud versions 23.1 and higher, the getRestrictions API response expiration time has been reduced from 24 to 6 hours.	• Operate on a first-come, first-served basis. Data submissions are processed in the order they are received, and the only constraint lies in ensuring that the payload adheres to the specified size guidelines.

Module: Inventory

Operation ID	Functionality	Limit	OPERA Cloud Version Dependencies (if any)	Recommendations
postSellLimitsProcess (POST) getSellLimitsProcessStatus (HEAD) getSellLimits (GET)	• Enables you to create Sell Limits • You can use this API to create sell limits in OPERA Cloud by date. Refer to Sell Limits in the OPERA Cloud Services User Guide for more information.	• Every identical request (employing the same query parameters) necessitates a mandatory 30-minute interval between submissions. Note that this limit applies regardless of whether the POST, HEAD, and GET cycle has been completed. • The primary and only limitation in this process is related to the payload size that the API user intends to upload. • Optimal payload size: 2 MB	• With the OPERA Cloud versions 22.5 and higher, every identical request (employing the same query parameters) necessitates a mandatory 30-minute interval between submissions. Note that this limit applies regardless of whether the POST, HEAD, and GET cycle has been completed. • With the OPERA Cloud versions 23.1 and higher, the getSellLimits API response expiration time has been reduced from 24 to 6 hours.	• Operate on a first-come, first-served basis. Data submissions are processed in the order they are received, and the only constraint lies in ensuring that the payload adheres to the specified size guidelines.

Anti-Patterns

Issue	Explanation	Anti-Pattern Consequence	Recommendations
Attempting to initiate a new request for the **startReservationsDailySummaryProcess** operation (request that uses the parameter "startLastModifiedDate/ endLastModifiedDate") while an existing request is still in progress, leading to the error "Previous request exists within the allowed 3-hour time frame."	This error indicates that there is an ongoing request for the **startReservationsDailySummaryProcess** operation that has not completed the entire three-step data process (POST, HEAD, and GET). The API enforces a restriction that only one request for this operation is allowed within a 3-hour time frame.	Initiating a new request without allowing the previous one to complete could lead to increased server load and potential disruptions in the expected data flow.	It is crucial to finalize the preceding request by executing all three steps (POST, HEAD, and GET) before attempting to initiate a new data request for the **startReservationsDailySummaryProcess** operation. This recommendation should be followed not only for **startReservationsDailySummaryProcess** but also for other POST asynchronous operations that fetch data from OPERA Cloud. Ignoring ongoing requests for any asynchronous operation may result in undesired outcomes. Always ensure that each asynchronous operation follows a sequential and orderly execution to maintain data integrity and API functionality.

Issue	Explanation	Anti-Pattern Consequence	Recommendations
Attempting to update data to OPERA Cloud immediately after its creation using asynchronous APIs without considering the payload size, assuming no restrictions on the update process.	While the operations that POST data to OPERA Cloud, including rate updates, operate on a first-come, first-served basis, allowing updates right after creation, the primary limitation lies in the payload size. Ignoring the payload size recommendation of 200 entries within the item array can lead to suboptimal performance, increased server load, and potential disruptions in the processing of API requests.	Frequent and unrestricted updates without adhering to payload size guidelines may result in inefficient API processing, increased latency, and potential errors. This can impact the overall reliability and performance of the asynchronous update operations.	It is crucial to consider and adhere to the recommended and optimal payload size of 200 entries within the item array when updating rates or performing other POST data to OPERA Cloud operations. Ignoring this guideline may lead to an anti-pattern where updates are made without proper consideration, potentially causing performance issues and compromising the reliability of the API. Always ensure that API usage aligns with recommended best practices for optimal performance and data integrity.

FAQs

How frequently should I send HEAD?

We suggest sending the HEAD request not more than once a minute, though the frequency will depend upon your use case. Refer to the Best Practices section for additional details.

Can I use this external system for polling Business Events?

Yes, if you already have an external system configured for the Asynchronous APIs, the OPERA environment owner would just need to follow steps 3 and 4 in Configuring the Polling API.

I posted a GET request and it returns an error "Not Found (HTTP Status 404)". Does it mean I can only fetch data once?

Yes, once you have sent the final GET call to obtain the results of the async processing, the data are no longer available on the same request ID. To retrieve the data again, you must restart the request sequence at step one and go through POST, HEAD, and GET again. Refer to the Sample Workflow section for additional details.

The HEAD request gives me an "HTTP Status 200 OK". What should I do?

If the HEAD request gives an **HTTP Status 200 OK** response, note that the HEAD response header parameter ‘retry-after’ will indicate the waiting time (in seconds) before resending the HEAD request. However, this is only available in OPERA Cloud versions 23.2 and higher. In the absence of a ‘retry-after,’ if the HEAD request gives an

HTTP Status 200 OK response, allow another 1-2 minutes before resending the HEAD request. In other words, if the HEAD response has not returned a 201 Created response with a header location, it is likely the job has not finished yet. After receiving an HTTP status 201 Created, you can proceed with the GET request.

When will my API responses (POST, HEAD, GET) be available?

There is no set length of time. Different requests require varying amounts of processing, and operational use of OPERA Cloud will affect the speed of responding to the request. Refer to the Types and Usage Recommendations for additional details on the API limits and recommendations.

How can I fetch Past Revenue?

By utilizing the asynchronous reservation, inventory, and block APIs in the OPERA Cloud Property, you can access historical revenue data.

When will my API response be available?

There is no set length of time. Different requests require varying amounts of processing, and operational use of OPERA Cloud will affect the speed of responding to the request.

My GET request returns an error "Too Many Requests (HTTP Status 429)." What does this mean?

Refer to the Types and Usage Recommendations for details on the API limits and then try again after waiting the recommended length of time.

Why do the APIs have an external system in the path?

OPERA Cloud identifies who is requesting the data using this external system variable and restricts the possibility of duplicate requests in a short time period. For details on configuring the external system, refer to the Polling API (pull).

I already have an external system configured. Can I use this system for polling Business Events?

Yes, if you already have an external system configured for the Asynchronous APIs, the OPERA environment owner would just need to follow steps 3 and 4 in Configuring the Polling API.

What is the difference between synchronous getReservations vs asynchronous ReservationDailySummary? When do I use each of these APIs?

Synchronize API getReservations is designed to support the hotel's front desk, housekeeping, and guest reservation journey use cases with a vast array of filters. It does not return revenue forecast data and fetch a maximum of 1000 records from OPERA Cloud, even while using pagination. This synchronous operation returns a lot more data for each individual reservation.

The asynchronous operation returns a summary of each individual reservation. However, it provides the ability to fetch bulk amounts of reservations in just 1 request.

Can I send delta changes with rate pricing update and avoid having to send full overlay in the request body?

Certainly. The Asynchronous SetDailyRatePlanSchedules enables you to modify rates exclusively for specific dates and room types. This behavior is consistent and applicable to the restriction and hurdle asynchronous APIs as well.

What is the difference between Standard Rate and Daily Rate?

Daily rates allow you to change the price schedule for each room type on a daily basis. This allows you to work with rates on a more detailed level by raising or lowering rate schedules for future dates as needed. You can manage rates according to your daily needs, regardless of whether or not a yield management system is integrated with OPERA Cloud. You can designate up to 20 rate codes as daily rates. Daily rates can coexist with other traditional date-range based rate codes. Refer to Configuring Rate Code Type in the OPERA Cloud Services User Guide for more information on rate code types.

What are the minimum requirements to update a rate? Do I need to update rate 2 to rate 5 with SetDailyRatePlanSchedulesProcess?

There is no requirement to change rate 2 to rate 5. You can view a Sample for updating the dailyRatePlanSchedule operation.

Can OPERA Cloud notify a revenue system when the business date has rolled?

Even though you can fetch the business Date of the hotel using getBusinessDate, it would require periodic calls to know if the hotel has rolled the Business Date. Our recommendation is to utilize Business Event functionality using the ‘NIGHT AUDIT’ module and the business event called ‘ROLL BUSINESS DATE.’ This triggers an event as soon as OPERA Cloud has rolled over their business date.

Do the APIs support Foreign Currencies in Rate Plan Codes?

Yes, OPERA Cloud property APIs fetch and update any currency code that is configured and available in the OPERA property.

How long is the data available after getting a response?

The data is accessible for a period of 6 hours. After this timeframe, the data is automatically purged, and you will need to submit a new request to access it again.

Oracle Payment Interface APIs

Overview

Reservations created via third-party channels often include a credit card number to secure the booking. The OPI Token Exchange API openPaymentBulkTokenExchange allows partners to exchange credit card numbers for tokens, using the Payment Service Provider that OPERA Cloud is integrated to for payment processing. The tokens are then stored in OPERA Cloud against the reservation and can be used for subsequent payments as needed.

Prerequisites

To call the OPI Token Exchange API, the hotel must purchase and enable the Oracle Payment Interface Cloud Service.

Calling the Oracle Payment Interface APIs

Oracle Payment Interface APIs are called in the same way as Oracle Hospitality Property APIs.

If the hotel does not have the Oracle Payment Interface Cloud Service enabled, the following error is returned:

HTTP status: 404

Response body: OPICS-NOT_FOUND

Refer to the explanation of this error for resolution steps.

HTTPS status: 504 from the OPI Token Exchange API means the Payment Service Provider did not respond in a timely manner.

For more details on OPI Cloud, refer to the Oracle Payment Interface Cloud 23.1 - Get Started documentation.

Oracle Hospitality Distribution APIs

Obtaining Details from Oracle Hospitality Distribution

As a partner, you must request access to the Distribution APIs by contacting the Distribution team (hgbu_distribution_partner_rqs_grp@oracle.com) and providing the following details:

- **Oracle Cloud Account Name**
- **Oracle Cloud Account ID**

You will then receive further instructions within 5 business days.

Once access is granted, Oracle will send you the following:

- **Channel Code** — This is your unique channel code that is global and used across all hotels.
- **Integration User**
- **Gateway URL** — This is used to access the Distribution APIs in each environment requested.

Authenticating to Oracle Hospitality Distribution APIs

Prerequisites

- A Channel Code in the Distribution platform provided by Oracle.
- A valid integration user provided by Oracle upon creation of the channel code.

Getting a Token for Distribution APIs

Use the Distribution Authorization API to obtain the access token (in the JWT format) that matches the provided credentials issued by the Oracle Hospitality Distribution administrator for a specific Distribution channel partner (see Prerequisites section).

Use this token in every call to the Distribution APIs. Once a token is generated, it can be used in subsequent calls until it expires. The expiry date is noted in the API response.

> **Note:**
>
> The Distribution APIs are separate from the Oracle Hospitality Property APIs, and tokens generated by the Distribution Authorization API can only be used for Distribution APIs.

Calling Oracle Hospitality Distribution APIs

The Distribution Reservation Notifications API enables authorized channel partners to create and update reservations for any active property in Oracle Hospitality Distribution, whether that property is using OPERA Cloud, OPERA 5, or on-premise PMS versions.

This API is meant to transmit reservations already created and committed in the channel external system and does not perform any Pricing and Availability validation.

Headers

The Oracle Hospitality Distribution APIs require the following mandatory headers:

- **Token**: A header “Authorization” that includes the OAuth token obtained from the Oracle Distribution Authentication API.
- **x-channelCode**: A channel code provided by Oracle.
- **Application key**: An unique application key is created for each application upon registering an application in the Oracle Hospitality Developer Portal. This key must be passed in the “x-app-key” header of each request to the Oracle Hospitality APIs. To obtain your application key, see Viewing the Application Key.

For a list of Oracle Hospitality Distribution APIs, see Oracle Distribution APIs.

Oracle Hospitality Nor1 Integrated Upsell APIs

Obtaining Details from Oracle Hospitality Nor1

You must request access to the Nor1 Upgrades APIs by emailing the Nor1 team (hgbu_nor1_partner_rqs_grp@oracle.com) and providing the following details:

- Partner Organization Name
- Oracle Cloud Account Name
- Oracle Cloud Account ID

Provide the following details if you are an OPERA Cloud Foundation customer:

- Tenant / Chain Code
- SSD URL
- Indicate if the chain is a Production or UAT chain

After sending the email, you will receive further instructions within 5 business days. Once access is granted, Oracle will send you the following:

- providerId — This is your Nor1 unique provider code that is global and used across all hotels.

Authenticating to Oracle Hospitality Nor1 Upgrades APIs

The Nor1 Upgrades APIs are secured the same way as the Oracle Hospitality Property APIs. For further details, refer to the following topics:

Changing Your Integration User Password

For environments secured by Resource Owner authentication, integration user passwords expire after 1 year and must be changed every year. To change your password:

1. Find the email you received when the hotel approved your integration user. This includes a URL for the Shared Security Domain identity server. Go to this URL.
2. Log in using your integration username and password.
3. Go to the “My Information” panel. This will show you basic information about your integration user.
4. Expand the “Change Password” section.
5. Reenter your current integration user password.
6. Enter your new integration user password twice. Please note the password policy, which can be viewed by clicking the “i” icon next to the New Password field.

API Troubleshooting

If you are experiencing issues when consuming APIs, check the following:

API

- Verify the API you are calling is visible in the APIs tab of the developer portal.
- Verify the API version in the URL matches the version v0 or v1 listed in the developer portal.
- If connecting to OPERA Cloud, verify the functionality being used is active and available for the relevant OPERA Cloud PMS version by reviewing the latest OPERA Cloud Services Release Readiness Guide and User Guide available on the Oracle Hospitality Hotels documentation page.
- Verify the input variables are relevant to the OPERA Cloud solution being called and are not a copy of Postman samples. Note that each OPERA Cloud environment is uniquely configured. You can determine the configuration specific to the hotel you are calling by reviewing the List of Values and Enterprise Configuration APIs.

Environment and Credentials

- Verify the environment (chain) card is on the Environments tab in the developer portal.
  - If you are an integration partner and the environment is not listed, then follow the steps in the Obtaining Details from the Hotel topic to gain access.
  - If you are an integration partner and this is a production environment, verify you have followed the steps in the Moving to Production topic.
- Verify the application's second tab shows the plans expected in the developer portal.
- Verify the application key being used matches the application checked in step 2 in the developer portal.
- Verify the clientId and clientSecret correspond to those on the Environment card under the Environments tab in the developer portal.
- If the environment is secured by Resource Owner authentication, verify the following for the integration user:
  1. The user is in the organization I<SSD org code>
     a. The user was created from the SSD URL ending "?apiuser=y"
  2. The username does not have spaces in it.
     a. If spaces in the current username exist, create a new integration user with a shorter username with no spaces.
  3. The user has the <SSD org code>-WSACCESS role.
     a. Log in to SSD using the link in the "Thank you" email and then go to "My Access" to see if the user has the WSACCESS role.
     b. If not, contact the environment owner and ask for approval for this role.
  4. Verify if the oAuth token is still valid (note the token lasts for 60 minutes). Obtain a new oAuth token to ensure it is valid.

Mandatory Headers

Verify the required headers:

- Calling OPERA Cloud Property APIs — Ensure the x-hotelid header matches a hotel in the chain being called.
  - OPERA Cloud Property APIs for OPERA Cloud 22.1+ — When verifying data for the hub level, ensure you send x-hubid and not x-hotelid.
- Calling Oracle Hospitality Distribution APIs — Verify the x-channelCode header matches the header provided via email by the Oracle Hospitality Distribution team.
- Calling NOR1 Upgrade API — Ensure the providerId header matches the header provided via email by the NOR1 team.
- If sending a POST request, ensure you are sending the "accept" header as "application/json."

Errors

Verify the list of errors and follow the suggested resolution paths.

Logging an Issue

Before reporting an issue, first exhaust the self-service troubleshooting.

If you still have an issue with an API, ensure you include the following information when logging the issue:

- Full CURL request (with credentials redacted)
- Response code
- OPERA Cloud environment version number (via this API)
- OPERA Cloud environment name or gateway URL
- Context of what is being attempted (for example, type of integration, task being carried out, and so on).

Common HTTP Errors and Messages

Common error codes produced by Oracle Hospitality APIs are listed in the following table.

For a complete list of OPERA Cloud REST API error codes, refer to the Web Service Error Codes topic.

Table 8-3 Common HTTP Error Messages

Error Status	Error Response Body	How to Resolve
400	Response body details which fields are at fault.	Change the fields mentioned in the error response, such that they align with the specifications; referencing the swagger spec will help here. In some cases the values are determined by a (hotel specific) configured List of Values (LOV), so ensure you supply a value that is in the LOV for that hotel; the List Of Values Oracle Hospitality APIs will help here.
400	This API is not supported for the current database version.	Contact Oracle Customer Support at the Customer Support Portal stating the gateway being called and the error message received.
401	No response body	Ensure your oAuth token is valid and latest. Also, ensure your Application Key is valid. Check it by Viewing the Application Key
402	"invalid_grant", "[Wrong Password]"	Check the password of the integration user. See Changing Your Integration User Password for more information.

Table 8-3 (Cont.) Common HTTP Error Messages

Error Status	Error Response Body	How to Resolve
403	Unauthorized to access the resource	This could be caused by the integration user missing the WSACCESS role. Ensure the environment owner has approved the integration user. This could also occur if you are sending the wrong hotelId. Ensure the hotel ID being sent in the x-hotelid header matches a hotel in the environment being called. In recently migrated environments secured by Client Credentials authentication (OPERA Cloud Identity Management), ensure you are using the reissued clientId and clientSecret, not the pre-migration clientId and clientSecret.
403	No response body	Ensure your oAuth token is valid and up to date. Re-request it by using Authenticating to Oracle Hospitality Property APIs. Also ensure your integration user in OPERA Cloud Services has access to the property (hotel) you supplied in the "x-hotelid" header.

Table 8-3 (Cont.) Common HTTP Error Messages

Error Status	Error Response Body	How to Resolve
403	No Subscribed Plan or API found	This occurs when accessing an API to which you do not have access. If this is an early adopter (v0) API and you think you already have access to the Early Adopter API Program, check that the application specified in the x-app-key header has access to the Early Adopter APIs by following this process: 1. In the Developer Portal, click the **Applications** tab. 2. Choose the application whose key matches the x-app-key being sent (double check against the end of the application key that appears in the list of applications). 3. Click **View details** 4. Click the **Subscriptions** tab and verify it shows all of the following: • Early Adopter • Hospitality APIs • OAuth If you have not yet called the v0 APIs but would like to, contact us as explained in the Early Adopter API Program. If this is a Distribution API, you must register in the Developer Portal to use the API.

Table 8-3 (Cont.) Common HTTP Error Messages

Error Status	Error Response Body	How to Resolve
403	User is not authorized to access data for resort	Check that your integration user has the WSACCESS role: • Find the email you received when the hotel approved your integration user. This includes a URL for the Shared Security Domain identity server. Go to this URL. • Log in using your integration username and password. • If the login fails, click the "Can't Sign In?" link to reset your password. • Verify the hotelId supplied in the "x-hotelid" header matches the one provided by the hotel. See Obtaining Details from the Hotel for more information.
404	No response body	If you are calling an Asynchronous API: Once you have sent the final GET call to obtain the results of the async processing, the data is no longer available on the same summaryId, so you will receive a 404 error. To retrieve the data again, you must restart the request sequence at step one.
404	OPICS-NOT_FOUND	Contact the environment owner to verify that Oracle Payment Interface Cloud Service has been set up by checking for product ID 14308 and ensuring that Token Exchange Service is selected. Customers can contact Oracle Consulting or a reseller to configure Oracle Payment Interface Cloud as needed.
405	No response body	Ensure the HTTP verb you are using is supported by the Oracle Hospitality APIs by checking the Oracle Hospitality APIs documentation.

Table 8-3 (Cont.) Common HTTP Error Messages

Error Status	Error Response Body	How to Resolve
406	No response body	Set your "accept" header to "application/json" as the Oracle Hospitality APIs will produce only "application/json".
413	No response body	Ensure your request matches the documented request body schema. If your scenario requires "bulk" fetch or update then consider the jobbed Oracle Hospitality APIs.
414	No response body	Consider whether you need to specify all the query parameters being specified; there may be more efficient resources or ways to structure your query.
415	Unsupported Media Type	Ensure your request payload has a content-type of "application/json".
500	Response body details the error.	Try your request again in a few moments or contact Oracle Customer Support at the Customer Support Portal.
502	No response body	Try your request again in a few moments or contact Oracle Customer Support at the Customer Support Portal.
503	No response body	Try your request again in a few moments or contact Oracle Customer Support at the Customer Support Portal.

HTTP Methods Supported

The Oracle Hospitality APIs use the following HTTP verbs:

- **GET** to retrieve resources.
- **HEAD** to query the status of jobbed requests.
- **POST** to create resources.
- **PUT** to replace resources.
- **DELETE** to delete resources.

HTTP Response Headers

POST always returns the location of the newly created resource in a Location header.

Github and Postman Collections

Oracle Hospitality has a Github repository containing both Oracle Hospitality REST API specifications and accompanying Postman Collections.

You can access Github and locate the REST API specifications and Postman Collections at the following URL: https://github.com/oracle/hospitality-api-docs.

REST API Specifications

In the rest-api-specs folder in Github, you can view the published V1 APIs for OPERA Cloud and/or download the json specifications for the Oracle Hospitality APIs.

Postman Collections

In the postman-collections folder in Github, you can download and use the Postman Collection to help you get you started with our APIs and become more familiar with using them. The postman-collections folder contains the following content:

- Postman collections with many different API call samples on how to perform different functional workflows (for example, digital check-in and checkout).
- Postman Environment defining the main environment variables required to use the postman collections against our Hospitality APIs. Ensure you update this file with the relevant credentials and data for the environment to which you are connecting.
- Document describing the different workflows supported in the postman collections.

The Postman collection is also available at the following URL: https://www.postman.com/hospitalityapis/workspace/oracle-hospitality-apis/overview.

Setting Up Your Postman Collection

To start calling the APIs, proceed to set up a postman environment collection using the below information. You might want to configure one environment collection for UAT and another one for Production as the variables will differ.

Table 8-4 Information for Postman Environment Collection

Hostname	This is the API gateway URL that can be viewed by logging into the Developer portal and viewing the environments tab.
Username & Password	This is the Integration username and password. This is relevant only for environments secured by Resource Owner authentication. See Authenticating to Oracle Hospitality Property APIs for the steps to obtain these.
CLIENT_ID & CLIENT_SECRET	The Client ID obtained from the OHIP Developer portal. See Viewing the Client Secret for details.
AppKey	The application key that was previously obtained. See Viewing Application Details for the specific steps.

Table 8-4 (Cont.) Information for Postman Environment Collection

HotelId	The Hotel ID against which you want to perform actions (for example, obtaining reservation data).

Once you have added these values into your postman environment, you can begin to call the oAuth Token. If everything is set up correctly, OHIP will respond with a HTTP 200 OK response, including the oAuth token. Now you can proceed to try out the collections as required.

Date Formats

Dates in the Oracle Hospitality APIs are expressed in the RFC3339 “full-date” format (that is, date-fullyear "-" date-month "-" date-mday. "T" time-hour ":" time-minute ":" time-second "Z" / ("+" / "-") time-hour ":" time-minute). The following are examples of this format:

- 2023-06-05 for 5th of June 2023
- 2023-06-05T08:43Z or 2023-06-05T09:43+01:00 for 5th of June 2023 at 8:43 AM in the UTC time zone or 9:43 in the local time zone

Most times are expressed in local time zones, not in Coordinated Universal Time (UTC).

Asynchronous APIs return times in Coordinated Universal Time (UTC).

For Profiles APIs, the time depends on where the profile was created. For example:

- If the profile was created at the hub level, then Profiles APIs return times in the hub time zone if the time zone is set. If no hub time zone is set, then for a profile created at the hub level, Profiles APIs return in Coordinated Universal Time (UTC).
- If the profile was created at the property level, then Profiles APIs return times in the property time zone if the time zone is set. If no property time zone is set, then for a profile created at the property level, Profiles APIs return in Coordinated Universal Time (UTC).

Table 8-5 Asynchronous API Times Zones

API	Where was the Profile Created?	Is the Time Zone Set?	Time Zone Returned
Asynchronous APIs	n/a	n/a	Coordinated Universal Time (UTC)
Profiles APIs	Hub	No	Coordinated Universal Time (UTC)
Profiles APIs	Hub	Yes	Hub timezone
Profiles APIs	Property	No	Coordinated Universal Time (UTC)
Profiles APIs	Property	Yes	Property timezone
All other property APIs	n/a	n/a	Property timezone

Special Characters in URLs

Query parameters for some Hospitality API operations require text to be entered, for example, a rate code name or a person's name. If the text includes special or multibyte characters, these characters must be URL encoded. For example, an asterisk URL encodes as %2A.

Forward slash characters (that is, /) or UTF-8 characters must be double URL-encoded so that, for example:

/ encodes as **%252F**

し encodes as **%25E3%2581%2597**

For encoding standards, refer to the RFC Series on the RFC Editor website. The standard that defines when to encode is RFC3986 (section 2.4, When to Encode or Decode).

9
Versioning

All v1 APIs are backwards compatible, and we maintain best efforts to maintain backwards compatibility of v0 APIs. In the event of a breaking change, this is noted on the following Customer Support Portal article: Oracle Hospitality Integration Platform Upcoming Major Changes. To be notified when the Upcoming Major Changes log is updated, press the **Subscribe** button on this page.

With REST APIs the expectation is that the server and client are loosely coupled. As mentioned in Anti-Patterns, it is recommended that you follow the "Tolerant Reader" pattern, using only the operations and resource definitions needed by your integration and ignore operations and resources that are not relevant to your integration.

When following this pattern, there is no need to regenerate the code when specifications change because backwards compatibility assures you that existing code will work. Further, any newly added optional fields or operations are irrelevant to the way the code works. This also keeps the code lean.

Question: How do I know when new APIs are released?

The Oracle Hospitality Integration Platform Patch Release Notes list all new and deprecated APIs. To be notified when a new patch release note is issued, click the **Subscribe** button on this page.

Question: What version of the OPERA Property APIs are available in OHIP?

The version number is the one in the API specifications themselves as displayed in the API search engine in the APIs tab of the developer portal and in the specifications themselves.

Question: Is the OPERA Property APIs version the same as the OHIP developer portal version?

No. The OHIP developer portal version, which is visible by clicking the My Profile dropdown at the top right of the Developer Portal, relates only to the OHIP version, not the version of any APIs available via OHIP.

Question: The environment I'm calling is on a different version of OPERA Cloud than the version in the specifications, is that a problem?

No. As mentioned above, v1 APIs are backwards compatible. Therefore, an operation available in OPERA Cloud 23.2 will work exactly the same way whether the version of specifications in OHIP is 23.2 or higher.

If the release notes include a new API operation added in a later release (for example, 24.1), then this will be available only in OPERA Cloud 24.1+ environments.

10

Analytics

The Analytics tab enables you to view the details of API usage, such as the total number of API calls, the number of API calls that were successful, and the number of API calls that failed or produced errors. With analytics, you can troubleshoot your API calls and quickly identify where the error is occurring in your application.

When paying for API usage, it is critical to know when one of your applications starts making unusual calls, so you can avoid unexpected bills and better plan your expense budget.

Analytics data enables you to ensure your applications are running correctly.

> **Note:**
>
> Usage does not exactly equate to what you are billed as there are a number of free calls. For a list of the API calls that are not billed, see FAQs.

Analytics data applies to both inbound and outbound API calls.

Search Parameters

You can filter analytics using the following parameters:

- **Hotel ID**
- **Environment Type** — **Non Production**, **Production**, or **All**.
- **Application**
- **Organization**
- **API**
- **Date** — You can select the **Last 24 Hours**, **Last 7 Days**, **Last 31 Days**, **Last 12 Months**, **Current Billing Cycle**, or **Last Billing Cycle** for partners. You can also select **Custom** to enter a specific date range. The last 24 hours is the default date in the search.

API Response Status Code

The response status code is the easiest way to understand what happened with an API request without reading and decoding the response body.

Using Analytics

You can view the details of usage analytics at a global level or at an application level. Each view includes a breakdown of error codes.

Viewing Analytics from the Main Analytics Tab

This provides a view of usage analytics at a global level.

1. Open the Developer Portal and click the **Analytics** tab.
2. Optionally choose search parameters to narrow results to specific hotel(s), environment type(s), application(s), organization(s), API(s), and date range:
    - **x-hotelids**: Enter the OPERA property ID of one or more hotels. This was sent in the x-hotelid header of the API request.
    - **Call Type**: Select either **REST** or **Streaming** for the call type. Alternatively, you can select **All**, which provides analytics for both REST and Streaming combined.
    - **Environment Type**: Select **All**, **Non Production**, or **Production** for your environment type. This selection is determined based on the environment being called, either a non production or production instance.
    - **APIs**: Select one or more **API(s)** from the list.
    - **Application**: Select one or more application(s) for which you want to view analytics.
    - **Organization**: Hoteliers can select whether to view their own analytics or those of one or more partner organization(s).
    - **API**: Select one or more **API(s)** from the list.
    - **Choose Range**: Select a date range from which to view usage analytics. You can select **Last 24 Hours**, **Last 7 Days**, **Last 31 Days**, **Last 12 Months**, **Current Billing Cycle**, or **Last Billing Cycle** for partners. You can also select **Custom** to enter a specific date range.

The Usage graph appears below and shows the total request number, the number of successful API calls, and the number of errors based on the search parameters you defined.

If there are errors, the error codes and the total number for each error code appear in a table below the graph.

Viewing Analytics from the Applications Analytics Tab

This provides a view of usage analytics at an application level.

1. Open the Developer Portal and click the **Applications** tab.
2. Search for an application and click the **View details** link for it.
3. Click the **Analytics** tab.
4. Optionally choose search parameters to narrow results to specific hotel(s), environment type(s), organization(s), API(s), and date range:
    - **x-hotelids**: Enter the OPERA property ID of one or more hotels. This was sent in the x-hotelid header of the API request.
    - **Call Type**: Select either **REST** or **Streaming** for the call type. Alternatively, you can select **All**, which provides analytics for both REST and Streaming combined.
    - **Environment Type**: Select **All**, **Non Production**, or **Production** for your environment type. This selection is determined based on the environment being called, either a non production or production instance.
    - **APIs**: Select one or more **API(s)** from the list.

- **Organization**: Hoteliers can select whether to view their own analytics or those of one or more partner organization(s).
- **Choose Range**: Select a date range from which to view usage analytics. You can select **Last 24 Hours**, **Last 7 Days**, **Last 31 Days**, **Last 12 Months**, **Current Billing Cycle**, or **Last Billing Cycle** for partners. You can also select **Custom** to enter a specific date range.

The Usage graph appears below and shows the total request number, the number of successful API calls, and the number of errors based on the search parameters you defined.

If there are errors, the error codes and the total number for each error code appear in a table below the graph.

Downloading Overall Analytics

1. Open the Developer Portal and click the **Analytics** tab.
2. Select the search parameters as required.
3. Click the **Download Report** button and click **Download** again to confirm.

Downloading Application Analytics

1. Open the Developer Portal and click the **Applications** tab.
2. Search for an application and click the **View details** link for it.
3. Click the **Analytics** tab.
4. Select the search parameters as required.
5. Click the **Download Report** button and click **Download** again to confirm.

This exports the report as a .csv file that you can open or save. The usage report also contains the breakdown by error codes for the given time period.

> **Note:**
>
> - The downloaded file reflects dates and times in UTC time zone.
> - Any single report can only cover a maximum period of 31 days per downloaded file.
> - If the x-hotelids search parameter is used, the report provides a total usage in the summary section and a detailed usage per x-hotelid in the detail section.
> - If no x-hotelids search parameter is used, the report provides a total usage in the summary section and an aggregated usage for all x-hotelids in the detail section.
> - To get a detailed usage for all x-hotelids, select all x-hotelids in the x-hotelids search parameter.
> - Be aware that some APIs, (for example, Nor1, Distribution, and Property OAuth token APIs), do not have x-hotelid information. Those API calls are not captured when using the x-hotelids search parameter, but only when searching without this parameter.

11

Business Use Cases

Business Use Cases is a searchable page for finding business use case articles. On this page, you can search for a business use case and view detailed information about it to assist you when building your application. The business use case articles include helpful information, such as an overview, key terminology, sample workflows, and FAQs. The following business use case articles are available and more will be added in future releases:

- Blocks
- Business Events and Streaming
- Nor1 Upsell
- Payments
- Property Interface - Posting Charges
- RMS - Revenue Management Systems

12

Business Events

A business event is an event (that is, an update) that happens to a resource, for example, a reservation. Rather than GETting the reservation repeatedly, subscribing to the Update Reservation event will inform you when this or other reservations are changed. Oracle Hospitality systems emit Business Events in multiple scenarios, such as creating a new resource, updating a resource, or deleting a resource. Each business event has the following pieces of information:

- Resource — This is the resource that has received the event (for example, a reservation).
- Event — The name of the event, for example, an update to a reservation.
- Old value — This is the old value for the field.
- New value — This is the new value for the field.

There are two methods for consuming events from Oracle Hospitality APIs: Polling and Streaming.

Both streaming and polling can be used to consume events from the same environment at the same time. For example, use streaming for events, such as checkin, where real time is important and use polling for less time critical events.

For a list of business events and data elements, refer to the Configuring Business Events topic in the OPERA Cloud User Guide.

Streaming API (push)

Overview of Event Consumption via the Streaming API

When partners or customers consume business event data from OPERA Cloud, they can choose which events an application will receive. These are the events to which the application subscribes.

A partner or customer can subscribe to business events from a specific environment, but approval from the environment owner is required.

For more information on streaming, read our blog post.

The following resources can help you get started with the streaming API:

- GraphiQL — This provides a visual playground for familiarizing yourself with the streaming API. The Connecting to the Streaming API via GraphiQL topic provides instruction on how to use this tool.
- Postman collection — The Github and Postman Collections topic provides instruction on how to set up and use this tool.
- Node reference implementation in TypeScript.

Customer Approval for Partners

A customer Developer Portal user can only approve the sending of business events for hotels to which they have access.

Customers can log in to the Developer Portal and do the following:

- See a list of the partners who are requesting to consume events.
- View a list of events a partner is consuming and see from which hotels and environments the partner is consuming these events.
- Approve event consumption for one or many hotels in the requested environment.

> **Note:**
>
> In the following circumstances, the event subscription will automatically be approved.
>
> - Partners consuming business events from the partner sandbox.
> - Partners consuming business events from their dedicated non production environment.
> - Customers consuming business events from their own non production environment.
> - Customers consuming business events from their own production environment.

Customer - Enabling Streaming

To get started with their first environment, customers must purchase the OHIP Premium Remote Assistance SKU (B93152) to engage with Oracle Professional Services.

For subsequent environments, customers must raise a technical SR. The Technical SR must include the following details, which are found on the customer's **Environments** card in the OHIP developer portal:

- **Customer Name** — The customer's name can be found on the hotelier's **Environments** card in the OHIP developer portal and appears before the brackets on the card.
- **Customer Chain** — The customer chain to be accessed. This can be found on the hotelier's **Environments** card in the OHIP developer portal and appears after the brackets on the card.
- **Environment Name** — This can be found on the hotelier's **Environments** card in the OHIP developer portal and appears in brackets on the card.
- **Shared Security Domain** — This is the URL via which the integration user for the hotelier's environment was created.

> **Note:**
>
> Streaming is enabled at the chain level, so once enabled for a chain, you do not need to enable it at the hotel level.

Partner - Enabling Streaming

Partners can initiate their testing in the partner sandbox environment. To gain access, request this in the partner Slack community. Partners should complete a successful streaming test by confirming that their code correctly receives Business Events and does not encounter errors either from OHIP or from downstream systems. This is not an Oracle validation, but a necessary assurance for both the partner and the hotelier before the partner accesses a hotelier's environment.

Following a successful streaming test, partners must open a Technical SR if they wish to get a customer environment UAT or Production environment enabled for streaming. The Technical SR must include the following details, which are found on the customer's **Environments** card in the OHIP developer portal:

- **Customer Name** — The customer's name can be found on the hotelier's **Environments** card in the OHIP developer portal and appears before the brackets on the card.
- **Customer Chain** — The customer chain to be accessed. This can be found on the hotelier's **Environments** card in the OHIP developer portal and appears after the brackets on the card.
- **Environment Name** — This can be found on the hotelier's **Environments** card in the OHIP developer portal and appears in brackets on the card.
- **Shared Security Domain** —This is the URL via which the integration user for the hotelier's environment was created.

> **Note:**
>
> Streaming is enabled at the chain level, so once enabled for a chain, you do not need to enable it at the hotel level.

Partners must also ensure approvals have been obtained for the events they have configured on the customer environment. Customers can grant access to consume events within their developer portal.

Working with Events in the Developer Portal

Prerequisites for Consuming Events

- You must first add an OPERA environment from which to consume events before adding and subscribing to events.
- The OPERA environment must either have streaming enabled (see Streaming API) or have polling configured (see Configuring the Polling Subscription).

For information on working with business events in OPERA Cloud, refer to the Configuring Business Events topic in the OPERA Cloud User Guide. This topic also provides a list of business events and data elements.

Creating a Template of Events for an Application

The event template represents the list of events a given integration will use. When subscribing to consume events from a given OPERA environment, the template is a faster way to ensure the correct list of events is subscribed.

Before subscribing to consume events from a given OPERA environment, you must first create a template for the integration.

The template can be edited at any time. For example, your integration might change, and it now needs to consumer more events or fewer events.

> **Note:**
>
> Changes made on the template do not affect existing subscriptions to OPERA environments but will take effect when subscribing to new OPERA environments.

To configure the template events, complete the following steps:

1. Open the Developer Portal and click the **Applications** tab.
2. Search for and click an application to open it.
3. Click the **Events** tab.
4. Click **Add Event to Template** and complete the following information:
   a. **Category**: Select a category for the event, such as Reservation.
   b. **Event**: Select an event for the category, such as Check In.

   > **Note:**
   >
   > To remove an event from the template, click the **Remove** link.

5. Click **Add** to add the event to the template.
6. Repeat these steps to add additional events to the template.

Once you have configured all the events your integration requires, click the **Subscribed** tab and choose any OPERA environment(s) from which to consume these events.

Subscribing to Events

A partner can subscribe to business events from a customer. Subscriptions to business events are specific to each application.

Use the Subscribe tab to subscribe to the list of events that you added on the Template tab.

1. Open the Developer Portal and click the **Applications** tab.

2. Search for and click an application to open it.
3. Click the **Events** tab.
4. Click the **Subscribed** tab.
5. Click the **Subscribe** link.
6. Select an **Environment** and click **Subscribe**.
7. As a customer subscribing to events from your environment, select which hotels will consume the events.

The subscribed to event will be in a ‘Pending Approval’ status.

> **Note:**
>
> Only OPERA environments enabled for streaming appear in the list of environments.

Unsubscribing from Events

Once events are approved, you can unsubscribe from them as needed.

> **Note:**
>
> Unsubscribing from events on the Subscribe tab does not affect the events that appear on the Template tab. Also, changes made to events on the Template tab do not affect event subscriptions.

1. Open the Developer Portal and click the **Applications** tab.
2. Search for and click an application to open it.
3. Click the **Events** tab.
4. Click the **Subscribed** tab.
5. Under Event Subscriptions, locate the event and click the green arrow to expand it and then click **Unsubscribe**.

Approving Events

A customer with Developer Portal access can approve the consumption of events from a specific environment. The customer can approve a partner's request to consume events before any events are sent to the partner. This enables customers to protect their data.

1. Open the Developer Portal and click the **Applications** tab.
2. Search for and click an application to open it.
3. Click the **Partner Applications** tab.

> **Note:**
>
> Applications with pending event subscriptions appear with a red ribbon icon on the application card.

4. Click an application with a pending event to open it.
5. Click the **Events** tab.
6. Click **Next** to proceed with approval.
7. Select the properties for the event subscription and click **Next**.
8. Click **Approve**.

The event status changes to ‘In Progress’ and then to ‘Approved’ status. The partner can see this status in the Developer Portal.

Rejecting Events

1. Open the Developer Portal and click the **Applications** tab.
2. Search for and click an application to open it.
3. Click the **Partner Applications** tab.
4. Click an application with a pending event to open it.
5. Click the **Events** tab.
6. Click **Reject** to reject the subscription.

   The event status changes to ‘In Progress’ and then to ‘Rejected’ status. The partner can see this status in the Developer Portal.

Revoking Events

As a customer, if you no longer want a partner to consume events from your chain, deselect all hotels on the partner's application:

1. Open the Developer Portal and click the **Applications** tab.
2. Search for and click an application to open it.
3. Click the **Partner Applications** tab.
4. Click an application with a pending event to open it.
5. Click the **Events** tab.
6. Click **Edit** next to the list of hotels.
7. Deselect all the hotels.
8. Click **Save**.

Adding Events to Existing Subscription

To add events to an existing approved subscription:

1. Add the events on the Template tab (see the ‘Creating a Template of Events for an Application’ procedure above).
2. Subscribe again to consume events from the same environment (see the ‘Subscribing to Events’ procedure above).

Adding Hotels to an Existing Subscription

A customer with Developer Portal access can adjust an existing approved subscription to consume events from different hotels.

1. Open the customer or partner application.
2. Go to the **Events** tab.
3. Click **Edit** next to the list of hotels.
4. Select the hotels that can consume the events.
5. Click **Save**.

Broken Connections

To avoid missing messages, you should keep the WebSocket connected. However, interrupted connections do occur, so OHIP retains a record of the last message sent to each connection as well as up to 7 days of messages. If the WebSocket gets disconnected, reconnect as described in the Authentication message. OHIP will then send any messages that have occurred since the disconnection occurred. For example:

> **Note:**
>
> OHIP holds the last offset for 24 hours only. If you do not reconnect within 24 hours, you must send the offset in the subscribe message that reflects the last offset you received.

Table 12-1 How the Streaming API works around broken connections

Oracle Hospitality System Emits Offset	OHIP Sends Partner Offset	Partner Receives Offset
191	191	191
192	192	192
193	193	193
194		Connection broken
195		Connection broken
196		Connection broken
	194	Connection re-established
	195	194
	196	195
		196
197	197	197

If the WebSocket remains disconnected for over 7 days, this might result in missed messages. Refer to the following table for examples.

Table 12-2 How the Streaming API stores and replays 7 days of events

Day	Oracle Hospitality System Emits Offset	OHIP Sends Partner Offset	Partner Receives Offset
1	191	191	191
1	192	192	192
1	193	193	193
2	194 to 317		Connection disconnected for day 1
3	318 to 520		Connection disconnected for day 2
4	521 to 701		Connection disconnected for day 3
5	702 to 929		Connection disconnected for day 4
6	930 to 1027		Connection disconnected for day 5
7	1028 to 1240		Connection disconnected for day 6
8	1241 to 1403		Connection disconnected for day 7
9	1404 to 1581		Connection disconnected for day 8
10	1582 to 1826		Connection disconnected for day 9
11		521 to 1826	Connection re-established 521 to 1826
11	1827	1827	1827

In this example, messages 194 to 521 are missed and it is not possible to replay them.

Replaying Messages

> **Note:**
>
> If you use the ‘offsetType’ parameter in the subscribe message, then events that occurred before that parameter was used are not included in any message replay.

The OHIP schema includes the metadata attribute "offset," which is the message number.

To replay messages already received, specify the offset value following the chainCode. For example in Postman:

```
{
   "id": "1",
   "type": "subscribe",
   "payload": {
         "variables": {},
         "extensions": {},
         "operationName": null,
               "query": "subscription { newEvent(input: { chainCode:
\"<CHAIN CODE>\" offset: \"<OFFSET>\" } ) { metadata { offset } moduleName
eventName detail { oldValue newValue elementName } } }"
   }
}
```

In GraphiQL add the offset parameter after the chainCode. For example:

```
subscription{
  newEvent(input:{chainCode: \"<CHAIN CODE>\" offset: \"<OFFSET>\"}) {
    moduleName
    eventName
    detail{
      newValue
      oldValue
      elementName
    }
    metadata{
      offset
      uniqueEventId
    }
  }
}
```

OHIP sends event 193 and all events that occurred after event 193.

As noted in Broken Connections, if offset 193 was emitted more than 7 days ago, OHIP will return all messages emitted since 7 days ago. This is because OHIP retains events for only 7 days.

Retrieving the Latest Message

To retrieve only the latest message, use the "offsetType" parameter and specify the value as "highest." For example:

```
{
   "id": "1",
   "type": "subscribe",
   "payload": {
         "variables": {},
         "extensions": {},
         "operationName": null,
               "query": "subscription { newEvent(input: { chainCode:
\"<CHAIN CODE>\" offsetType: \"highest\" } ) { metadata { offset }
moduleName eventName detail { oldValue newValue elementName } } }"
   }
}
```

Streaming Scenario

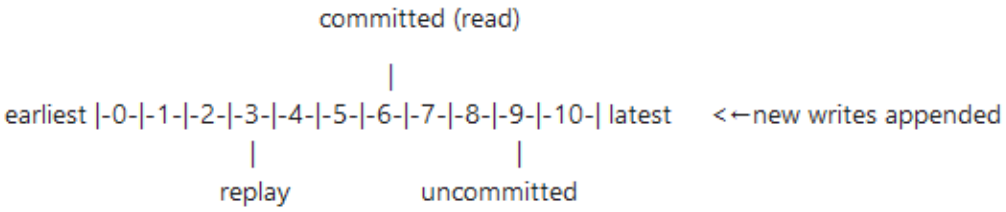
Imagine the following scenario: A stream has produced 10 events over the previous 24 hours. A consumer has connected and consumed events 0 to 6 inclusive, and then disconnected again.

While the consumer is disconnected, events 7 to 10 occur.

If the consumer connects again without specifying an offset, then events 7 to 10 are sent to the caller followed by subsequent events.

If the consumer connects again and specifies an offset of 3, then events 3 to 10 are sent to the caller followed by subsequent events.

If the consumer connects with the offsetType highest, then only event 10 is sent, followed by subsequent events.



Therefore, only use the offsetType highest if you have no interest in historic events because you will no longer receive them. Similarly, you can never maintain the sequence of events if you use offsetType highest.

A good use case for offsetType highest is when a resource is changing rapidly, and you are only interested in the last state of the resource.

> **Note:**
>
> Do not use offsetType highest if you must synchronize an external system with OPERA Cloud as it would get (and remain) out of sync.

Interpreting the Event

Determining Changes

Three key fields are used to determine what has changed:

- elementName
- oldValue
- newValue

New Records

When receiving the record of a new Reservation, the payload will include the following for each field in the Reservation:

```
{
   "elementName": "First Name",
   "newValue": "NewFirstName"
   "oldValue": null
}
```

> **Note:**
>
> For the polling API, the "oldValue" will not be in the payload.

Updated Records

The following scenarios reflect when a record is updated and a field is changed:

1. If a field was blank before the update,

   The polling API returns:

   ```
   {
      "dataElement": "First Name",
      "newValue": "NewFirstName",
   }
   ```

   The streaming API returns:

   ```
   {
      "elementName": "First Name",
      "newValue": "NewFirstName",
   ```

```
   "oldValue": ""
}
```

2. If the field is updated with a blank value,

   The polling API returns:

```
{
   "dataElement": "First Name",
   "oldValue": "TheOldFirstNameValue"
}
```

   The streaming API returns:

```
{
   "elementName": "First Name",
   "newValue": "",
   "oldValue": "OldFirstName"
}
```

3. If the field was not changed during this event,

   The polling API returns:

```
{
   "elementName": "First Name",
   "newValue": "MyFirstName",
   "oldValue": "MyFirstName"
}
```

   By default, the streaming API returns the same fields. However, if the "delta" input variable is set to true, the streaming API returns only fields whose value has changed. In this scenario, the elementName "First Name" would not be sent by the streaming API.

Removed Fields

If a field has been removed, this is reflected as:

```
{
   "elementName": "First Name",
   "oldValue": "OldFirstNameValue"
}
```

Retrieving Unchanged Fields

As you can see in Updated records, if a field remains unchanged neither the "oldValue" nor "newValue" is sent. To retrieve the current (unchanged) value, make a GET call to the resource using the "primaryKey" field that is sent in the event payload. For example, if the event is UPDATE RESERVATION, the primaryKey will be a reservationId, so the full resource can be obtained using the getReservation API.

Comparing the Event Payload from Polling and Streaming APIs

Polling API Business Event payload


```
{
    "businessEventData": [
        {
            "businessEvent": {
                "header": {
                    "moduleName": "Reservation",
                    "actionType": "NEW RESERVATION",
                    "actionId": "1234567",
                    "primaryKey": "987654",
                    "publisherId": "15951",
                    "createdDateTime": "2021-06-03 16:45:48.0",
                    "hotelId": "ABC123"
                },
                "detail": [                    {
                        "dataElement": "FIRST NAME"
                    },
                    {
                        "dataElement": "CONFIRMATION NO",
                        "newValue": "123456789"
                    }
                ]
            },
            "businessEventId": {
                "id": "1234567"
            }
        }
    ]
}
```


Streaming API Business Event payload


```
{
  "data": {
    "newEvent": {
      "metadata": {
        "offset": 100,
        "uniqueEventId": "0ed06ced-843e-4e35-86ec-e2564cf495ee"
      },
      "moduleName": "Reservation",
      "eventName": "NEW RESERVATION",
      "primaryKey": "123456",
      "timestamp": "2021-06-03 16:45:48.000",
      "hotelId": "ABC123",
      "publisherId": "15951",
      "actionInstanceId": "222222",
      "detail": [
        {
          "newValue": "NewFirstNameValue",
          "oldValue": "",
```

```
            "elementName": "FIRST NAME",
            "scopeFrom": "",
            "scopeTo": "",
            "elementType": null,
            "elementRole": null,
            "elementSequence": null
          },
          {
            "newValue": "123456789",
            "oldValue": "",
            "elementName": "CONFIRMATION NO",
            "scopeFrom": "",
            "scopeTo": "",
            "elementType": null,
            "elementRole": null,
            "elementSequence": null
          }
        ]
      }
    }
}
```


Table 12-3 Comparing Fields in the Polling and Streaming APIs

Section	Field Name in Polling	Section	Field Name in Streaming	Use
Header	moduleName	Event Header	moduleName	Grouping of events, useful guide to which API to use if unchanged data are needed.
Header	actionType	Event Header	eventName	Name of the event that occurred.
Header	actionId			Number of the event emitted from OPERA.
		Event Header	actionInstanceId	Number of the event emitted from OPERA.

Table 12-3 (Cont.) Comparing Fields in the Polling and Streaming APIs

Section	Field Name in Polling	Section	Field Name in Streaming	Use
Header	primaryKey	Event Header	primaryKey	OPERA internal ID of the resource on which the event occurred. For example, an UPDATE RESERVATION event occurs on a reservation resource, so the primaryKey is the reservationId.
Header	parentActionId			
Header	publisherId	Event Header	publisherId	User reference Id of the user who created the transaction in OPERA.
Header	createdDateTime	Event Header	timestamp	Date and time the event occurred.
Header	hotelId	Event Header	hotelId	OPERA internal Id of the hotel in which the event occurred.
		Subscription	chainCode	OPERA Shared Security Domain organization code within which the event occurred.
		Metadata	offset	Offset number of the event emitted from OHIP.
		Metadata	uniqueEventId	Unique identifier for the event emitted from OHIP.
Detail	dataElement	Detail	elementName	Name of the field that changed.

Table 12-3 (Cont.) Comparing Fields in the Polling and Streaming APIs

Section	Field Name in Polling	Section	Field Name in Streaming	Use
Detail	oldValue	Detail	oldValue	Old value of the field that changed (see notes in the 'Determining Changes' section)
Detail	newValue	Detail	newValue	New value of the field that changed (see notes in the 'Determining Changes' section)
Detail	scopeFrom	Detail	scopeFrom	Beginning of the data range for which the business event is valid. For example, if a rate code or rate amount is different for different stay dates and the rate is updated, the scopeFrom determines the start of the stay date range that was updated.
Detail	scopeTo	Detail	scopeTo	End of the data range for which the business event is valid. For example, if a rate code or rate amount is different for different stay dates and the rate is updated, the scopeTo determines the end of the stay date range that was updated.

Errors

All errors result in the connection being broken and require the below fixes before connecting again.

Table 12-4 Potential Streaming API Errors and Fixes

Error Returned	How does this Happen?	How to Fix	Error Type
429 Too Many Requests	Sending more than 150 requests to upgrade to WebSocket and sending the requests within 2 minutes from the same application.	Retry after 2 minutes.	HTTP
499	The client disconnected before the server was able to send a response to the initial HTTP upgrade, so the WebSocket was not opened.	Ensure your calling architecture is leaving the connection open long enough for the server to reply.	HTTP
1000 Normal Closure	The WebSocket was disconnected by the caller	Reconnect from the Authentication message.	WebSocket
1001 Going Away: Going away	Maintenance by Oracle	Reconnect from the Authentication message.	WebSocket
1006 Abnormal Closure: Abnormal Closure	Connection broken	Wait 4 seconds and then reconnect from the Authentication message.	WebSocket
Error: Unexpected server response: 400	Incorrect key or URL	• Check the sha256 hash of the application key is correct. • Check that the application key is subscribed to consume events and that the event subscription is approved. • Check the URL matches the environment listed in the Developer Portal. • Check that the environment is Streaming Enabled.	HTTP

Table 12-4 (Cont.) Potential Streaming API Errors and Fixes

Error Returned	How does this Happen?	How to Fix	Error Type
4400 Error: Bad Request - Invalid GraphQL Subscription	Invalid field in subscription message	• Check the response body which includes details of the invalid field. For example, "Cannot query field XYZ on type ABC" or "Value for <FIELD> is invalid - Expected ABC to match pattern ZYZ." • Check that the fields in the subscription request match the OHIP schema Github link. • Reconnect from the Authentication message.	WebSocket
4401 Error: Unauthorised - Invalid credentials	Invalid credentials sent in the Authentication message	• Check you are sending the application key, not the sha256 hash of the application key. • Check that the application is subscribed to consume events and that the event subscription is approved. • Check that the oAuth token is valid and current.	WebSocket

Table 12-4 (Cont.) Potential Streaming API Errors and Fixes

Error Returned	How does this Happen?	How to Fix	Error Type
4403 Error: Forbidden - You are not authorized to access this resource	Invalid chainCode in subscription message	• Check that the chain code in the subscription message matches the chain code being accessed. This is the value at the start of the integration that the user sent to obtain the oAuth token. • Check that streaming is enabled for the environment you are accessing • Reconnect from the Authentication message.	WebSocket
4406 Error: Subprotocol not acceptable	Failing to include the WebSocket protocol header.	Include the header "Sec-WebSocket-Protocol: graphql-transport-ws"	WebSocket
4408 Error: Disconnect	Connection initialization time out	Send the Authentication message within 5 seconds of opening the connection.	WebSocket

Table 12-4 (Cont.) Potential Streaming API Errors and Fixes

Error Returned	How does this Happen?	How to Fix	Error Type
4409 Too Many Requests	More than one client or process is trying to consume events from the same gateway using the same application key and chain code. This often happens in the following circumstances: 1. When consuming events in code and in Postman/ GraphiQL. 2. Where multiple developers are trying to consume events from a single application. 3. Where an application has spawned additional threads all consuming events from the same gateway, application key, and chain code. 4. When disconnecting without sending the Complete message.	Ensure that only one client or process consumes events from a given gateway using a given application key and chain code. Ensure that you send the Complete message before disconnecting from the WebSocket.	WebSocket
4429 Error: Too Many Requests	Resending the subscription message when the connection is already open Resending the connection_init message when the connection is already open	The WebSocket connection is already open, so it cannot be reopened. Look for the connection_ack message and do not resend connection_init if you have received a connection_ack.	WebSocket
4500 Server Error - No configuration found. If this problem persists, contact Oracle Customer Support at the Customer Support Portal.	A Subscribe message was sent but either there are no events subscribed for this application, or the subscription is not approved by the environment owner.	Check that the application is subscribed to events and shows as approved for that environment.	WebSocket

Table 12-4 (Cont.) Potential Streaming API Errors and Fixes

Error Returned	How does this Happen?	How to Fix	Error Type
4500 Unable to find configuration between Application, Chain, and Environment. Please ensure your application has access. If this problem persists, contact Oracle Customer Support at the Customer Support Portal.	A piece of internal configuration might be missing.	Ensure your application is approved to consume events and that the URL you are calling is a streaming enabled gateway. If the problem persists, contact Oracle Customer Support at the Customer Support Portal.	WebSocket
4500 Internal Server Error received 6 minutes after sending the subscribe message.	When subscribing, the consuming application should wait up to 6 minutes before receiving any events. If the subscription is valid and an internal issue has occurred, this message is received 6 minutes after sending the subscribe message.	Reconnect and send the Authentication message and then send the Subscription message.	WebSocket
4501 Socket closed with error 4501 Not Supported Environment for subscriptions	Connecting to an environment that does not support streaming Business Events.	Ensure the environment shows as "Streaming Enabled" on the Environments tab of the developer portal. Streaming Business Events from OPERA requires a minimum of OPERA Cloud version 22.3.0.1.	WebSocket
4504 Socket closed with error 4504 Service Timeout	This error occurs when the OCI Streaming Service goes down for more than 30 seconds.	Try again after 15 seconds. If this problem persists, contact Oracle Customer Support at the Customer Support Portal.	Websocket

Connecting to the Streaming API

Overview

The streaming API is built on GraphQL over WebSocket protocol. This protocol defines that a given stream (identified by application key, URL, and chainCode) can only be consumed by one application/process/thread. Therefore, do not use multithreading when connecting to the Streaming API.

Prerequisites

Streaming Business Events from OPERA Cloud requires a minimum version of OPERA Cloud 22.3.0.1.

Before connecting to consume Business Events from the WebSocket, verify the following prerequisites are met:

- Onboard to OHIP by following the steps in Getting Started for Partners.
- Create an application by following the steps in Registering an Application.
- Take a note of the application key.

> **Note:**
>
> It is important that you write down the application key.
>
> You can only connect to the streaming API from one application at a time using the same application key. Using a single application key to simultaneously access the API from multiple applications will stop the streaming service.

- Add an environment from which to consume events. For more information, see Environments (Gateways and Credentials). Take note of the following:
  - **ClientId and ClientSecret** — These are needed to obtain the oAuth token.
  - **Gateway URL**
- Check that the environment is streaming enabled. It will have a label "Streaming Enabled" if this is the case.
- Set up the template of events. For more information, see Working with Events and Creating a Template of Events for an Application.
- Subscribe to consume the events from an environment. For more information, see Working with Events and Subscribing to Events.

If the environment is the partner sandbox or an environment owned by your organization, the request to consume events is automatically approved. If not, environment owners (usually customers) must approve the request to consume events from their environment. To do this, following the process in Working with Events and Approving Events.

Once the request to consume events is approved, the event is listed as "Approved" on the Developer Portal.

GraphQL

The streaming API uses GraphQL subscriptions delivered via WebSocket. To learn more about this technology, read our blog post.

WebSocket Authorization

Access to the streaming API is protected by oAuth and an application key. The application key comes from the application you created in Registering an Application.

To obtain the oAuth token, follow the processes described in Using the Oracle Hospitality APIs and in Authenticating to Oracle Hospitality Property APIs. For an example of the oAuth token, visit the Get OAuth Token page on the Postman API Platform site at https://www.postman.com/hospitalityapis/workspace/oracle-hospitality-apis/request/15729853-58d0804e-b607-4c37-bf2f-0656c254573c.

Information required to call the WebSocket

To connect to the streaming API, the following pieces of information are required:

- Application Key (Create an application by following the steps in Registering an Application.)
- Valid oAuth Token
- URL

> **Note:**
>
> If the URL listed on the Environment panel in the Developer Portal is "https://www.oracle.com," the URL for the streaming API should be "wss://www.oracle.com/subscriptions." The change from "https" to "wss" is required for connection via WebSocket.

Browsing the OHIP Schema

The Documentation Explorer is located on the right-side of the screen.

1. Click the **Subscription** link to browse the schema for creating the subscription.

   The Subscription object requires you to specify a "NewEventInput" object containing the chainCode and the offset.
2. Click **NewEventInput** to view the data type and limits on these fields.
3. Click the **Subscription** arrow to go back to the definition of the Subscription object.

   Returned in the response to this subscription is the type called "EventHeader."
4. Click the **EventHeader** to view details of the fields that can be returned (if included in the query).
5. Click any of these fields to see the object definition, data types, and limits on the fields.

   One of the response fields is the "detail" array, which includes fields such as newValue, oldValue, and elementName (that is, the name of the data element that changed).
6. Compose your GraphQL query using the left-side of the screen, which includes intellisense to speed up the completion of the query.
7. Click **Start** to open the WebSocket connection. Any events meeting the subscription configured in the Developer Portal are sent and appear on the right-side of the screen, but include only those fields specified in the subscription query on the left-side of the screen.

   The right-side of the screen only shows the latest event. Since many events will be sent at once, use the developer tools in the browser to view the full list of sent events.
8. Click **Stop** to disconnect the WebSocket.

> **Note:**
>
> When the Auth Token expires, the connection must be severed and then reestablished with a fresh Auth Token every hour.

Connecting to the Streaming API via GraphiQL

To better visualize the streaming API, a playground application is available that uses a tool called GraphiQL. The GraphiQL tool is often utilized with GraphQL APIs. It is available as a standalone web page from our public Github repository.

> **Note:**
>
> The GraphiQL page has a limitation that a single browser can only stream events from one application at a time. GraphiQL cannot be open in multiple tabs or windows consuming events from different applications.

Connecting via GraphiQL

1. Enter the **URL** (see the 'Information required to call the WebSocket' section in WebSocket Authorization).
2. Enter an **Auth Token** as described in WebSocket Authorization.
3. Enter the **API key** as it is listed in the Developer Portal.

   This is the not the sha256 hash of the API key.
4. Click the **Start** button.
   - This stores the values on your machine not on the server. Only you can view these saved values.
   - This populates the Socket Key.

   The connection is open if you see the three boxes appear below the Start button.
   - The left-hand box holds the subscription request. By default, this includes a getHelp query and an example subscription.
   - The middle box holds the latest response.
   - The right-hand box is a navigable representation of the GraphQL schema showing the fields you can include in the subscription.

   Since the middle box holds only the latest response, you should open developer tools in your browser, usually the Network tab, so that all events sent on the stream are visible.
5. Click the **play** button to open the WebSocket connection. By default, this returns the getHelp query with useful links to this guide.
6. Comment out the getHelp query and remove the comments in front of the example subscription.

   Use **CTRL + /** to comment or uncomment.

7. Optionally, add elements from the GraphQL schema into the subscription. Press **CTRL + space** to list fields from the schema.
8. Click the **play** button to open the WebSocket connection. All events that occurred after the subscription was approved are now sent with the latest event shown in the middle pane. The WebSocket connection remains open until you click the stop button. This means if any of the events chosen in the Developer Portal occur in the subscribed hotel(s), the events are immediately sent on the streaming API and appear in the middle pane.

> **Note:**
>
> When the Auth Token expires, the connection must be severed by clicking the stop button and then reestablished with a fresh Auth Token every hour. As noted in the Broken Connections topic, OHIP sends any events that occurred between stopping and restarting the WebSocket connection.

Viewing Historic Events

To view historic events, use your web browser's developer tools. For example:

1. In Mozilla Firefox, press **F12** to open developer tools.
2. Click the **Network** tab.
3. Click **WS** to show only WebSocket requests.
4. After clicking the **play** button, click the last web service request sent.
5. When the details appear, click the **Response** tab.

You will see all the sent business events. You can view each business event inside the developer tools by clicking it, or you can copy it by right clicking the business event in the list of responses and clicking **Copy Message**.

Connecting to the Streaming API with Postman

Postman can be accessed online or from the Postman API Client https://www.postman.com/product/api-client/. You must be signed in to a Postman workspace to use WebSocket APIs in Postman.

> **Note:**
>
> While it is possible to support WebSocket via Postman, it is not yet possible to save WebSocket requests in Postman except while signed in to a Postman workspace.
>
> You can connect to a given stream from only one application and thread at a time. Ensure the stream (identified by the application key, URL, and chainCode) is not being used by any other applications.
>
> Postman does not support the sending of a ping, so any postman connection closes after 5 minutes.

Consuming Events in Postman

To consume events in Postman, follow one of the two options below:

Option A

1. Fork this Postman collection and fork this Postman environment file.
2. Use this oAuth API example to first obtain an oAuth token.

Option B

1. Select **New** and then select **WebSocket Request**.
2. Enter the **URL** in the following format: wss://www.oracle.com/subscriptions (as described in ‘Information required to call the WebSocket’).
3. Add the **query parameter** (see the ‘Query Parameters’ section).
4. Add the **headers** (see the ‘Headers’ section).
5. Send the **Authentication message**.
6. Send the **subscription message** within 10 seconds of the Authentication message.
7. View the **events** returned.

Query Parameters

The GraphQL subscription resource has one mandatory query parameter named "key." The value is a sha256 hash of the application key obtained from Registering an Application.

If LINUX environments are running, echo-n ABC-123 | sha256sum (where "ABC-123" is the application key) will return the required hash. In Microsoft Windows environments, use a code snippet or download GitBash or Windows Subsystem for Linux. To run this command, visit the Install WSL page on the Microsoft website at https://docs.microsoft.com/en-us/windows/wsl/install. Alternatively, you can use an online sha256 hash generator.

> **Note:**
>
> Ensure only lowercase hashes are used.

To add this to Postman:

1. In Postman, click the **Params** tab
2. In the first column, enter the value "key."
3. In the second column, enter the value of the 256 hash that was calculated above.

Headers

The Oracle Hospitality streaming API uses the GraphQL-WS protcol passed in headers:

1. In Postman, click the **Headers** tab.

2. In the first column. enter the value "Sec-WebSocket-Protocol."
3. In the second column, enter the value "graphql-transport-ws."

ID

All messages, except the "ping" message (see Keeping the Stream Open), require you to specify an ID in the request payload.

The ID should be a GUID to avoid collision with other consumers.

The same ID value must be used throughout a stream's life.

Authentication Message

Send the authentication "Connection Initialization" message before sending the subscription message.

1. In Postman, click the area marked **New Message** and enter the following:

```
{
  "id": "<GUID>"
  "type": "connection_init",
  "payload": {
      "Authorization": "Bearer <OAUTH TOKEN>
      "x-app-key": "<APPLICATION KEY>"
   }
}
```

Where:

- oAuth Token is the access_token returned from the oAuth request. For more information, see Web Socket Authorization.
- APPLICATION KEY is the application key from Registering an Application.

Once the authentication message is successfully received, the following "Connection Acknowledged" message is returned:

```
{
   "type": "connection_ack",
   "payload": {
         "applicationName":"<APPLICATION NAME>"
   }
}
```

Where APPLICATION NAME is the name of the application that corresponds to the application key. This validates that the correct application is being used.

In Postman, the connection will show as "CONNECTED."

Subscription Message

1. Send the subscription message within the next 10 seconds now that the connection is authenticated and connected.

2. In Postman, overwrite the Authentication message request body with the following request body:

```
{
   "id": "<GUID>",
   "type": "subscribe",
   "payload": {
       "variables": {},
       "extensions": {},
       "operationName": null,
       "query": "subscription { newEvent(input: { chainCode:
\"OHIPCN\" }) { metadata { offset } moduleName eventName detail
{ oldValue newValue elementName } } }"
    }
}
```

   The "query" is the GraphQL query formatted against the OHIP schema. It contains the fields to be returned along with business events.

   For the full schema of what can be sent in the Subscribe request body, consult the Documentation navigator in GraphiQL.

3. For the chainCode, specify the "tenant" entered when creating the integration user. Refer to step 1 in Authenticating to Oracle Hospitality Property APIs. This is also prefixed on your integration username.

4. Click the **Send** button.

Viewing the Events Returned

In Postman, messages from the server begin with a down arrow.

To see the JSON response payload:

1. Click to expand the message.

The payload includes only those elements from the "query" specified in the Subscription message.

When multiple events are pending subscription, each event is returned in sequence.

Keeping the Stream Open

To keep the connection open, you must send a "ping" request at least every 15 seconds on the WebSocket (see the below JSON example).

```
{"type":"ping"}
```

> **Note:**
>
> You are not charged for this "ping" request.

An example implementation is listed on GitHub in the Recipes section under ‘Client usage with ping/pong timeout and latency metrics.’

The server also sends pings to the client. It is important that as soon as it receives a "ping" from the server the client return a "pong" message to this request from the client as documented in the Protocol:

```
{"type":"pong"}
```

Updating Authentication

The streaming API is secured by an oAuth token whose life is limited to one hour (sixty minutes). To continue receiving events, you must keep the stream open by doing the following:

1. Close the WebSocket (see Disconnecting the WebSocket for more information).
2. Request a new oAuth token from the oAuth API.
3. Reopen the websocket and specify the new "access_token" received from the oAuth API.

Disconnecting the WebSocket

WebSockets are designed to stay open. When planning to disconnect from a WebSocket (refer to the Broken Connections topic), be aware of the following:

- Events will queue up while disconnected, but only 7 days of events are retained. This can be a large volume of events, so before reconnecting, verify your consuming architecture can support the volume.
- Send the "Complete" message (see the example below).

Sending the Complete Message

Before disconnecting, ensure you send the below Complete message (see the protocol for more information):

```
{
    "id": "<GUID>",
    "type": "complete"
}
```

You must send the Complete message to connect to the stream (identified by the application key, URL, and chainCode). Failure to send the Complete message will make it impossible to connect to the stream. If a subscribe message is sent to a stream that has not yet received a complete message, a 4409 error will occur.

Reconnecting after Complete

Ensure there is a minimum of 500 ms between sending the "Complete" message to close a WebSocket connection and sending the next "Subscribe" message to reopen a WebSocket connection.

Streaming Best Practices

> **Note:**
>
> You must connect to the WebSocket at least once every 24 hours. If not, then you must send the optional "offset" input parameter together with the value of the last offset you received.
>
> Ensure there is a minimum of 500 ms between sending the "complete" message to close one WebSocket connection and the next "subscribe" message to reopen a WebSocket connection.

The Streaming API is not throttled. As events are produced, they are immediately sent out. Potentially, this means many events will be pushed at the same time and create a backpressure.

To cope with this potential backpressure, it is essential the consuming architecture can scale, for example, by buffering before writing to back-end systems. If the backpressure exceeds the ability of the consuming architecture to scale, you can disconnect the WebSocket and then restart it later once more capacity is available in the consuming architecture. Any events that occurred since the WebSocket disconnect will be sent as soon as you reconnect.

While no option is currently available to clear the backlog of events, the "latest" parameter allows you to skip to the latest Business Event in the stream and continue processing from there.

For more information on anti-patterns and best practices on the streaming API, refer to the Anti-Patterns topic.

Streaming Troubleshooting

If you are experiencing issues consuming the Streaming API, check the following:

Environment

- Verify the OPERA environment from which you wish to stream events is listed on the Environments tab of the developer portal.
- Verify the OPERA environment shows as Streaming Enabled on the Environments tab of the developer portal.
- Verify you have valid credentials to obtain the oAuth token (see API Troubleshooting for more information).
- Verify the oAuth token is current. They expire after 60 minutes. If not, request a new oAuth token.Check the oAuth token is current. They expire after 60 minutes. If not, request a new oAuth token.

Configuration

Ensure your application streaming configuration is both requested and approved (see Working with Events in the Developer Portal for more information).

Postman

Postman cannot send "ping" on an open WebSocket, so the connection will automatically close. When it closes, you must resend the "init" message and then resend the "subscribe" message.

It is not yet possible to save WebSocket requests in Postman, except when signed in to a Postman workspace.

GraphiQL

If you receive errors from GraphiQL, ensure you have it opened only once. It cannot support streaming events from different applications in different tabs (see Connecting to the Streaming API via GraphiQL for more information).

Not Receiving the Expected Data

If you are not receiving the pieces of data you expected, check the subscribe message to ensure it includes the expected field.

The hotelId will always be null for chain-level entities, such as profiles, which tend to be shared across all hotels in the chain.

Verify this page lists the expected data values for the event(s) to which you are subscribed (see the ‘Business Events - Activity’ heading under the Configuring Business Events topic in the OPERA Cloud User Guide).

Receiving Unexpected Events

Keep in mind that a single action, such as checking a guest in, can trigger multiple business event notifications because that single action modifies multiple resources.

For integration partners developing against the sandbox, keep in mind that the actions taken by other partners in the sandbox will generate events.

Not Receiving Expected Events

If you are not receiving all new reservations, it is possible the customer has an external CRS setup. Request the customer to set up a "publisher" on your external system (the external system code is displayed on the **Application**, **Events**, **Subscribed** tab in the developer portal) following this process (see the ‘Managing External System Publishers’ heading under the Configuring External Systems topic in the OPERA Cloud User Guide).

If you are not receiving any events, it is possible you have not connected for greater than 24 hours. In this case, stop the WebSocket and reopen it, specifying in the subscribe message the last offset you received.

Verify you are not unexpectedly using the "hotelId" filter in the subscribe message.

Getting Overwhelmed with Events

The Streaming API sends events as soon as they are available, so it is important that consuming architecture can scale and potentially buffer events to avoid choking database connections (see Streaming Best Practices for more information).

Other Errors

See the Errors topic for suggested resolutions to common errors.

If the "init" request is failing, ensure you send it within 5 seconds of the HTTP upgrade request.

If the connection keep closing:

- Ensure you send "ping" every 15 seconds (see Keeping the Stream Open for more information).
- Postman cannot send "ping," so the connection will automatically close.
- Wait at least 500ms between closing and re-opening the WebSocket, or you will receive 4409 errors (see Disconnecting the WebSocket for more information).
- Ensure that only one process/thread/user is connected to a given stream (identified by application key, URL, and chainCode) at any one time.
- Ensure you have connected only one WebSocket per application: Connecting more than one WebSocket with the same application key will result in 4409 errors.
- If the socket closes with a 4401 error, obtain a new oAuth token before reconnecting. The socket will automatically close every one hour when the oAuth token expires (see Keeping the Stream Open for more information).
- Receiving a 4403 error:
  - Verify the chain to which you are subscribing in the subscribe message matches the chain for the integration user that was used in the oAuth request.
- Receiving a 4409 error:
  - Ensure only one client or process consumes events from a given gateway using a given application key and chain code.
  - Ensure you send the Complete message before disconnecting from the WebSocket.
  - Ensure you send the ping message to keep the stream open (see Keeping the Stream Open for more information).
  - Reconnect after 5 minutes.

Confused About Implementation?

We adhere to the GraphQL over WebSocket protocol, so ensure your implementation meets this protocol.

An example implementation is available on this GitHub page.

It is suggested that you create a second application using the GraphiQL tool to better distinguish between implementation and configuration issues.

Polling API (pull)

To consume business events generated in OPERA Cloud using the polling APIs, visit OPERA Cloud Integration Processor API and follow the instructions.

Prerequisites

- The customer must be onboarded to consume Business Events via OHIP. For more information, see Getting Started.
- OPERA Cloud properties on OPERA Cloud version 21.2.1+ can configure Business Events in OPERA Cloud without impediment. If an OPERA Cloud

property is below version 21.2.1, the customer can raise a Service Request to apply the OPP_BSEV license without cost, which enables them to configure Business Events.

You can discover your OPERA Cloud version using the following API call:

https://www.postman.com/hospitalityapis/workspace/oracle-hospitality-apis/request/15729853-8dca939b-a6b5-4dca-bc0f-663027d11a88

Configuring the Polling Subscription

The polling API requires the owner of the OPERA Cloud environment to configure the subscription in OPERA Cloud.

1. Verify a chain-level user has the below tasks assigned. To assign these tasks, follow the steps in the Assigning Tasks to a Role topic in the OPERA Cloud User Guide.
   a. **Interfaces Admin** — **Property Interfaces** —**External Systems**:
      - **New/Edit External Systems**
   b. **Toolbox**:
      - **External Databases**
   c. **Interfaces Admin** — **Property Interfaces** —**Business Events**:
      - **New/Edit Business Events**
2. Create an external system for your organization by following the steps in Configuring External Systems topic in the OPERA Cloud User Guide.

> **Note:**
>
> Make sure you remember the external system code you created.

3. Create an external database by following the steps in the Configuring External Databases topic in the OPERA Cloud User Guide.
4. Configure the chosen events on the external system by following the Configuring Business Events steps in the OPERA Cloud User Guide. For the list of events that can be consumed, see the Business Events Data Elements guide.

> **Note:**
>
> If there are challenges with this process, the owner of the OPERA Cloud environment should contact Oracle Customer Support at the Customer Support Portal, raise a Technical SR, and select Oracle Hospitality OPERA Cloud as the product.

5. Communicate or make note of the external system code created at step 2.

Consuming Business Events using the Polling API

1. Onboard to OHIP by following the steps in Getting Started for Hoteliers.
2. Create an application by following the steps in Registering an Application. Take a note of the application key.

> **Note:**
>
> Make sure you remember the application key.

3. Add an environment for the environment from which you need to consume events. For more information, see Environments (Gateways and Credentials). Take note of the following:
   - **ClientId and ClientSecret** — These are needed to obtain the oAuth token.
   - **Gateway URL**
4. Determine the hotelId of the hotel from which to consume business events.
5. Call the FetchBusinessEvents API and specify the external system code as configured in the 'Configuring the Polling Subscription' task above.

   Do not use the external system code created by the streaming API or the events will end up out of sequence.
6. Specify the x-hotelId as the hotelId from step 4 above.

   If the following response is received from the FetchBusinessEvents API, contact Oracle Customer Support at the Customer Support Portal to raise a technical Service Request quoting this error. (This is resolved in OPERA Cloud version 21.2.1.)

```
"shortText": "Failed to initialize <your external system code>
 interface for <hotelCode> resort: <your external system code> is
not registered as an application user."
```

> **Note:**
>
> A maximum of 20 events can be retrieved in one call. A maximum of 300 requests per minute per gateway can be made to the polling API. If a larger amount of events is required, than these limits allow consider the streaming API.
>
> There is a limit of 300 requests per minute for the fetchBusinessEvents API.

For information on how to interpret the event responses and for the differences between the polling and streaming APIs, see Interpreting the Event.

For more information on the polling API, watch OPERA Cloud Business Events.

13

Moving to Production for Environments with Resource Owner Group Authentication (SSD)

This chapter consists of two sections:

- **Hoteliers Moving to Production** — This section explains how a hotelier would move their own integrations to Production once the development and testing phase is complete.
- **Partners Moving to Production** — This section explains how a partner would move an application to Production once testing and development is complete (otherwise known as "activating an application with a customer chain"). In this scenario, all actions are on the Partner side within the Partner Organization Developer Portal, and the only action required by the customer is to approve the integration, either by the Chain Administrator approving the integration user submitted by the partner (for which the Chain Administrator will receive notice in their Shared Security Domain) or by approving the integration in the Oracle Hospitality Integration Platform Developer Portal.

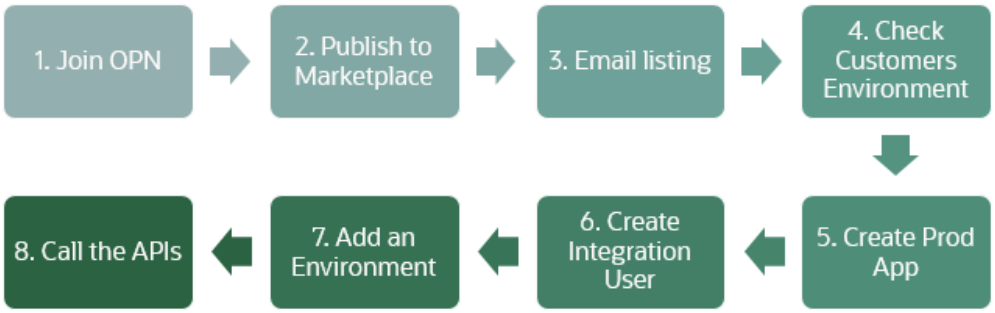
Hoteliers, note that Oracle does not perform any certification for integrations. It is recommended that hoteliers first trial partner integrations in the hotelier's UAT environment, perform their own audits and assessments of the integration, and transition vendors to production only when you are finally satisfied. Hoteliers can use the Oracle Hospitality Integration Platform Developer Portal Analytics tab to assist their auditing of partner integrations. Hoteliers could consider hiring Oracle Hospitality Professional Services to conduct a vendor review, but this is optional.

In the event of any concerns with partner integrations, hoteliers can disconnect them by revoking the WSACCESS role for Resource Owner environments behind Oracle Shared Security Domain.

Hoteliers Moving to Production

1. Create and register an application. Ensure you choose “Production” as the environment.
2. Obtain credentials.
3. View environment.
4. Call the APIs.

Partners Moving to Production



> **Note:**
>
> You only need to perform steps 1 to 3 on your first integration.

1. Join OPN. Please note that the minimum level required for OHIP is the OPN Member Level. For more information, visit How Do I Join Modern OPN?
2. Publish your app to the Marketplace. To publish your solution, visit Publish Services and Applications on the Oracle Cloud Marketplace and follow the steps on this page.
3. Contact us via Slack once your application is listed in the marketplace under the OHIP product category. If you're not part of the OHIP Slack Community, email us at hgbu_integrations_provisioning_grp@oracle.com to request your private channel.
4. Check the customer's environment by asking customers to verify they are running OPERA Cloud and have an active subscription for OPERA Cloud Foundation. Customers can verify this with their account manager if in doubt.

   If you are using both Early Adopter (v0) and v1 APIS, you will need to subscribe your app to both.

   If your solution involves calling Asynchronous APIs, contact Oracle Support to check if they are supported in the production environment being called. Include the OPERA Cloud URL or gateway URL in the support request.
5. Create and register an application. Ensure you choose "Production" as the environment.
6. Obtain credentials.
7. View environment.
8. Call the APIs.

FAQ

How do I know the customer has approved my integration?

You will receive an email with a subject line.

How do I know the URL for the vendor registration portal?

If you have not received this from customers, ask them for the URL with which they access OPERA Cloud.

14

Moving to Production for Environments with Client Credentials Authentication (OCIM)

This chapter consists of two sections:

- **Hoteliers Moving to Production** — This section explains how a hotelier would move their own integrations to Production once the development and testing phase is complete.
- **Partners Moving to Production** — This section explains how a partner would move an application to Production once testing and development is complete (otherwise known as "activating an application with a customer chain"). In this scenario, all actions are on the Partner side within the Partner Organization Developer Portal, and the only action required by the customer is to approve the partner connection by approving the integration in the Oracle Hospitality Integration Platform Developer Portal.

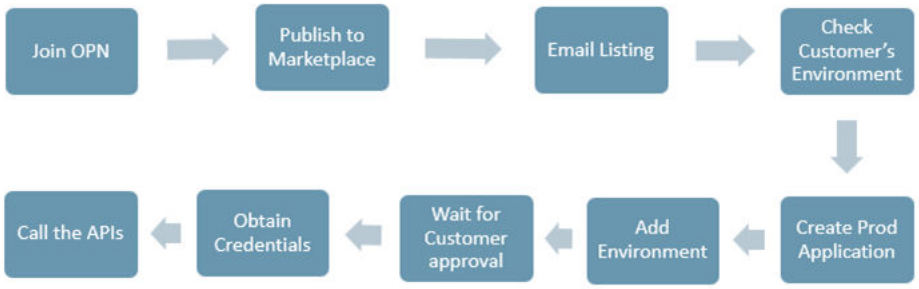
Hoteliers, note that Oracle does not perform any certification for integrations. It is recommended that hoteliers first trial partner integrations in the hotelier's UAT environment, perform their own audits and assessments of the integration, and transition vendors to production only when you are finally satisfied. Hoteliers can use the Oracle Hospitality Integration Platform Developer Portal Analytics tab to assist their auditing of partner integrations. Hoteliers could consider hiring Oracle Hospitality Professional Services to conduct a vendor review, but this is optional.

In the event of any concerns with partner integrations, hoteliers can disconnect them by revoking access in the Oracle Hospitality Integration Platform Developer Portal for Client Credentials environments behind OPERA Cloud Identity Manager.

Hoteliers Moving to Production

1. Create and register an application. Ensure you choose “Production” as the environment.
2. Obtain credentials.
3. View environment.
4. Call the APIs.

Partners Moving to Production



> **Note:**
>
> You only need to perform steps 1 to 3 on your first integration.

1. Join OPN. Please note that the minimum level required for OHIP is the OPN Member Level. For more information, visit How Do I Join Modern OPN?
2. Publish your app to the Marketplace. To publish your solution, visit Publish Services and Applications on the Oracle Cloud Marketplace and follow the steps on this page.
3. Contact us via Slack once your application is listed in the marketplace under the OHIP product category. If you're not part of the OHIP Slack Community, email us at hgbu_integrations_provisioning_grp@oracle.com to request your private channel.
4. Check the customer's environment by asking customers to verify they are running OPERA Cloud and have an active subscription for OPERA Cloud Foundation. Customers can verify this with their account manager if in doubt.

   If you are using both Early Adopter (v0) and v1 APIS, you will need to subscribe your app to both.

   If your solution involves calling Asynchronous APIs, contact Oracle Support to check if they are supported in the production environment being called. Include the OPERA Cloud URL or gateway URL in the support request.
5. Create and register an application. Ensure you choose "Production" as the environment.
6. Add the Customer Environment.
7. Wait for Customer Approval.
8. Obtain credentials.
9. View environment.
10. Call the APIs.

FAQ

How do I know the customer has approved my integration?

The status of the customer environment will change to "Approved" from "Waiting for Approval" in the OHIP Developer Portal.

15

Migrating from Legacy APIs to REST

Integrations with Oracle Hospitality products now utilize REST APIs exposed through Oracle Hospitality Integration Platform (OHIP). As there are many active integrations certified on our legacy integration products, such as OXI, OWS, ADS, HTNG, and Kiosk, this chapter will help Hoteliers and Partners understand the process to migrate an existing legacy interface to the new REST APIs.

In OHIP, we have the following Oracle Hospitality products:

- Property APIs (OPERA Cloud)
- Distribution APIs

Depending on your integration requirements, there are different scenarios for integrating with Property APIs versus Distribution APIs.

OPERA Xchange Interface (OXI)

The OXI Interface works on an asynchronous pattern which follows the below flow:

1. A message is sent from an external system to OXI.
2. The message sits in a queue.
3. The message is processed.
4. A record is inserted or updated in OPERA Cloud.

Once processed, a response message is returned to the external system, and it again sits in a queue until processed. This same pattern occurs for messages being generated from OPERA Cloud to an external system. The REST APIs work on a synchronous pattern. Conforming to REST architecture, the APIs allow for interaction with RESTful web services.

> **Note:**
>
> The REST APIs use JSON formatting.

You can move your integration from OXI to REST and even enhance it as there is greater functionality available with the REST APIs.

OXI Outbound Messages

For messages sent from OPERA Cloud to an external application, the REST APIs utilize the same Business Event functionality as OXI. Therefore, the same data can be triggered from OPERA Cloud. There are two approaches available for use: polling for Business Events using getBusinessEvents operation or using streaming services. For more information on property Business Events available in OHIP, refer to the Business Events topic that explains both options in detail.

OXI Inbound Messages

Data from the external application to OPERA Cloud can be achieved using REST APIs. The below mapping table shows the OXI messages and shows the equivalent Property REST API to use.

Key Considerations

OXI can synchronize data between the two applications. For example, users could enter a date range of reservations, and a batch of resync messages are sent to the external system. This was often used when a hotel went live with a new interface, ensuring the external system had all the required data, including rates, inventory, restrictions, and so on. The REST Property APIs have Asynchronous APIs, which enable the user to send a REST request to OPERA Cloud for a significant amount of data (for example, 30 days of reservations). OPERA Cloud processes the request, collects all the data, and then allows the data to be fetched using the get request.

OXI provided the ability to map OPERA Cloud codes to external system codes, such as room type codes, rate codes, package codes, and so on. The REST APIs with OPERA Cloud no longer follow this approach, and codes must now be one to one. Alternatively, the external system would need to cater for mapping on their side and ensure that when posting a message to OPERA Cloud, the message contained the OPERA Cloud code(s).

OXI XML messages send a full object in XML format as opposed to business event messages (either pull or push approach), which send key value pairs in JSON format. You will receive a JSON message with the old value and the new value. For more information, see Interpreting the Event.

Rest APIs handle credit card data differently than OXI, which allowed tokenization of card data. With REST APIs, the consumer must perform a step to fetch the token from the Payment Service Provider (PSP) and then use that token in the reservation. This only applies to customers with OPI active.

Table 15-1 OXI to REST

Functional Module	OXI Inbound Operation (External System to OPERA Cloud)	REST API Operation	REST Module
Activity	Create Activity	postActivity	ACT
Activity	Update Activity	putActivity	ACT
Activity	Delete Activity	deleteActivity	ACT
Activity	Fetch Activity	getActivity	ACT
Profile	New Profile	postProfile	CRM
Profile	Update Profile	putProfile	CRM
Profile	Delete Profile	deleteProfile	CRM
Profile	Merge Profile	postMergeProfiles	CRM
Profile	Create Relationship	postProfileRelations	CRM
Profile	Update Relationship	putProfileRelations	CRM
Profile	Delete Relationship	deleteProfileRelations	CRM

Table 15-1 (Cont.) OXI to REST

Functional Module	OXI Inbound Operation (External System to OPERA Cloud)	REST API Operation	REST Module
Profile	Lookup	getExternalProfiles	CRM-OUTBOUND
Profile	Download	downloadExternalProfile	CRM-OUTBOUND
Reservations	Add Reservation	postReservation	RSV
Reservations	Edit Reservation	putReservation	RSV
Reservations	Cancel Reservation	postCancelReservation	RSV
Reservations	Share Reservations	postCombineShareReservations	RSV
Reservations	checkout	postCheckOut	CSH
Reservations	checkin	postCheckIn `	FOF
Reservations	Cancel Checkin(reverse check in)	deleteCheckin	FOF
Reservations	Fintrx	postDepositPayment	CSH
Reservations	Turnaway	postTurnawayReservation	RSV
Reservations	Routing	putRoutingInstructions	RSV
Reservations	Reinstate	putReinstateReservation	CSH
Reservations	No Show	putReservationStatusToNoShow	BOF
Reservations	Reservation Trace	getTracesByReservation	RSV
Reservations	Waitlist Reservation	postReservation	RSV
Reservations	Reactivate Waitlist Reservation	putReservation	RSV
Reservations	Room Move	moveInHouseGuest	FOF
Allotment (Group Blocks)	New Allotment	postBlock	BLK
Allotment (Group Blocks)	Edit Allotment	putBlock	BLK
Allotment (Group Blocks)	New Allotment - Header	postBlock	BLK

Table 15-1 (Cont.) OXI to REST

Functional Module	OXI Inbound Operation (External System to OPERA Cloud)	REST API Operation	REST Module
Allotment (Group Blocks)	New Allotment - Header with Detail	postBlock	BLK
Allotment (Group Blocks)	Allotment - Header - Status Change	putBlockStatus	BLK
Allotment (Group Blocks)	New Block Reservation	postReservationByBlock	RSV
Allotment (Group Blocks)	Update Block Reservation	putBlockReservations	BLK
Allotment (Group Blocks)	Fetch Block Reservation	getBlockReservations	BLK
Allotment (Group Blocks)	Update Block Grid Allocation	putBlockAllocation	BLK
Allotment (Group Blocks)	Block Grid Wash	putBlockAllocationWash	BLK
Allotment (Group Blocks)	Update Block Rates	putBlockRates	BLK
Allotment (Group Blocks)	Update Block Pickup	putBlockStatusToOpen	BLK
Allotment (Group Blocks)	Update Catering Status	putCateringStatus	BLK
Allotment (Group Blocks)	Cancel Block	postCancelBlock	BLK
Hurdles	Hurdle	Block Asynchronous, Inventory Asynchronous, Par Asynchronous & Reservation Asynchronous Modules	ASYNC
Inventory	Inventory - KATOVER (overbook)	changeSellLimitByDate	INV
Inventory	Inventory - OOO	postOutOfOrderRooms	HSK
Stay Records		getStayHistory	CRM

Table 15-1 (Cont.) OXI to REST

Functional Module	OXI Inbound Operation (External System to OPERA Cloud)	REST API Operation	REST Module
Packages	Packages - New	postPackage	RTP
Packages	Packages - Edit	putPackage	RTP
Rates	Rate - Header	postRatePlan	RTP
Rates		postNegotiatedRates	RTP
Rates		postRatePlanPackages	RTP
Rates	Rate - Detail (Rate Set)	postRatePlanSchedules	RTP
Rates	Rate Strategy	postRateStrategy	RTP
RAVL (Rate Restrictions)	Restriction - Change	postRestriction	PAR
RAVL (Rate Restrictions)	Restriction - Change	clearAllRestrictions	PAR
RAVR (Rate/ Room Type Restrictions)	RAVR - New	postRestriction	PAR
RTAV (Inventory Snapshot)	RTAV	getHotelInventory	INV
Configuration	Configuration - RATE_CATEGORY	createRateCategory	ENTCFG
Configuration	New Item Inventory	postInventoryItems	EVMCFG
Configuration	New Item Load	postItemPools	EVMCFG
Configuration	Update Item Inventory	putInventoryItems	EVMCFG
Configuration	New Item Inventory Class	postItemClasses	EVMCFG

HTNG

The legacy HTNG integration was most commonly used for Hoteliers and partners with activity management applications, such as spa and golf booking applications. It used the OPERA Electronic Distribution Systems (OEDS) integration and provided web services utilizing SOAP calls. The HTNG interface with OPERA used a combination of synchronous SOAP requests from the external system to OPERA as well as Business Events from OPERA to the external system. The Business Event functionality is still available when migrating to the REST APIs. Although the messages look a little different, the available data is still in REST format. For further information on Business Event functionality, refer to the Business Events topic.

The legacy HTNG interface also provided the ability for posting financial transactions to OPERA.

The below mapping table provides the suggested APIs to use when migrating from the legacy HTNG interface to the REST APIs available in OHIP.

Table 15-2 HTNG to REST

HTNG Functionality	HTNG Operation	REST API Operation	REST Module
Lookup Profile Information	ProfileLookup	getProfiles	CRM
Retrieve Profile information	FetchProfile	getProfiles	CRM
Send a Profile Update	UpdateProfile	putProfile	CRM
Establish a link between system	Subscription	putProfile	CRM
Send a New Profile	NewProfile	postProfile	CRM
Merge Profiles	MergeProfile	mergeProfiles	CRM
Send a Delete Profile	DeleteProfile	deleteProfiles	CRM
Lookup Reservation Information	ReservationLookup	getReservations	RSV
Retrieve Reservation Information	FetchReservation	getReservation	RSV
Send Reservation Updates	GuestStatusNotification	getBusinessEvents	INT
Send Messages for Guest	GuestMessage	postGuestMessages	RSV
Send Location information	LocationNotification	postReservationLocators	RSV
Send a New Activity	CreateActivity	postActivityBooking	LMS
Update an Activity	UpdateActivity	putActivityBooking	LMS
Cancel an Activity	CancelActivity	deleteActivityBooking	LMS
Lookup Activities	ActivityLookup	getActivityBooking	LMS
Post Charges to guest folio	PostPayment	postBillingCharges	CSH
Retrieve Extended Reservation Information	FetchReservationExt	getReservation	RSV

Table 15-2 (Cont.) HTNG to REST

HTNG Functionality	HTNG Operation	REST API Operation	REST Module
Send Extended Reservation Updates	GuestStatusNotificationExt	getBusinessEvents	INT
Retrieve Booked Packages from reservation	FetchBookedPackages	getReservation	RSV
Retrieve Package Details from configuration	FetchProductItems	getReservationPackagesLOV	LOV
Retrieve List of values for a field	FetchLOV	getLov	LOV
Send a check-in reservation	CheckIn	postCheckIn	FOF
Send Queue Room Information	QueueRoomBE	getBusinessEvents	INT
Send Room Status update	RoomStatusUpdateBE	getBusinessEvents	INT
Send Room Status update	UpdateRoomStatus	putRoomRelatedStatus	HSK
Retrieve Housekeeping Task Sheets	FetchHousekeepingTask	getHousekeepngTasks	HSK
Retrieve Room Status Information	FetchRoomStatus	getHousekeepingOverview	HSK
Activity Updates with reservation change	OutofScopeNotification	Not available in OPERA Cloud	Not available in OPERA Cloud
View Check information on guest Folio	FetchCheckDetails	Not available in OPERA Cloud	Not available in OPERA Cloud
OPERA Cloud to Retrieve Activities for Guest	FetchActivities	Not available in OPERA Cloud	Not available in OPERA Cloud

OPERA Web Services (OWS) and KIOSK

The legacy OWS and Kiosk interfaces use SOAP Web Services technologies to support data transfer from a client application to OPERA. The interface is commonly used for web booking engines and check-in applications, but it is also used for customer relationship management and membership functionality. While the legacy interfaces provide a lot of functionality,

migrating to the new REST API provides even more functionality for your integration to OPERA Cloud. Furthermore, the REST APIs also have an asynchronous interface just like OWS.

The below table provides suggested APIs and operations to use when migrating from legacy OWS/KIOSK interface to the REST APIs. However, there are a few operations currently not available in REST that are available in OWS. Management of membership functionality is not yet available.

Table 15-3 OWS to REST

OWS Module	OWS Operation	REST Operation	REST Module
Availability	Availability	getHotelAvailability	PAR
Availability	FetchAvailableItems	getItemInventory	INV
Availability	FetchAvailablePackages	getPackages	RTP
Availability	FetchBlockInventory	getBlock	BLK
Availability	FetchCalendar	getRestrictionsByDateRange getInventory	PAR INV
Availability	FetchExpectedCharges	getRateInfo	REV
Availability	FetchItemGroups	getItemInventory	INV
Availability	RegionalAvailability	getHotels getHotelAvailability	PAR
Availability	RegionalAvailabilityExt	getHotels getHotelAvailability	PAR
GuestServices	UpdateRoomStatus	putRoomRelatedStatus	HSK
GuestServices	WakeUpCall	getWakeUpcalls	HSK
HouseKeeping	ChangeRoomMaintenance	putRoomMaintenance	HSK
HouseKeeping	CreateRoomMaintenance	postRoomMaintenance	HSK
HouseKeeping	DeleteRoomMaintenance	deleteRoomMaintenance	HSK
HouseKeeping	FetchHouseKeepingDiscrepancies	getHousekeepingDiscrepancies	HSK
HouseKeeping	FetchHouseKeepingRoomStatus	getHousekeepingOverview	HSK
HouseKeeping	FetchHouseKeepingRoomTaskStatus	getHouseKeepingTasks	HSK
HouseKeeping	FetchHouseKeepingStatistics	getFrontOfficeStatistics	FOF
HouseKeeping	FetchHouseStatus	getFrontOfficeStatistics	FOF
HouseKeeping	FetchOOSRooms	getOutOfServiceRooms	HSK
HouseKeeping	FetchRoomMaintenance	getRoomMaintenance	HSK
HouseKeeping	ResolveRoomMaintenance	putRoomMaintenance	HSK

Table 15-3 (Cont.) OWS to REST

OWS Module	OWS Operation	REST Operation	REST Module
HouseKeeping	UnResolveRoomMaintenance	putRoomMaintenance	HSK
HouseKeeping	UpdateHouseKeepingDiscrepancies	putRoomRelatedStatus	HSK
HouseKeeping	UpdateHouseKeepingRoomStatus	putRoomRelatedStatus	HSK
Information	CurrencyConverter	calculateForeignCurrency	CSH
Information	QueryAwardsSchedules	getPromotioncodes	RTP
Information	QueryChainInformation	getChain	ENTConfig
Information	QueryHotelInformation	getHotelDetails	ENTConfig
Information	QueryLov	getLov	LOV
Information	QueryPackageItems	getPackages	RTP
Information	QueryRate	getAvailabilityRatePlanInfoByMultipleRatePlans	PAR
MeetingRoom	CreateBlock	postBlocks	BLK
MeetingRoom	MeetingAvailability	getEventCalendarMultipleHotelIds	EVM
MeetingRoom	MeetingCreateEvent	postEvents	EVM
MeetingRoom	MeetingFetchEvent	getEvent	EVM
MeetingRoom	MeetingFetchMenu	getEventResourceByMenu	EVM
MeetingRoom	MeetingFetchMenuItem	getCateringMenuItems	EVM
MeetingRoom	MeetingFetchMiscellaneousItem	getEventResourceByMenu	EVM
MeetingRoom	MeetingFetchPackageEvent	getCateringPackages	EVM
MeetingRoom	MeetingModifyEvent	putEvents	EVM
MeetingRoom	MeetingModifyPackageEvent	putCateringPackages	EVM
MeetingRoom	MeetingMultiPropertyAvailability	getFunctionSpaceAvailability	EVM
MeetingRoom	MeetingPackageAvailability	getCateringPackages	EVM
MeetingRoom	MeetingRoomCopyBlock	putCateringCopys	EVM
MeetingRoom	MeetingRoomCreateRelationship	putEvent	EVM

Table 15-3 (Cont.) OWS to REST

OWS Module	OWS Operation	REST Operation	REST Module
MeetingRoom	MeetingRoomFetchBlock	getEvent	EVM
MeetingRoom	ModifyBlock	putBlocks	EVM
Membership	AddPromoSubscription	putProfile	CRM
Membership	CreateEnrollment	postMemberships	CRM
Membership	DeletePromoSubscription	deleteAward	CRMConfig
Membership	FetchAvailableECertificates	getMembershipIssueAwardsList	CRM
Membership	FetchBenefits	getFlexibleBenefitAwards	CRM
Membership	FetchECertificates	getAwards	RSV
Membership	FetchEnrollmentCode	fetchMembershipEnrollmentCodes	CRMConfig
Membership	FetchMemberPoints	getMembershipAwardPointsByHotel	CRM
Membership	FetchMembershipTransactions	getMembershipTransaction	CRM
Membership	FetchNextCardNumber	postMembershipNumber	CRM
Membership	FetchProductAwards	getAvailableAwardsBasedOnType	CRM
Membership	FetchPromoSubscriptions	getAwardsToGrant	CRM
Membership	FetchRateAwards	getAvailableAwardsBasedOnType	CRM
Membership	FetchTransactionAwards	getAvailableAwardsBasedOnType	CRM
Membership	FetchUpgradeAwards	getAwards	RSV
Membership	IssueTransactionAward	postAwards	RSV
Membership	ModifyEnrollment	putMemberships	CRM
Membership	ReIssueMemberCard	postMembershipNumber	CRM
Membership	TransferPoints	transferMembershipPoints	CRM
Membership	UpdateEnrollmentCode	putMemberships	CRM
Name	DeleteAddress	putprofile	CRM
Name	DeleteComment	putprofile	CRM
Name	DeleteEmail	putprofile	CRM
Name	DeleteGuestCard	deleteMembership	CRM

Table 15-3 (Cont.) OWS to REST

OWS Module	OWS Operation	REST Operation	REST Module
Name	DeletePassport	putprofile	CRM
Name	DeletePhone	putprofile	CRM
Name	DeletePreference	deletepreferences	CRM
Name	DeletePrivacyOption	putprofile	CRM
Name	FetchAddressList	putprofile	CRM
Name	FetchClaimsStatus	getMembershipClaim	CRM
Name	FetchCommentList	getProfile	CRM
Name	FetchDocumentList	getProfile	CRM
Name	FetchEmailList	getProfile	CRM
Name	FetchGuestCardList	getProfile	CRM
Name	FetchName	getProfile	CRM
Name	FetchNameUDFs	getProfile	CRM
Name	FetchPhoneList	getProfile	CRM
Name	FetchPreferenceList	getPreferenceForProfile	CRM
Name	FetchPrivacyOption	getProfile	CRM
Name	FetchProfile	getProfile	CRM
Name	FetchProfileBenefits	getFlexibleBenefitAwards	CRM
Name	FetchSubscription	getProfile	CRM
Name	ForgetProfile	deleteProfile	CRM
Name	GetPassport	getProfile	CRM
Name	InsertAddress	postProfile	CRM
Name	InsertClaim	putProfile	CRM
Name	InsertComment	putProfile	CRM
Name	InsertEmail	putProfile	CRM
Name	InsertGuestCard	postMembership	CRM
Name	InsertPhone	putProfile	CRM
Name	InsertPreference	postPreferences	CRM
Name	InsertUpdateNameUDFs	putProfile	CRM
Name	InsertUpdatePrivacyOption	putProfile	CRM
Name	NameLookup	getProfiles	CRM
Name	RegisterName	postProfile	CRM
Name	TravelAgentLookup	getProfiles	CRM
Name	UpdateAddress	putProfile	CRM
Name	UpdateClaim	putProfile	CRM
Name	UpdateComment	putProfile	CRM

Table 15-3 (Cont.) OWS to REST

OWS Module	OWS Operation	REST Operation	REST Module
Name	InsertUpdateDocument	uploadFileAttachment	CRM
Name	UpdateEmail	putProfile	CRM
Name	UpdateGuestCard	putMemberships	CRM
Name	UpdateName	putProfile	CRM
Name	UpdatePassport	putProfile	CRM
Name	UpdatePhone	putProfile	CRM
Name	ValidateForgetProfile	putvalidateForgetProfiles	CRM
Reservation	AddAccompanyGuest	putReservation	RSV
Reservation	AssignRoom	postRoomAssignment	FOF
Reservation	BookHoldItems	putReservation	RSV
Reservation	BreakShare	deleteShareReservation	RSV
Reservation	CancelBooking	postCancelReservations	RSV
Reservation	ClearItemHold	putItemInventoryHold	INV
Reservation	CombineShare	Postcombinesharesreservations	RSV
Reservation	ConfirmBooking	putReservation	RSV
Reservation	CreateBooking	postReservation	RSV
Reservation	CreateItemHold	postHoldItemInventory	RSV
Reservation	CreateMultipleBookings	postReservation	RSV
Reservation	DeleteAccompanyGuest	putReservation	RSV
Reservation	DeleteInventoryItem	putReservation	RSV
Reservation	DeletePackages	putReservation	RSV
Reservation	FetchAvailableOffers	getReservationUpsellInfo	RSV
Reservation	FetchBookedInventoryItems	getReservation	RSV
Reservation	FetchBookedPackages	getReservation	RSV
Reservation	FetchBooking	getReservation	RSV
Reservation	FetchBookingForPointUpdate	getReservation	RSV
Reservation	FetchHoldItems	getHoldItemsInventory	INV
Reservation	FetchRoomUpgrades	getReservationUpsellInfo	RSV
Reservation	FetchSummary	getReservation	RSV

Table 15-3 (Cont.) OWS to REST

OWS Module	OWS Operation	REST Operation	REST Module
Reservation	FutureBookingSummary	getReservations getProfile	RSV CRM
Reservation	GetReservationStatus	getReservation	RSV
Reservation	GuestRequests	getServiceRequest	RSV
Reservation	MergeReservations	putMergeReservations	RSV
Reservation	ModifyBooking	putReservation	RSV
Reservation	ModifyItemHold	putHoldItemsInventory	INV
Reservation	PreCheckin	postPreCheckin	RSV
Reservation	ReInstateReservation	putReservations	RSV
Reservation	ReleaseRoom	putRoomAssignment	FOF
Reservation	RoomMove	putMoveInHouseGuest	FOF
Reservation	UpdateInventoryItem	putReservation	RSV
Reservation	UpdatePackages	putReservation	RSV
Reservation	UpgradeReservation	putReservation	RSV
Reservation	UpsellReservation	postUpsellReservation	RSV
ResvAdvanced	AddPayment	getCreditCardUniqueId	FOF
ResvAdvanced	AdditionalKeys	postRoomKeys	FOF
ResvAdvanced	AlternateRooms	putVerifyCheckinReservation	FOF
ResvAdvanced	AssignRoom	postRoomAssignment	FOF
ResvAdvanced	CancelCheckIn	deleteCheckin	FOF
ResvAdvanced	CheckIn	postCheckin	FOF
ResvAdvanced	CheckOut	postCheckout	CSH
ResvAdvanced	CreateFixedCharges	postFixedCharges	CSH
ResvAdvanced	CreateGuestLocator	postReservationLocators	RSV
ResvAdvanced	DeleteFixedCharges	deleteFixedCharges	CSH
ResvAdvanced	DeleteGuestLocator	deleteReservationlocators	RSV
ResvAdvanced	DeletePayRouting	deleteRoutingInstructions	RSV
ResvAdvanced	ExternalPayment	postBillingPayments	CSH
ResvAdvanced	FetchAuthorizationsHistory	getAuthorizationHistory	CSH
ResvAdvanced	FetchFixedCharges	getFixedCharges	CSH
ResvAdvanced	FetchKeyData	getRoomKey	FOF
ResvAdvanced	FetchPromotionCode	putReservation	RSV
ResvAdvanced	FetchQueueReservations	getQueuedReservations	FOF

Table 15-3 (Cont.) OWS to REST

OWS Module	OWS Operation	REST Operation	REST Module
ResvAdvanced	FetchResPromotionCode	getReservation	RSV
ResvAdvanced	FetchRoomSetup	gethousekeepinvoverview	HSK
ResvAdvanced	FetchRoomStatus	getHotelRooms	HSK
ResvAdvanced	FolioTransactionTransfer	putTransferTransactionToReservation	CSH
ResvAdvanced	GuestMessages	getReservation	RSV
ResvAdvanced	InsertPayRouting	postRoutingInstruciton s	RSV
ResvAdvanced	Invoice	getFolios	CSH
ResvAdvanced	KioskAvaliablity	getHotelAvailability	PAR
ResvAdvanced	MakePayment	postBillingpayments	CSH
ResvAdvanced	PayRouting	getRoutinginstructions	RSV
ResvAdvanced	PostCharge	postBiilingCharges	CSH
ResvAdvanced	PrintPreCheckOutBill	postDepositFolio	CSH
ResvAdvanced	QueueReservation	getQueuedReservation	FOF
ResvAdvanced	ReleaseRoom	putRoomAssignment	FOF
ResvAdvanced	ReservationRequestCode	getRequestCodes	V0
ResvAdvanced	SetResPromotionCode	putReservation	RSV
ResvAdvanced	ToggleTurndownFlag	postTurnawayReservation	RSV
ResvAdvanced	UpdateFixedCharges	putFixedcharges	RSV
ResvAdvanced	UpdateGuestLocator	putLocators	RSV
ResvAdvanced	UpdateMethodOfPayment	putReservation	RSV
ResvAdvanced	UpdatePayRouting	putRoutingInstrucitons	RSV
StayHistory	StayHistory	getStayhistory	CRM
HouseKeeping	DeleteRoomMaintenanceImages	Not available in OPERA Cloud	Not available in OPERA Cloud
HouseKeeping	FetchRoomMaintenanceImages	Not available in OPERA Cloud	Not available in OPERA Cloud
MeetingRoom	MeetingCreatePackageEvent	Not available in OPERA Cloud	Not available in OPERA Cloud
Name	DeleteDocument	Not available in OPERA Cloud	Not available in OPERA Cloud
Reservation	EmailConfirmation	Not available in OPERA Cloud	Not available in OPERA Cloud
ResvAdvanced	InsertSignedRegCard	Not available in OPERA Cloud	Not available in OPERA Cloud

Table 15-3 (Cont.) OWS to REST

OWS Module	OWS Operation	REST Operation	REST Module
ResvAdvanced	GenerateRegistrationCard	Not available in OPERA Cloud	Not available in OPERA Cloud
ResvAdvanced	FetchSignedRegCard	Not available in OPERA Cloud	Not available in OPERA Cloud
Availability	GdsAreaAvailability	Not available in OPERA Cloud	Not available in OPERA Cloud
Availability	GetCacheStatus	Not available in OPERA Cloud	Not available in OPERA Cloud
Brochure	SendBrochure	Not available in OPERA Cloud	Not available in OPERA Cloud
GuestServices	FetchOptInSetup	Not available in OPERA Cloud	Not available in OPERA Cloud
GuestServices	UpdateReservationForOptIn	Not available in OPERA Cloud	Not available in OPERA Cloud
Information	GetScreenItems	Not available in OPERA Cloud	Not available in OPERA Cloud
MeetingRoom	MeetingFetchBlockDelegates	Not available in OPERA Cloud	Not available in OPERA Cloud
MeetingRoom	MeetingFetchContract	Not available in OPERA Cloud	Not available in OPERA Cloud
MeetingRoom	MeetingFetchMyRegisteredEvents	Not available in OPERA Cloud	Not available in OPERA Cloud
MeetingRoom	MeetingRegisterEventAttendees	Not available in OPERA Cloud	Not available in OPERA Cloud
MeetingRoom	MeetingRoomFetchMyBlocks	Not available in OPERA Cloud	Not available in OPERA Cloud
Membership	AddBenefit	Not available in OPERA Cloud	Not available in OPERA Cloud
Membership	CancelConsumedPoints	Not available in OPERA Cloud	Not available in OPERA Cloud
Membership	CancelECertificate	Not available in OPERA Cloud	Not available in OPERA Cloud
Membership	ConsumeECertificate	Not available in OPERA Cloud	Not available in OPERA Cloud
Membership	ConsumePoints	Not available in OPERA Cloud	Not available in OPERA Cloud
Membership	ConsumePointsOthers	Not available in OPERA Cloud	Not available in OPERA Cloud
Membership	FavoriteGuest	Not available in OPERA Cloud	Not available in OPERA Cloud
Membership	FetchEcertificateTierPoints	Not available in OPERA Cloud	Not available in OPERA Cloud
Membership	FetchFeeHistory	Not available in OPERA Cloud	Not available in OPERA Cloud

Table 15-3 (Cont.) OWS to REST

OWS Module	OWS Operation	REST Operation	REST Module
Membership	FetchMemberTierWizard	Not available in OPERA Cloud	Not available in OPERA Cloud
Membership	FetchPointsExchange	Not available in OPERA Cloud	Not available in OPERA Cloud
Membership	FetchStatement	Not available in OPERA Cloud	Not available in OPERA Cloud
Membership	FetchStatementRefs	Not available in OPERA Cloud	Not available in OPERA Cloud
Membership	IssueECertificate	Not available in OPERA Cloud	Not available in OPERA Cloud
Name	AssociateBooker	Not available in OPERA Cloud	Not available in OPERA Cloud
Reservation	SetDailyPoints	Not available in OPERA Cloud	Not available in OPERA Cloud
Reservation	ConsumeVouchers	Not available in OPERA Cloud	Not available in OPERA Cloud
ResvAdvanced	SetKeyData	Not available in OPERA Cloud	Not available in OPERA Cloud
ResvAdvanced	InstantPaymentNotification	Not available in OPERA Cloud	Not available in OPERA Cloud
UnitOwners	AddProfileToContract	Not available in OPERA Cloud	Not available in OPERA Cloud
UnitOwners	CreateContract	Not available in OPERA Cloud	Not available in OPERA Cloud
UnitOwners	DeleteProfileFromContract	Not available in OPERA Cloud	Not available in OPERA Cloud
UnitOwners	FetchAuthorizedProfiles	Not available in OPERA Cloud	Not available in OPERA Cloud
UnitOwners	FetchContract	Not available in OPERA Cloud	Not available in OPERA Cloud
UnitOwners	FetchContractDetails	Not available in OPERA Cloud	Not available in OPERA Cloud
UnitOwners	OwnedUnitAvailability	Not available in OPERA Cloud	Not available in OPERA Cloud
UnitOwners	OwnerStatementDetails	Not available in OPERA Cloud	Not available in OPERA Cloud
UnitOwners	UpdateContract	Not available in OPERA Cloud	Not available in OPERA Cloud
UnitOwners	UpdateProfileInContract	Not available in OPERA Cloud	Not available in OPERA Cloud
Reservation	IgnoreBooking	Not available in OPERA Cloud	Not available in OPERA Cloud
Name	UpdateCreditCard	Not available in OPERA Cloud	Not available in OPERA Cloud

Table 15-3 (Cont.) OWS to REST

OWS Module	OWS Operation	REST Operation	REST Module
Name	InsertCreditCard	Not available in OPERA Cloud	Not available in OPERA Cloud
Name	FetchCreditCardList	Not available in OPERA Cloud	Not available in OPERA Cloud
ResvAdvanced	DCCCheck	Not available in OPERA Cloud	Not available in OPERA Cloud
Name	DeleteCreditCard	Not available in OPERA Cloud	Not available in OPERA Cloud

Property Interfaces — IFC8, FIAS and XML_POS

The Oracle OPERA Hotel Property Interface application (IFC8) is used to connect various on-premise vendor management systems with the Oracle Property Management system. It sends out room and guest data to the vendor system, such as, for example, check-in of guest and check-out of guest. The interface can also receive data (for example, charges or status changes) from the external system to be stored in OPERA using synchronous TCP/IP or serial connection exchanging messages, which are based on either Oracle's universal FIAS API, XML-POS API, or many vendor-based specifications.

Some examples of vendor systems connecting via the Property Interface IFC8 application are the following:

- Telephone Management Systems (TMS) or middleware to TMS systems
- Door Locking Systems
- HSIA Internet Access Systems
- Point of Sale Systems (Restaurant, SPA, Golf)
- Video Services Systems
- Guest Service systems
- Minibar systems
- Building Management systems
- Miscellaneous / Other systems installed at property

The typical data exchanged via the Property Interface are check-in / check-out notifications (for example, room and guest details, guest rights), charge postings received from vendor systems, make door key requests, wake up requests, credit card payment requests, guest message notifications, room maid status notifications, and many more. This data is exchanged at the time the action occurs in OPERA or on the external system and is usually exchanged without a delay in transmission.

For IFC8 actions exchanged between OPERA Cloud and an external application, the REST APIs utilize multiple ways of connection.

There are inbound actions from the external system that will be exposed through the OHIP REST APIs. Some examples of these inbound actions are sending charges to the guest's room or hotel account, inquiry requests for Guest information by a Point-Of-Sale system, sending room status changes, wake up requests, and guest bill balance requests.

Some outbound IFC8 actions will be exposed using streaming APIs. Some examples include check-in, check-out, room-move notifications, wake-up requests, and guest text messages.

Specific outbound synchronous messages will be exposed using additional outbound service connections from the PMS to the external system. Some examples are door lock key actions (for example, make key and delete key), which require related response messages.

The below table provides suggested APIs and operations to use when migrating from legacy FIAS/XML-POS IFC8 interface to the REST APIs. There are a few operations currently not available in REST that are available via IFC8.

> **Note:**
>
> Posting charges via room number only (instead of reservationId) is not yet available with REST.

Table 15-4 Suggested APIs and Operations

IFC8 Module	IFC8 Operation	REST Module	OHIP REST API	OPERA Cloud Business Event (Push or Pull)
Enhanced Posting	PostInquiry - by Room number, Name, Track2 Post List	RSV	getReservations	
Extended Guest Data	GuestMessage-Request, GuestMessage-Text	RSV	guestMessages	
Extended Guest Data	GuestMessage Text-online GuestMessage-retrieved GuestMessage-Delete	Business Event		NEW GUEST MESSAGE UPDATE GUEST MESSAGE DELETE GUEST MESSAGE
Room Data	Room Equipment: Voice mail notification from vendor to PMS	RSV	guestMessages	
Simple Posting	PostSimple: Phone charge, Minibar charge, Video charge using room number only PostAnswer	CSH	postBillingCharges	

Table 15-4 (Cont.) Suggested APIs and Operations

IFC8 Module	IFC8 Operation	REST Module	OHIP REST API	OPERA Cloud Business Event (Push or Pull)
Enhanced Posting	PostRequest: Point-of-Sale, SPA, Golf charges using Reservation Number PostAnswer	CSH	postBillingCharges	
Room Data	RoomEquipment: send Room Maid Status from vendor to PMS	HSK	SetRoomStatus	
Room Data	RoomEquipment: send Room Maid status from PMS to vendor	Business Event		UPDATE ROOM STATUS
Room Data	RoomEquipment: send Class Of Service, TVRights, MinibarRights, Do not Disturb (Phone) from PMS to vendor	Business Event as of OPERA 23.4		UPDATE INTERFACE STATUS
Room Data	RoomEquipment: send MessageLight ON/OFF as message notification from PMS to vendor	Business Event as of OPERA 23.4		UPDATE INTERFACE STATUS
Wake-up Call	Wakeup Request Wakeup Clear Wakeup Answer from vendor to PMS	FOF	postwakeUpCalls delete wakeUpCalls put wakeUpCalls	
Wake-up Call	Wakeup Request Wakeup Clear Wakeup Answer from PMS to vendor	Business Event as of OPERA 23.4		available soon
Guest Data	GuestIn notification from PMS GuestOut notification from PMS GuestdataChange notification from PMS, Room Move notification	Business Event		CHECK IN CHECK OUT UPDATE RESERVATION

Table 15-4 (Cont.) Suggested APIs and Operations

IFC8 Module	IFC8 Operation	REST Module	OHIP REST API	OPERA Cloud Business Event (Push or Pull)
Room Data	RoomEquipment: set Guest Service Status (Make Up Room, Do Not Disturb) from vendor to PMS	HSK	serviceRequests	
Room Data	RoomEquipment: receive Guest Service Status change (Make Up Room, Do Not Disturb) from PMS to vendor	Business Event as of OPERA 23.4		GUEST SERVICE STATUS REQUEST
Guest Locator	Locator On, Locator Off, Retrieve Locator	RSV	guestLocators	
Extended Guest Data	Guest Bill Request / Guest Bill Items / Guest Bill Balance	CSH	getGuestsTransactions or getFolios	
Extended Guest Data	Remote check-out Request (Video Check-out)	CSH	postBillingPayments + postFolios + postCheckout	
Night Audit	Night Audit Start notification Night Audit End notification	Business Event		CLOSE BUSINESS DATE ROLL BUSINESS DATE
Key Services	Key Request (New Key, Duplicate Key, One Shot Key) Key Delete Key Data Change (Room move, change check out date) - Online Key card systems Key Read	FOF Outbound as of OPERA 23.4	post externalRoomKeys OPERA Cloud Outbound Syncronous	

Table 15-4 (Cont.) Suggested APIs and Operations

IFC8 Module	IFC8 Operation	REST Module	OHIP REST API	OPERA Cloud Business Event (Push or Pull)
Virtual Numbers (DID)	GuestIn notification with Virtual Number/ Equipment Number, Status, Pool Id GuestdataChange notification with Virtual Number/ Equipment Number, Status, Pool Id GuestOut notification with Virtual Number/ Equipment Number, Status, Pool Id	Feature is not available with OPERA Cloud	N/A	N/A

Sample Messages

Along with the mappings provided with legacy to REST APIs, there are a number of Postman Workflows put together to aid in the development of integrations. The Property Workflow collection navigates and suggests operations to perform for a common workflow. For example, how to perform a check-in or the steps required to search availability and make a new Reservation. There is another collection for Property APIs by Module. This collection has over 2200 sample messages for the operations in each Property API, including Reservation, Blocks, Profiles, and Asynchronous APIs. The collection for Distribution APIs by Module contains sample messages for the Distribution APIs, such as Shop, Book, and Reservation. You can find these collections in gitHub or in Postman.com.

16
Anti-Patterns

An anti-pattern is a software design practice that is ineffective or counterproductive—in other words, the opposite of a "best practice." To put it another way, an anti-pattern is something that the software allows you to do, but that may have an adverse functional or performance impact.

Table 16-1 Anti-Patterns

Anti-Pattern	Category	Rationale	Recommendation
Using REST APIs to extract bulk data. For example, Folios, Profiles, often specifying a large "limit" query parameter or a large date range.	Functional	Faster access with less operational impact on the hotel	The Oracle Hospitality APIs accommodate many use cases but were not designed for bulk data extract. It is recommended to use either an extract from Oracle Reporting and Analytics for bulk data use cases.
Multi-property shop against property availability API	Functional	Faster access with less operational impact on the hotel	The Oracle Hospitality OPERA APIs are optimized for resort-level resources. While it is possible to shop for availability across many different properties using the Oracle Hospitality OPERA APIs, it is recommended to use the Shop API which is specially designed for this purpose and returns data from a live cache. For example, I want to be able to create a reservation for a guest at Hotel1, and then for the same guest book another reservation at Hotel2. The Shop API has the functionality available for you to look at availability across multiple properties.

Table 16-1 (Cont.) Anti-Patterns

Anti-Pattern	Category	Rationale	Recommendation
Implementing multi-threaded Java application calling streaming API	Technical	Functional and scalable	The Streaming API adheres to the graphql-ws protocol which requires that a given stream receives a connection from only one source to preserve the ordering of events. A stream is identified as a combination of the following: gateway + the chainCode + an application key. For an example, refer to the Spring Boot documentation **Tip**: Use a single WebSocketGraphQlClient instance for each server to have a single, shared connection for all requests to that server. Each client instance establishes its own connection, which is typically not the intent for a single server. While we recommend using our GraphiQL tool or using Postman to solidify understanding of the Streaming API, it is important to use separate applications for GraphiQL, Postman, and your client code. Similarly, your client code must be single threaded. Our recommendation is to use a single-threaded application to consume the stream, then deploy a tool like Apache Kafka and multithreaded clients to consume events from Apache Kafka to populate back end systems.

Table 16-1 (Cont.) Anti-Patterns

Anti-Pattern	Category	Rationale	Recommendation
Sending a token request with every API call	Functional	More cost effective for partners, protects identity servers Performance impact and operational impact as a result of rate limiting by the identity servers	oAuth tokens have a lifetime of 60 minutes, and requesting an oAuth token is billable to integration partners. Our recommendation is to request a token only once every 59 minutes and implement code that caches and automatically renews the token every 59 minutes and stores it securely. Code that makes API calls can then use the cached token and be assured it is always valid.

Table 16-1 (Cont.) Anti-Patterns

Anti-Pattern	Category	Rationale	Recommendation
postReservation, putReservation, postProfile and putProfile with invalid codes, such as rate, room type, source and market codes, address type, membershipType, and so on.	Functional	Corrupt data	OPERA Cloud is highly configurable and most of our OPERA Cloud customers leverage this to create the unique experiences offered by their resort. This means that many configuration items will differ from one resort to another. When creating or modifying reservations in OPERA Cloud, code valid to the resort must be used in the request body. For example, Market Code or Source Code. Failure to use codes valid for the property will result in reservations being created in OPERA Cloud with invalid codes. As soon as a user views the reservation, the user must update the reservation with valid codes. To avoid this, integrators should use the List of Values Management and Enterprise Configuration APIs to determine the configuration particular to the resort they are calling. An example is postReservation. When creating a new reservation, there are codes required as part of the request body. Prior to posting the reservation, ensure you have called the List Of Values, such as getSourceCode, getMarketCodes, getGuaranteeCodes, getMembershipTypes, and so on. The postman workflow samples we have offer an integrator a suggested set of operations to call prior to posting the new reservation. Please take a

Table 16-1 (Cont.) Anti-Patterns

Anti-Pattern	Category	Rationale	Recommendation
			look at this to start building your integration.
Writing straight to back end databases after receiving an event	Technical	Overloads back end databases	OPERA Cloud can generate many thousands of events in certain circumstances, and the Streaming API is not throttled. If the streaming client is coded to write straight to a back end database, this can overwhelm the back end database. To avoid this, implement a buffering mechanism such that events are consumed from the buffer before being written to the back end database. Ensure that the code reading from the buffer can scale to accommodate large numbers of events without overwhelming the back end database.
Call APIs directly from browsers or mobile apps	Technical	Security of credentials and data	The Oracle Hospitality Integration Platform APIs are certified only to be called from back end systems. This is partly a security posture and partly that the APIs are not optimized for mobile data restrictions. If Oracle Hospitality APIs are needed as part of a browser or mobile app-based experience, implement a "Backend for Frontend (BFF)" pattern, which creates an abstraction layer consuming OHIP APIs and provides Experience APIs that are better suited to be called from mobile apps or web browsers.

Table 16-1 (Cont.) Anti-Patterns

Anti-Pattern	Category	Rationale	Recommendation
Generating implementations tied to API specifications	Technical	Resilience against change	As business APIs, the Oracle Hospitality APIs contain a wealth of data. However, a given integration may need only some of that data. To protect the integration against API changes, it is recommended to implement the "Tolerant Reader" pattern such that consuming code looks only at the fields needed by the implementation. While we always support backwards compatibility for v1 APIs, it is our goal to also support this for v0 APIs. However, changes can occur and the Tolerant Reader pattern can reduce the impact radius of these API changes.
Mapping Experience APIs 1:1 to OHIP APIs	Technical	Chatty, network heavy clients	When writing Experience APIs for consumption by mobile apps or web browsers, the APIs provided by Experience APIs do not need to map 1:1 to Oracle Hospitality APIs. We recommend using "API Composition" to gather all the required information from multiple Oracle Hospitality APIs together with the "Backends for Frontends" pattern to orchestrate multiple OHIP API calls but expose as a single Experience API.

Table 16-1 (Cont.) Anti-Patterns

Anti-Pattern	Category	Rationale	Recommendation
Sending GET calls to either a collection or single resource repeatedly in case it has changed	Functional	Expensive for integration partners and not real time	Assuming Business Events are configured, every change that occurs within our Hospitality applications triggers a Business Event. Rather than continuously GETting a resource to see if it has changed, we provide the ability to consume Business Events as they occur. For Property APIs, see Business Events. For example, my integration requires housekeeping information to be kept in sync with OPERA Cloud. Rather than GETting the data using the HouseKeeping APIs, configure the business events for the housekeeping module. This will ensure events are generated and sent to the external system each time a resource is changed directly in OPERA Cloud. As soon as someone changes a room to Out of Order in OPERA, a business event will be generated for the external system.

Table 16-1 (Cont.) Anti-Patterns

Anti-Pattern	Category	Rationale	Recommendation
Placing consuming architecture far from the OHIP gateway	Technical	Creates a high latency and poor consumer experience. For the streaming API, this can result in a massive backlog of events that can never be consumed.	When implementing a Backends for Frontends (BFF) against a given OHIP API Gateway instance, verify that high levels of network latency do not exist between these two components. This is because latency can have a negative impact to users of the application connecting to OHIP. For example, fetching an OAuth token should take no longer than 100ms. For ultimate speed and low latency, consider implementing your BFF in the same region as the OHIP API Gateway and housing the OPERA Cloud instance(s) within the Oracle Cloud Infrastructure (OCI). Not only does this reduce latency, but it also increases security because the API traffic remains inside of Oracle Cloud.

Table 16-1 (Cont.) Anti-Patterns

Anti-Pattern	Category	Rationale	Recommendation
Brittle handling of errors	Technical	Incomplete orchestration, unexpected results, operational impact on the hotel	Errors tend to be short lived, so to help create a fault tolerant consumer and a safe Backends for Frontends (BFF), use the circuit breaker pattern to retry the same API call when receiving an error. This is particularly important when orchestrating multiple OHIP API calls. When the resource is very large, the retry logic must be the following: • Retry the same call 30 minutes later • If the retry also times out, move the call to an error hospital and create a technical Service Request with all the details by following the process described in the Preface.

Table 16-1 (Cont.) Anti-Patterns

Anti-Pattern	Category	Rationale	Recommendation
Specifying all fetch instructions on a resource	Functional	Slow performance	Many of the Oracle Hospitality OPERA Cloud APIs by default return only a subset of a resource or only the parent resource. For example, getProfile returns only the basic information about a person. Many of these APIs use a standard query parameter "fetchInstructions," which allows additional, often child, information on the resource to be returned. By tailoring which additional pieces of information is returned to your use cases, you can increase the performance of and reduce the response body size of your API calls. To achieve this, orchestrate using the "indicators" fetchInstruction that will show which child elements are filled, and then send a call listing only those child elements as fetchInstructions. For example: First call: /crm/v1/profiles?profilesIds={{profileId}}&fetchInstructions=Indicators Then based on the results Second call: /crm/v1/profiles?{{{profileId}}&fetchInstructions=Communication<additional fetchInstructions based upon the results of the first call>

Table 16-1 (Cont.) Anti-Patterns

Anti-Pattern	Category	Rationale	Recommendation
Not using the "summaryInfo" query parameter	Functional	Slow performance	If only the summary information of a resource is needed, not child elements, then use the "summaryInfo" query parameter. This will both increase the performance of and reduce the response body size of your API calls. For example: /roomTypes?summaryInfo=true
Using Business Events to publish yield updates out to other systems	Functional	Performance impact to OPERA Cloud and to all external systems of that OPERA Cloud	Yield systems supply OPERA Cloud with updated rates to ensure price per room is optimized. This requires sending a large amount of price adjustments, each of which triggers many business events. By default, changes made by one external system are not pushed to another external system. The "publisher" feature within OPERA Cloud enables this to be overridden. However, the "publisher" feature must not be enabled on external systems in OPERA that supply yield updates because this will flood other external systems with needless rate updates and impact the timeliness of sending unrelated business events to those external systems.

Table 16-1 (Cont.) Anti-Patterns

Anti-Pattern	Category	Rationale	Recommendation
As a revenue management system, using Synchronous REST APIs to fetch bulk data, or update bulk data.	Functional	Performance impact	Revenue partners should be using asynchronous APIs to perform actions that will take some time for the OPERA Cloud database to action. For example, fetching a year's worth of reservations in OPERA Cloud. The Property APIs have asynchronous operations that cater to these business use cases. For more information, see Business Use Case.
Using APIs beyond their stated scope	Functional	Functional	Ensure you understand the scope of APIs by referencing the API specifications and any Business Use Case articles in the Oracle Hospitality Integration Platform developer portal.

Table 16-1 (Cont.) Anti-Patterns

Anti-Pattern	Category	Rationale	Recommendation
Enabling business events for an external system in OPERA or in OHIP ahead of being ready to consume	Functional	Causes a large backlog of events, potentially choking the consumer. This can have an operational impact.	As soon as events are subscribed to an external system in OPERA — be it from the OPERA Cloud user interface or OHIP — the subscribed events will start to be enqueued. If not consumed, this will result in a very large queue, which is challenging for consuming systems to process. Further, if the external system sends responses back to OPERA Cloud as a result of events received and is slow to process the events, then operational impact can occur as the state of the data in OPERA Cloud would differ from the state of the data perceived by an external system that is running behind. To avoid this, when creating external systems in the OPERA Cloud user interface, configure the Business Events, but mark the external system inactive. Activate the external system only once the consuming architecture is ready. Similarly, subscribe to business events via the streaming API only when the consuming architecture is ready.

Table 16-1 (Cont.) Anti-Patterns

Anti-Pattern	Category	Rationale	Recommendation
Not staying connected to the streaming API	Functional	Causes a large backlog of events, potentially choking the consumer.	The Streaming API is built upon WebSocket, where the connection remain open. Barring network events, it is expected that WebSocket connections will remain open and connected permanently to the Streaming API, subject to disconnecting every 1 hour to refresh the oAuth token. Disconnecting and then reconnecting some time later risks a large backlog of events queuing up, which can be challenging for the consuming architecture to process.
Incorrectly disconnecting from the streaming API	Functional	Connection will not re-open	When disconnecting from the Streaming API, it is important to follow the protocol and send the "Complete" message. After closing this, the connection can be closed. However, consuming systems must wait 500ms before reconnecting. Consuming systems that do not follow this process will find the WebSocket does not re-open.
Not sending "ping" to the Streaming API	Functional	Connection closes	To keep the WebSocket connection, the consumer must send a non-billable "ping" every 15 seconds. For more information, see Keeping the Stream Open. If the consumer does not send a ping, then the connection will automatically be closed after 30 seconds.

Table 16-1 (Cont.) Anti-Patterns

Anti-Pattern	Category	Rationale	Recommendation
Disconnecting then reconnecting from the Streaming API starting from a given offset	Functional	Functional	Each business event sent on the streaming API is given a different offset number. While these appear to increment, a linear progression is not guaranteed. The offset number also changes if the consumer has been disconnected from the stream for over 24 hours. It is recommended to remain connected. When you disconnect (for example to obtain a new oAuth token), reconnect without specifying an offset in the subscribe message. OHIP will resume sending events starting from the next event.
Two developers using the same application for streaming API	Functional	Functional	The Streaming API adheres to the graphql-ws protocol which requires that a given stream receives a connection from only one source to preserve the ordering of events. A stream is identified as a combination of the following: gateway + the chainCode + an application key. If two developers are trying to use the same application to access the Streaming API, they will lock each other out. Instead, create one application per developer, then once development is complete, delete those development applications.

Table 16-1 (Cont.) Anti-Patterns

Anti-Pattern	Category	Rationale	Recommendation
Using any cashierId in payment APIs	Functional	Impairs auditing payments	The cashierId usually represents a person at the front desk, but since it is possible to make payment changes using APIs, it is important to tie back the change to the organization or user who made the change. When multiple organizations use the same cashierId, it hinders auditing payments. Contact the environment owner and ask the owner to allocate a cashierId to your organization. Only use this provided cashierId when calling payment APIs.

17 Accessibility

Accessibility involves making your application usable for differently abled persons such as low vision or blindness, deafness, or other physical limitations. This means creating applications that can be used without a mouse (keyboard only), used with a screen reader for blind or low-vision users, and used without reliance on sound, color, or animation and timing.

Keyboard Only Users

When you log in to the Oracle Hospitality Developer Portal and go to the APIs page, you can use the keyboard keys to bring the focus to the Early Adopter message for Early Adopter APIs.

1. On the Early Adopter APIs panel:
    a. Press the **Tab** key in any of the rows on the APIs page.
    b. Press the **Tab** key and select the highlighted text for the Early Adopter and then press **F6**. The content is highlighted.
    c. Press the **Tab** key to select the email address in the highlighted text. To send an email, press **Enter**. The selected email address opens in your default email client.
    d. Press the **Esc** key to let go of the pop-up text.
2. To copy an application key using keyboard keys:
    a. Press the **Tab** key to move to the list of applications while on the Applications page.
    b. Press the **Arrow** keys to move from one application to the next.
    c. Press **F2** to select an application.
    d. Press the **Tab** key to select **Copy** or to select **View Details**.
    e. Press **Enter** to
        - view the application details while on View Details.

          or
        - copy the application key to the clipboard while on Copy.
3. For users who can view Early Adopter APIs, tab to the early adopter API banner and use the below keyboard commands.

Table 17-1 Keyboard Commands for Early Adopter API Users

Target	Key	Action
Focus within Popup	Tab or Shift + Tab	Navigate the content of the popup. Close the open popup if there are no tab stops in the popup.

Table 17-1 (Cont.) Keyboard Commands for Early Adopter API Users

Target	Key	Action
	F6	Move focus to the launcher for a popup with modeless modality. Close the open popup if the modality is modal.
	Esc	Close the open popup.
Popup Launcher	F6	Move focus to the first tab stop within the open popup. If there is not a tab stop within the content, focus is established on the popup.
Side Filters	Press Tab once	Move focus to side filters.
Search bar	Press Tab again when focus is set on side filters	Move focus from side filters to search bar.
Category	Up and Down Arrows	When focus is set on side filters on any category, you can use the arrow keys to move up or down a category (for example, API Lifecycle, Methods).
Check boxes	F2	Press F2 to move within a category and access the check boxes.

4. Tab to the data visualization chart to navigate between the data points on the Analytics tab.

Table 17-2 Keyboard Commands for Data Visualization Chart

Key	Action
Tab	Move focus to the next element.
Shift + Tab	Move focus to the previous element.
Up Arrow	Move focus and selection to the previous data item.
Down Arrow	Move focus and selection to the next data item.
Left Arrow	Move focus and selection to the previous data item (on left).
Right Arrow	Move focus and selection to the next data item (on right).

18
Additional Resources

The resources below can help you familiarize yourself with the Oracle Hospitality Integration Platform, existing integrations to OPERA Cloud, hospitality industry terminology, and the application nomenclature associated with the Oracle Hospitality APIs.

OHIP Overview, Registration, and Marketplace

- Oracle Hospitality Integration Platform Product Page
- Oracle Hospitality Integration Platform Self Service Registration
- Oracle Hospitality Cloud Marketplace
- Oracle Hospitality Integration Platform Cloud Premium Remote Assistance
- Oracle Partner Network

Blogs

- Oracle Hospitality Integration Blogs

OPERA Cloud Resources

- OPERA Cloud Services

19 FAQs

APIs

Do I need to supply the x-hotelid in the header as well as the URL?

Yes, the header parameter is used for authorization where any **hotelIds** in the URL refer to the resource being accessed.

How can I call early adopter (v0) APIs and what are the implications?

See the Early Adopter API Program topic for details.

How long after a v0 API is promoted to v1 will I have to repoint my integration to v1?

Integrations calling v0 APIs have six months during which to point to v1 after a v0 API is promoted to v1. We highly recommend the Tolerant Reader pattern to mitigate this work.

Do I need to request an OAuth each time I make an API call?

No, please avoid doing this. Instead of requesting a token repeatedly, just request a token once per user and keep using that same token for the 1 hour it is valid. Additional calls are unnecessary and may add to your per cost calls.

Billing

If I am using Oracle Hospitality Integration Cloud Service, how do I view my billing and API usage?

View your billing and usage in the Billing Metrics area of the Oracle Cloud Console.

Will all API calls be counted in my API charged usage for Oracle Hospitality Integration Cloud Service?

No, 500, 502, 503, and 504 errors will not be charged, and the API calls you make to your own OPERA Cloud nonproduction environment will not be counted or charged as API Cloud usage.

How am I charged for consuming events?

With the streaming API, partners are charged for each event consumed at a rate of $0.001 per event. When replaying events, partners are charged the first time they replay an event. With the polling API, partners are charged for each API call made at a rate of $0.001 per API call.

For any other questions about Oracle invoicing, refer to Oracle Invoicing FAQ | Oracle United Kingdom.

Connecting to a Customer

How can a partner connect to an OPERA Cloud property?

See the Partners Moving to Production topic for details.

How can I verify that an integration user was created correctly?

1. Find the email you received when the hotel approved your integration user. This includes a URL for the Shared Security Domain identity server. Go to this URL.
2. Log in using your integration username and password.
3. Go to the **My Access** tab.
4. Verify you have the <Tenant>-WSACCESS role.

If you have the WSACCESS role, then the integration user is correctly set up and has been approved by the environment owner.

My integration suddenly stopped working!

First, check that your credentials are correct. Integration Users can be checked by logging in to the Shared Security Domain and clicking the **My Access** tab.

If this is a production integration and there has been a major incident with your integration, Oracle Hospitality will contact you using the contact details listed on the application in the Developer Portal. If you do not respond, Oracle Hospitality will disable your integration. If you have missed this communication and your integration has stopped working, reach out to Oracle Hospitality via a Service Request to request details of what changes must be made before the integration will be reenabled. We will work with you to reenable your integration once the issue has been resolved.

If Oracle Hospitality has not contacted you about your production integration, then reach out to the environment owner and ask the owner to reset your integration user password. If the environment is an OPERA Cloud Identity Manager environment, log in to the Developer Portal and reset the clientSecret from the **Environments** tab.

Eligibility

What are the minimum requirements for a partner to call production OPERA Cloud environments?

1. Your solution must be posted in the Oracle Hospitality Cloud Marketplace (listed under the OHIP product category).

   > **Note:**
   >
   > A current Oracle Partner Network Membership (that is, Member Level) is required to post a solution on the Marketplace.

   - To join the Oracle Partner Network (OPN), follow the steps here: https://www.oracle.com/middleeast/partnernetwork/program/join/
   - To publish your solution, follow the steps here: https://cloudmarketplace.oracle.com/marketplace/en_US/partnerLandingPage
2. Once your application is listed in the marketplace in the OHIP product category, email your listing ID to **hgbu_integrations_provisioning_grp@oracle.com** and Oracle will grant you access to call production OPERA Cloud Environments.
3. Create a new application by selecting **Production** at step 7 in Registering an Application.
4. Create an integration user within the customer's production tenancy and ensure the customer chain administrator has approved your user.

5. Follow the procedure in Adding an Environment and enter the user name of the integration user.

How can a customer confirm if their subscription is to OPERA Cloud Foundation?

If there is a customer employee at the hotel who is familiar with the hotel's OPERA contract, they can confirm this by reviewing their active subscriptions. Alternatively, the hotel can contact their Account Manager to verify this.

What are the minimum requirements for the streaming API?

For the streaming API, a hotel must be running OPERA Cloud Foundation version 22.3.0.1 or beyond. To get started, customers must engage with Oracle Professional Services.

How do I know if a customer is eligible to use OHIP?

We recommend partners check the following to ensure a smooth production move:

- The customer must be running OPERA Cloud and have an active subscription for OPERA Cloud Foundation.

> **Note:**
>
> OHIP is only available for OPERA Cloud Foundation customers at this point.

Events

How long will OHIP retain events?

When using the streaming API, OHIP retains messages for 7 days; therefore, messages sent in those 7 days can be replayed (see Replaying Messages for more information). Messages not consumed within 7 days are lost and cannot be replayed.

When using the polling API, events are retained forever, but events cannot be replayed via the polling API.

How frequently should I connect to consume events?

It is recommended to keep the WebSocket connected, but depending on use case, this may be neither practical nor needed. However, connecting at least every few days ensures no messages will be lost.

There is no guarantee that a subscription to consume events will remain in place if the application fails to consume the subscribed events for 2 weeks.

What do the terms WebSocket and GraphQL mean?

To learn more, read our blog posts at https://blogs.oracle.com/hospitality/post/ohip-introduces-state-of-the-art-streaming-api-and-rich-analytics and https://blogs.oracle.com/hospitality/post/ohip-streaming-api-understanding-our-strategy.

Is the polling API for fetching business events going away?

No, for some use cases it is the right fit, so it will be available alongside the streaming API.

Can I use the streaming API to fetch ARI (availability, rates, and inventory)?

Yes, but we are working on a more cost effective alternative for ARI.

How many events can I consume at once?

With the polling API, a maximum of 20 events can be retrieved in one API call. The streaming API makes events available as soon as they occur, so there is no maximum number of events.

Will streaming affect throttling?

Only inbound API calls to OPERA Cloud are throttled; events produced from OPERA Cloud are not.

The events tab is not appearing in the application window. What am I missing?

You must have at least one OPERA Cloud environment with streaming enabled added to your Environments tab. You can identify if streaming is enabled for an environment by looking for the Streaming Enabled flag on the environment's card (located under the Environments tab).

Is it possible to clear the backlog of events for my application?

No option is currently available to clear the backlog of events, and you must continue to process those events. Alternatively, you can create a new application and delete the old one if it is not required.

Onboarding

Do we need to sign an agreement to work with OHIP as an integration partner?

No, you must purchase Oracle Hospitality Integration Cloud Service from the Oracle Shop, but there is no further agreement required.

Is there a sandbox provided for testing?

As a customer. we recommend developing and testing in your UAT environment. For more information, see the Quick Start Guide for customers.

Integration partners have complimentary access to a sandbox environment. For more information, see the Quick Start Guide for partners.

OPN and Marketplace

Is a validation or certification needed for OHIP?

A review is optional. By default, OHIP is intended to offer a self-service experience that enables developers to bring their solutions to life without the need for a formal validation. If you would like to request an Oracle Hospitality review, please email hospitality-integrations_ww@oracle.com.

How do I add my application to the Cloud Marketplace?

Click the link below and follow the instructions on the page:

https://cloudmarketplace.oracle.com/marketplace/en_US/partnerLandingPage

Where can I find information for joining the Oracle Partner Network (OPN)?

To join the Oracle Partner Network (OPN), follow the steps here: https://www.oracle.com/middleeast/partnernetwork/program/join/

> **Note:**
>
> Please note that the minimum level required for OHIP is the OPN Member Level.

20

Appendix A

Web Service Error Codes

Error codes in OPERA Cloud are standardized based on the format Message MODxxxxx. For example, the Reservation module has error RSV00001. Other than the module specific errors, there are generic error codes that apply to all modules. The error code for generic errors is GENxxxxx. For any system generated errors or unhandled exceptions, the detailed error is logged in the application log and the generic error is returned to the web service consumer.

Each web service has a response with the following common pattern:

SuccessType

Each web service response returns an element of Success type to indicate the successful processing of the This is used in conjunction with the Warning Type to report any warnings.

WarningType

After a message has been successfully processed to report warnings, this type returns the collection of warnings. Consumers can catch these warnings and take further action.

Multi-record handling operations provide warnings for partial failures. Successfully processed data changes are saved in the OPERA Cloud database.

ErrorType

Returned when the processing of a user's request generates an error. This type returns a collection of the errors, and the web service responds successfully without raising a SOAP fault

> **Note:**
>
> After handling an error, OPERA Cloud never raises a web services fault for business errors or unhandled exceptions. All errors are reported by the "Errors" element.

For a complete list of error codes, refer to the following Oracle Customer Support Portal article — Web Service Error Codes.