

Oracle® Cloud

Sign and Verify Messages Using OCI Vault



F97733-01
May 2024



Oracle Cloud Sign and Verify Messages Using OCI Vault,
F97733-01

Copyright © 2024, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

Contents

1 About This Recipe

Overview	1-1
System and Access Requirements	1-1

2 Before You Install the Recipe

3 Install and Configure the Recipe

Configure the OCI Vault REST Connection	3-1
Configure the Lookup Table	3-2

4 Activate and Run the Recipe

Preface

This document describes how to install, configure, and run this recipe in Oracle Integration 3.

Topics:

- [Documentation Accessibility](#)
- [Diversity and Inclusion](#)
- [Related Resources](#)
- [Conventions](#)

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Resources

For more information, see these Oracle resources:

- Oracle Integration documentation on the Oracle Help Center.
- Oracle Cloud at <http://cloud.oracle.com>.

Conventions

The following text conventions are used in this document.

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.

Convention	Meaning
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

About This Recipe

Use this recipe to sign and verify messages using Oracle Cloud Infrastructure (OCI) Vault.

Note:

This recipe is available as **Oracle Integration — OCI Vault | Sign and Verify Messages** in Oracle Integration. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe enables you to sign and verify messages received from external systems using either Oracle or customer-managed keys in OCI Vault.

The recipe uses the OCI Vault REST APIs to sign and verify messages received from external applications. It contains two integration flows:

- **Sign Message:** Accepts message in string format as part of the request payload and invokes the [OCI Vault Sign REST API](#), which generates a message signature and sends the signed message as part of the response.
- **Verify Message:** Accepts signed message as part of the request payload and invokes the [OCI Vault Verify REST API](#), which verifies the message signature and sends the response as true or false.

System and Access Requirements

- Oracle Integration, Version 24.04 or higher
- OCI Vault
- An account on OCI with the Administrator role
- OpenSSL version 3

2

Before You Install the Recipe

To access OCI Vault from Oracle Integration and sign or verify messages, you'll require to perform certain configuration tasks on OCI.

Log in to your OCI instance as an **Administrator** and perform the following tasks.

1. Create an OCI Vault. See [Managing Vaults](#).
2. Import an external private 2048-bit RSA key. See [Importing Asymmetric Keys](#).

When you create a new master encryption key or key version, you can import your own key material instead of letting the OCI Vault generate the key material internally. Importing your own key is called Bring Your Own Key (BYOK).

 **Note:**

This recipe uses BYOK, but you can also use Oracle managed and generated master encryption keys.

3. Get your Tenancy and User OCIDs. See [Where to Get the Tenancy's OCID and User's OCID](#).
4. Create your own private key or generate it from the OCI console. See [How to Generate an API Signing Key](#).
5. Get the key's fingerprint. See [How to Get the Key's Fingerprint](#).


3

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

1. On the Oracle Integration Home page, in the **Get started** section, click **Browse store**.
2. Find the recipe you want to install, then click **Get**.

A message confirms that the recipe was successfully installed, and the recipe card shows **In use**.

3. Click **Configure**  on the recipe to configure its resources.

The project workspace opens, displaying all the resources of the recipe. Configure the following resources before you activate and run the recipe.

Configure the OCI Vault REST Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Leave REST API Base URL selected.
Connection URL	Enter the Cryptographic Endpoint URL of the Vault you created earlier. See Before You Install the Recipe .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Select OCI Signature Version 1 .
Tenancy OCID	Enter the Tenancy OCID. See Before You Install the Recipe .
User OCID	Enter the User OCID.
Private Key	Enter the API key generated.


 **Note:**

Before you upload the private key, you must convert it into the PKCS1 format.

Finger Print	Enter the key's fingerprint.
---------------------	------------------------------

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.


6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the **Sign_Verify_Message_Lookup** lookup table and enter necessary values for the lookup keys.

1. In the Lookups section, click the lookup name.
2. Edit the lookup keys with appropriate values.

Key	Value
KeyID	Enter the OCID of the key. You can obtain the key OCID from the key details page.
KeyVersionId	Enter the OCID of the key version. You can obtain the key OCID from the version table.
SigningAlgorithm	Enter <code>SHA_256_RSA_PKCS_PSS</code>
EmailFrom	This is the email address from which notifications are sent in case of errors. Leave the default email address <code>no-reply@oracle.com</code> or enter an email of your choice.
EmailTo	Enter an email to which notifications are to be sent in case of errors.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

4

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. In the project workspace, click **Activate**. In the Activate project panel, with the default project deployment selected, choose an appropriate tracing option, then click **Activate**.

A message confirms that the integration has been activated. Refresh the page to view the updated status of the integration.

2. Run the recipe from an external application.

- a. In the Integrations section of the project workspace, click **Actions** **...** on the integration flow, then select **Run**.

- b. On the Configure and run page, click **Endpoint metadata**.

- c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.

- d. To sign a message from an external application, send a `POST` request to this endpoint URL along with the message to be signed. Provide the message in the `POST` request's **Body**. See the subsequent step for example request payload.

The recipe generates a signature for the message you pass in the request body.

- e. To verify a message and signature from an external application, send a `POST` request to this endpoint URL along with the message and signature to be verified. Provide the message and signature in the `POST` request's **Body**. See the subsequent step for example request payload.

The recipe verifies the signature against the message you pass in the request body.

3. Test the recipe.

- a. In the Integrations section of the project workspace, click **Actions** **...** on the **Sign Message** integration flow, then select **Run**.

- b. On the Configure and run page, in the Request section, click the **Body** tab and enter the message as request payload.

Example request payload to sign message:

```
{
  "message ":
  " {\\"EmpId\\":\\"011\\",\\"Name\\":\\"xyz\\",\\"Email\\":\\"xyz@oracle.com\\"}"
}
```

- c. In the Response section of the Configure and run page, you'll find the response returned. The response will be the signature of the message.

- d. In the Integrations section of the project workspace, click **Actions** **...** on the **Verify Message** integration flow, then select **Run**.

- e. On the Configure and run page, in the Request section, click the **Body** tab and enter the message and its signature as request payload.

Example request payload to verify message:

```
{
  "signature":
  "WX4vporM4aX0djjRHvCKN3Oxkx0gjixUSXVkpqCU2IijhI6ElUda9hOP5RynDF7nJrDd8D
  VKpFrdti9jT68APD4UTyqS+Dt7M/
  qsGzyjqdOsAEMeLLhYrZVMD+F3m49w1YGTase7o7PtSVW7uhoJ2Mt9i/
  A4fYVtYoR0T3IXK4DBr6==",
  "message ":
  " {\"EmpId\": \"011\", \"Name\": \"xyz\", \"Email\": \"xyz@oracle.com\"}"
}
```

- f. In the Response section of the Configure and run page, you'll find the response returned. The response would be either `true` or `false` depending on the validity of the signature against the message.

For example:

```
{
  "isSignatureValid" : "true"
}
```

4. Monitor the running of the integration flow in Oracle Integration.
 - a. In the project workspace, click **Observe**. You'll see the integration flow being triggered and running successfully.
 - b. To manage errors in your project, see [Manage Errors in a Project](#).

Related Documentation

- [Using the REST Adapter with Oracle Integration 3](#)