Oracle® Cloud

Using the Google Gmail Adapter with Oracle Integration 3

F45555-04
April 2024

Oracle Cloud Using the Google Gmail Adapter with Oracle Integration 3,

F45555-04

Primary Author: Oracle Corporation

Contents

Custom Legal Notice

Preface

This guide describes how to configure this adapter as a connection in an integration in Oracle Integration.

> **Note:**
>
> The use of this adapter may differ depending on the features you have, or whether your instance was provisioned using Standard or Enterprise edition. These differences are noted throughout this guide.

Topics:

- Audience
- Documentation Accessibility
- Diversity and Inclusion
- Related Resources
- Conventions

Audience

This guide is intended for developers who want to use this adapter in integrations in Oracle Integration.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `https://www.oracle.com/corporate/accessibility/`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `https://support.oracle.com/portal/` or visit `Oracle Accessibility Learning and Support` 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

See these Oracle resources:

- Oracle Cloud at `http://cloud.oracle.com`
- *Using Integrations in Oracle Integration 3*
- *Using the Oracle Mapper with Oracle Integration 3*
- Oracle Integration documentation on the Oracle Help Center.

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.
italic	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

Understand the Google Gmail Adapter

Review the following conceptual topics to learn about the Google Gmail Adapter and how to use it as a connection in integrations in Oracle Integration. A typical workflow of adapter and integration tasks is also provided.

Topics:

- Google Gmail Adapter Capabilities
- What Application Version Is Supported?
- Workflow to Create and Add a Google Gmail Adapter Connection to an Integration

> **Note:**
>
> There are overall service limits for Oracle Integration. A service limit is the quota or allowance set on a resource. See Service Limits.

Google Gmail Adapter Capabilities

The Google Gmail Adapter enables you to create an integration with a Google Gmail application.

Google Gmail is a free email service provided by Google. Users can access Google Gmail as secure web mail and also through the Post Office Protocol version 3 (POP3) or Internet Message Access Protocol version 4 (IMAP4) protocols.

The Google Gmail Adapter is one of many predefined adapters included with Oracle Integration. You can configure the Google Gmail Adapter as a connection in an integration in Oracle Integration.

What Application Version Is Supported?

For information about which application version is supported by this adapter, see the Connectivity Certification Matrix.

Workflow to Create and Add a Google Gmail Adapter Connection to an Integration

You follow a very simple workflow to create a connection with an adapter and include the connection in an integration in Oracle Integration.

Step	Description	More Information
1	Create the adapter connections for the applications you want to integrate. The connections can be reused in multiple integrations and are typically created by the administrator.	Create a Google Gmail Adapter Connection
2	Create the integration. When you do this, you add trigger and invoke connections to the integration.	Understand Integration Creation and Best Practices and Add the Google Gmail Adapter Connection to an Integration
3	Map data between the trigger connection data structure and the invoke connection data structure.	Map Data in *Using Integrations in Oracle Integration 3*
4	(Optional) Create lookups that map the different values used by those applications to identify the same type of object (such as gender codes or country codes).	Manage Lookups in *Using Integrations in Oracle Integration 3*
5	Activate the integration.	Manage Integrations in *Using Integrations in Oracle Integration 3*
6	Monitor the integration on the dashboard.	Monitor Integrations During Runtime in *Using Integrations in Oracle Integration 3*
7	Track payload fields in messages during runtime.	Assign Business Identifiers for Tracking Fields in Messages and Track Integration Instances in *Using Integrations in Oracle Integration 3*
8	Manage errors at the integration level, connection level, or specific integration instance level.	Manage Errors in *Using Integrations in Oracle Integration 3*

2

Create a Google Gmail Adapter Connection

A connection is based on an adapter. You define connections to the specific cloud applications that you want to integrate.

Topics:

- Prerequisites for Creating a Connection
- Create a Connection
- Upload a Certificate to Connect with External Services

Prerequisites for Creating a Connection

To use the Google Gmail Adapter, you must first have access to the Google Gmail API for your integration. To access the Google Gmail API, you must create a Google project.

To create a Google Gmail project:

1. Log in to your Google account and access the Google Gmail Developer Console at https://console.developers.google.com.
2. In the **Title** bar of the Google Cloud Platform page, click the **Select a project** drop-down list.
3. Select an existing project to use or create a new project.

   > **Note:**
   >
   > If you create a new project, ensure that the project is selected from the drop-down list after saving the new project. You can view the project's dashboard only if you select the project.

4. Click **ENABLE APIS AND SERVICES**. Alternatively, click **Library** in the left navigation pane.

   The API Library page opens.
5. In the **Search** field, enter `Gmail API`, and then select **Gmail API** from the search results.
6. In the Gmail API window, click **MANAGE**. Alternatively, for a new project, click **ENABLE**.
7. In the left navigation pane, click **Credentials**.
8. At the top of the Credentials window, click **CREATE CREDENTIALS**, and then choose **OAuth client ID**.
9. If you created a new project in step 3, perform the following additional configurations. Otherwise, proceed to step 10.
   a. In the Create OAuth client ID window, click **CONFIGURE CONSENT SCREEN**.
   b. In the OAuth consent screen window, select **External** as the user type, and click **CREATE**.

c. In the App information window, enter the necessary information in the required fields, then click **SAVE AND CONTINUE**.

d. Go back to the Credentials window.

e. Click **Credentials** in the left navigation pane.

10. In the **Application type** field, select **Web Application** from the drop-down list.
11. Enter a name for the OAuth Client ID in the **Name** field.
12. Scroll to the **Authorized redirect URIs** section, and click **ADD URI**.
13. Enter the following URL in the **URIs** field.

> **Note:**
>
> If you don't know the following information, check with your administrator:
>
> - If your instance is new or upgraded from Oracle Integration Generation 2 to Oracle Integration 3.
> - The complete instance URL with the region included (required for new instances).

For Connections…	Include the Region as Part of the Redirect URL?	Example of Redirect URL to Specify…
Created on new Oracle Integration 3 instances	Yes.	`https://OIC_instance_URL.region.ocp.oraclecloud.com/icsapis/agent/oauth/callback`
Created on instances upgraded from Oracle Integration Generation 2 to Oracle Integration 3	No. This applies to both: • New connections created after the upgrade • Existing connections that were part of the upgrade	`https://OIC_instance_URL.ocp.oraclecloud.com/icsapis/agent/oauth/callback`

14. Click **Create**.

    You receive a confirmation message that the OAuth Client has been created.

15. In the resulting dialog, copy the **Client ID** and **Client Secret**. Note the client ID and secret values because you'll need these values when configuring the Gmail connection on the Connections page.
16. Click **OK**.
17. Return to the Credentials window. The OAuth 2.0 client ID that you created is listed under the **OAuth 2.0 Client IDs** section.

> **Note:**
>
> Before creating a Gmail connection, you must upload the trusted Google Gmail public certificate to Oracle Integration. The trusted Google Gmail public certificate can be downloaded from https://gmail.com. Rename the `GoogleCertificate.txt` file extension to `.cer`. See Certificate Errors to obtain the trusted certificate from Google and Upload a Certificate to Connect with External Services to upload the certificate.

Create a Connection

Before you can build an integration, you must create the connections to the applications with which you want to share data.

To create a connection in Oracle Integration:

1. In the navigation pane, click **Design**, then **Connections**.
2. Click **Create**.

   > **Note:**
   >
   > You can also create a connection in the integration canvas. See Define Inbound Triggers and Outbound Invokes.

3. In the Create connection panel, select the adapter to use for this connection. To find the adapter, scroll through the list, or enter a partial or full name in the **Search** field.
4. Enter the information that describes this connection.

Element	Description
Name	Enter a meaningful name to help others find your connection when they begin to create their own integrations.
Identifier	Automatically displays the name in capital letters that you entered in the **Name** field. If you modify the identifier name, don't include blank spaces (for example, `SALES OPPORTUNITY`).

Element	Description
Role	Select the role (direction) in which to use this connection (trigger, invoke, or both). Only the roles supported by the adapter are displayed for selection. When you select a role, only the connection properties and security policies appropriate to that role are displayed on the Connections page. If you select an adapter that supports both invoke and trigger, but select only one of those roles, you'll get an error when you try to drag the adapter into the section you didn't select. For example, assume you configure a connection for the Oracle Service Cloud (RightNow) Adapter as only an **invoke**. Dragging the adapter to a **trigger** section in the integration produces an error.
Keywords	Enter optional keywords (tags). You can search on the connection keywords on the Connections page.
Description	Enter an optional description of the connection.
Share with other projects	**Note**: This field only appears if you are creating a connection in a project. Select to make this connection publicly available in other projects. Connection sharing eliminates the need to create and maintain separate connections in different projects. When you configure an adapter connection in a different project, the **Use a shared connection** field is displayed at the top of the Connections page. If the connection you are configuring matches the same type and role as the publicly available connection, you can select that connection to reference (inherit) its resources. See Add and Share a Connection Across a Project.

5. Click **Create**.

   Your connection is created. You're now ready to configure the connection properties, security policies, and (for some connections) access type.

Configure Connection Security

Configure security for your Google Gmail connection by selecting the security policy and specifying the client ID and client secret. The security policy grants you authorization access to the resources of the Google Gmail application. When your Google Gmail connection requests access to the resources stored on the resource server, your connection is authenticated by sending the client ID and the client secret to the authorization server.

1. Go to the **Security** section.
2. In the **Security Policy** field, note that the **Google OAuth Authorization Code Credentials** security policy is displayed by default, and cannot be deselected.

3. In the **Client ID** field, enter the client ID created after completing the steps in Prerequisites for Creating a Connection.
4. In the **Client Secret** field, enter the client secret created after completing the steps in Prerequisites for Creating a Connection.
5. In the **Scope** field, click to display a list of available scopes:
   - `https://mail.google.com/`
   - `https://www.googleapis.com/auth/gmail.compose`
   - `https://www.googleapis.com/auth/gmail.metadata`
   - `https://www.googleapis.com/auth/gmail.labels`
   - `https://www.googleapis.com/auth/gmail.modify`
   - `https://www.googleapis.com/auth/gmail.readonly`
   - `https://www.googleapis.com/auth/gmail.send`

   See https://developers.google.com/gmail/api/auth/scopes or https://developers.google.com/identity/protocols/googlescopes for more details.
6. Copy and paste the scopes to use, separated by blank spaces.
7. Click **Provide Consent** to allow consent.

   A dialog is displayed indicating that an OAuth request is being initiated from Oracle Integration to Google Gmail. If the redirect URL in the project in the developer console is correct, the consent screen is displayed. Otherwise. an error occurs indicating that there is a redirect URI mismatch. Once consent is given, a successful consent page is displayed.

Test the Connection

Test your connection to ensure that it's configured successfully.

1. In the page title bar, click **Test**. What happens next depends on whether your adapter connection uses a Web Services Description Language (WSDL) file. Only some adapter connections use WSDLs.

If Your Connection...	Then...
Doesn't use a WSDL	The test starts automatically and validates the inputs you provided for the connection.
Uses a WSDL	A dialog prompts you to select the type of connection testing to perform: • **Validate and Test**: Performs a full validation of the WSDL, including processing of the imported schemas and WSDLs. Complete validation can take several minutes depending on the number of imported schemas and WSDLs. No requests are sent to the operations exposed in the WSDL. • **Test**: Connects to the WSDL URL and performs a syntax check on the WSDL. No requests are sent to the operations exposed in the WSDL.

2. Wait for a message about the results of the connection test.
   - If the test was successful, then the connection is configured properly.
   - If the test failed, then edit the configuration details you entered. Check for typos and verify URLs and credentials. Continue to test until the connection is successful.

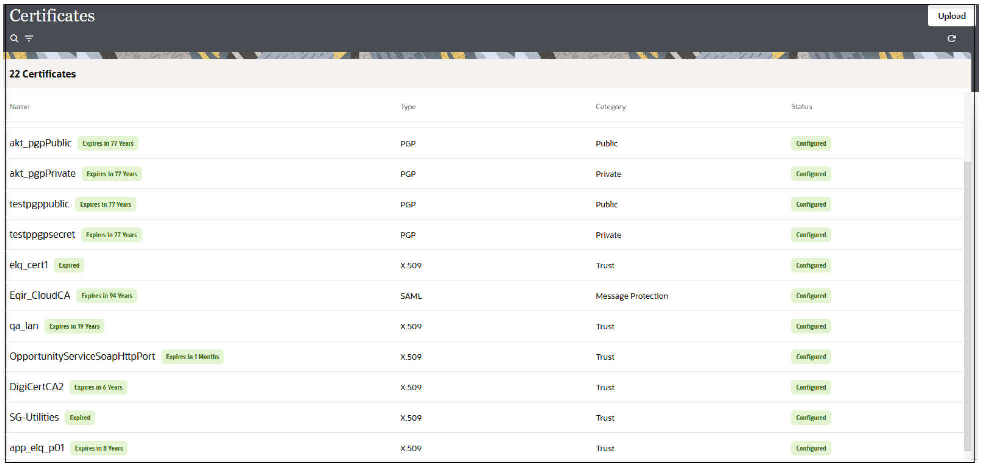
3. When complete, click **Save**.

Upload a Certificate to Connect with External Services

Certificates allow Oracle Integration to connect with external services. If the external service/endpoint needs a specific certificate, request the certificate and then import it into Oracle Integration.

If you make an SSL connection in which the root certificate does not exist in Oracle Integration, an exception error is thrown. In that case, you must upload the appropriate certificate. A certificate enables Oracle Integration to connect with external services. If the external endpoint requires a specific certificate, request the certificate and then upload it into Oracle Integration.

1. Sign in to Oracle Integration.
2. In the navigation pane, click **Settings**, then **Certificates**.
   All certificates currently uploaded to the trust store are displayed on the Certificates page.
3. Click **Filter** ☰ to filter by name, certificate expiration date, status, type, category, and installation method (user-installed or system-installed). Certificates installed by the system cannot be deleted.



22 Certificates

Name	Type	Category	Status
akt_pgpPublic Expires in 77 Years	PGP	Public	Configured
akt_pgpPrivate Expires in 77 Years	PGP	Private	Configured
testpgppublic Expires in 77 Years	PGP	Public	Configured
testppgpsecret Expires in 77 Years	PGP	Private	Configured
elq_cert1 Expired	X.509	Trust	Configured
Eqir_CloudCA Expires in 94 Years	SAML	Message Protection	Configured
qa_lan Expires in 19 Years	X.509	Trust	Configured
OpportunityServiceSoapHttpPort Expires in 1 Months	X.509	Trust	Configured
DigiCertCA2 Expires in 6 Years	X.509	Trust	Configured
SG-Utilities Expired	X.509	Trust	Configured
app_elq_p01 Expires in 8 Years	X.509	Trust	Configured

4. Click **Upload** at the top of the page.
   The Upload certificate panel is displayed.
5. Enter an alias name and optional description.
6. In the **Type** field, select the certificate type. Each certificate type enables Oracle Integration to connect with external services.
   - Digital Signature
   - X.509 (SSL transport)
   - SAML (Authentication & Authorization)
   - PGP (Encryption & Decryption)
   - Signing key

Digital Signature

The digital signature security type is typically used with adapters created with the Rapid Adapter Builder. See Learn About the Rapid Adapter Builder in Oracle Integration in *Using the Rapid Adapter Builder with Oracle Integration 3*.

1. Click **Browse** to select the digital certificate. The certificate must be an X509Certificate. This certificate provides inbound RSA signature validation. See Implement Digital Signature Validation (RSA) in *Using the Rapid Adapter Builder with Oracle Integration 3*.
2. Click **Upload**.

X.509 (SSL transport)

1. Select a certificate category.
   a. **Trust**: Use this option to upload a trust certificate.
      i. Click **Browse**, then select the trust file (for example, `.cer` or `.crt`) to upload.
   b. **Identity**: Use this option to upload a certificate for two-way SSL communication.
      i. Click **Browse**, then select the keystore file (`.jks`) to upload.
      ii. Enter the comma-separated list of passwords corresponding to key aliases.

      > **Note:**
      >
      > When an identity certificate file (`.jks`) contains more than one private key, all the private keys must have the same password. If the private keys are protected with different passwords, the private keys cannot be extracted from the keystore.

      iii. Enter the password of the keystore being imported.
   c. Click **Upload**.

SAML (Authentication & Authorization)

1. Note that **Message Protection** is automatically selected as the only available certificate category and cannot be deselected. Use this option to upload a keystore certificate with SAML token support. Create, read, update, and delete (CRUD) operations are supported with this type of certificate.
2. Click **Browse**, then select the certificate file (`.cer` or `.crt`) to upload.
3. Click **Upload**.

PGP (Encryption & Decryption)

1. Select a certificate category. Pretty Good Privacy (PGP) provides cryptographic privacy and authentication for communication. PGP is used for signing, encrypting, and decrypting files. You can select the private key to use for encryption or decryption when configuring the stage file action.
   a. **Private**: Uses a private key of the target location to decrypt the file.
      i. Click **Browse**, then select the PGP file to upload.
      ii. Enter the PGP private key password.

b. **Public**: Uses a public key of the target location to encrypt the file.
    i. Click **Browse**, then select the PGP file to upload.
    ii. In the **ASCII-Armor Encryption Format** field, select **Yes** or **No**.
        - **Yes** shows the format of the encrypted message in ASCII armor. ASCII armor is a binary-to-textual encoding converter. ASCII armor formats encrypted messaging in ASCII. This enables messages to be sent in a standard messaging format. This selection impacts the visibility of message content.
        - **No** causes the message to be sent in binary format.
    iii. From the **Cipher Algorithm** list, select the algorithm to use. Symmetric-key algorithms for cryptography use the same cryptographic keys for both encryption of plain text and decryption of cipher text. The following supported cipher algorithms are FIPS-compliant:
        - AES128
        - AES192
        - AES256
        - TDES

c. Click **Upload**.

Signing key

A signing key is a secret key used to establish trust between applications. Signing keys are used to sign ID tokens, access tokens, SAML assertions, and more. Using a private signing key, the token is digitally signed and the server verifies the authenticity of the token by using a public signing key. You must upload a signing key to use the OAuth Client Credentials using JWT Client Assertion and OAuth using JWT User Assertion security policies in REST Adapter invoke connections. Only PKCS1- and PKCS8-formatted files are supported.

1. Select **Public** or **Private**.
2. Click **Browse** to upload a key file.
   If you selected **Private**, and the private key is encrypted, a field for entering the private signing key password is displayed after key upload is complete.
3. Enter the private signing key password. If the private signing key is not encrypted, you are not required to enter a password.
4. Click **Upload**.

3

Add the Google Gmail Adapter Connection to an Integration

When you drag the Google Gmail Adapter into the invoke area of an integration, the Adapter Endpoint Configuration Wizard appears. This wizard guides you through configuration of Google Gmail Adapter endpoint properties.

These topics describe the wizard pages that guide you through configuration of the Google Gmail Adapter as an invoke in an integration. The Google Gmail Adapter cannot be used as a trigger in an integration.

Topics:

Basic Info Page

You can enter a name and description on the Basic Info page of each adapter in your integration.

Element	Description
What do you want to call your endpoint?	Provide a meaningful name so that others can understand the responsibilities of this connection. You can include English alphabetic characters, numbers, underscores, and hyphens in the name. You can't include the following characters: • No blank spaces (for example, `My Inbound Connection`) • No special characters (for example, `#;83&` or `righ(t)now4`) except underscores and hyphens • No multibyte characters
What does this endpoint do?	Enter an optional description of the connection's responsibilities. For example: `This connection receives an inbound request to synchronize account information with the cloud application.`

Invoke Operation Selection Page

Select the Google Gmail API operation to perform.

Element	Description
Select Operation	Select the Google Gmail API operation to perform. - **Notification Watch** - **List Threads** - **Send Message with RFC 2822 Format (Deprecated)** **Note**: Do *not* select this operation. It is provided for backward compatibility only. Instead, select **Send Message**. - **Get Thread** - **Trash Thread** - **List Messages** - **Get Message** - **Get Message Attachment** - **Send Message** - **List Drafts** - **Create Draft** - **Get Draft** - **Delete Draft** - **Send Draft** - **List Labels** - **Create Label** - **Delete Label** The Google Gmail resource URI and template parameter associated with your selected operation are displayed on the Summary page of this wizard. For information about the Google Gmail API, visit the following URL: https://developers.google.com/gmail/api/v1/reference

Invoke Parameters Page

Select the parameters to use with the Google Gmail API operation. Not all APIs selected on the Operations page support parameters.

Element	Description
Query Parameters	Type the initial letters to filter the display of query parameters.
Available Query Parameters	Select the query parameters.
Selected Query Parameters	Displays the selected query parameters.

Summary Page

You can review the specified adapter configuration values on the Summary page.

Element	Description
Summary	Displays a summary of the configuration values you defined on previous pages of the wizard. The information that is displayed can vary by adapter. For some adapters, the selected business objects and operation name are displayed. For adapters for which a generated XSD file is provided, click the XSD link to view a read-only version of the file. To return to a previous page to update any values, click the appropriate tab in the left panel or click **Go back**. To cancel your configuration details, click **Cancel**.

4 Troubleshoot the Google Gmail Adapter

Review the following topics to learn about troubleshooting issues with the Google Gmail Adapter.

Topics:

- Certificate Errors
- Send Message Operation Failure

Certificate Errors

Note the following certificate errors.

If the certificate is not uploaded, the following error is displayed.

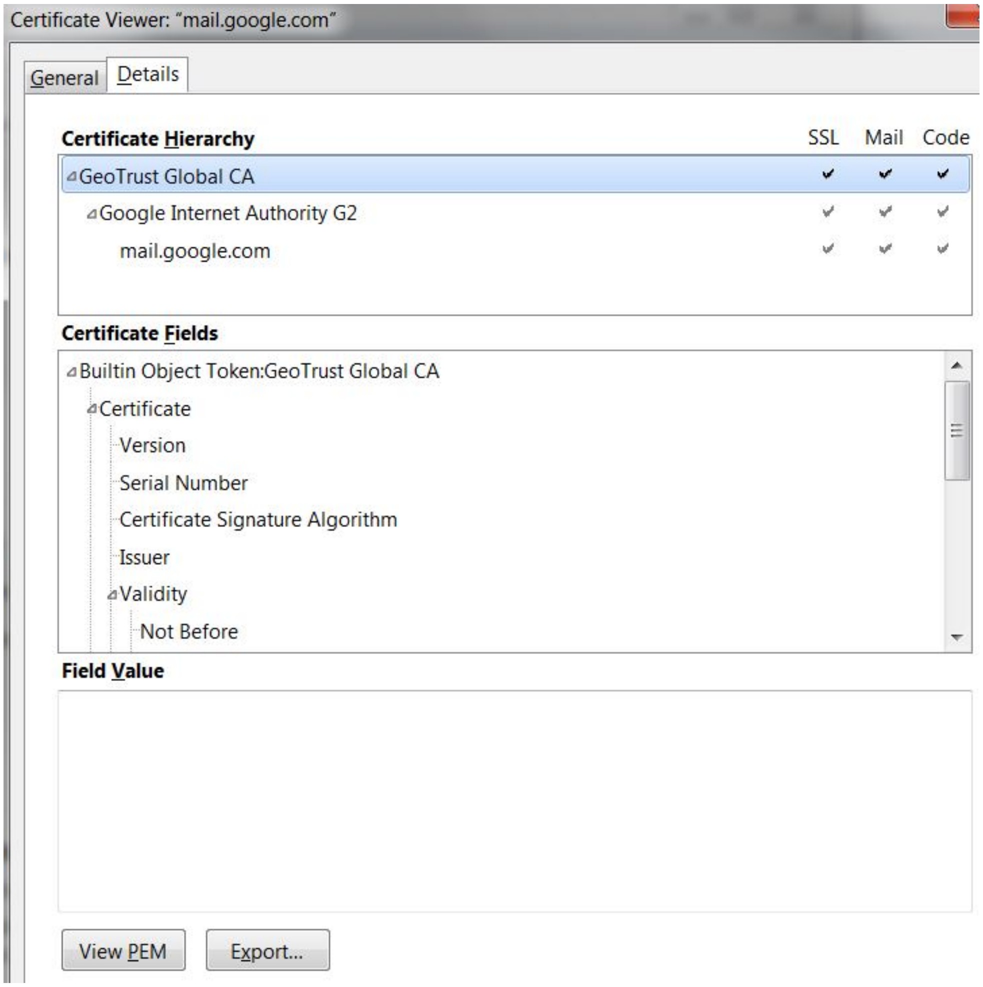
```
Authorization Failed: PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to find
valid certification path to requested target
```

If the certificate uploaded is incomplete or the client ID and client secret are invalid, the following error is displayed.

```
Authorization Failed: null
```

To upload the trusted Google certificate to Oracle Integration:

1. Open your browser.
2. Log in to https://gmail.com/.
3. Click the **Lock** icon in front of the URL.
4. Click **More Information** > **Security**.
5. Click **View Certificate** > **Details**.
6. Select the root certificate in the **Certificate Hierarchy** section.
7. Export the root certificate.



Certificate Hierarchy

	SSL	Mail	Code
GeoTrust Global CA	✓	✓	✓
Google Internet Authority G2	✓	✓	✓
mail.google.com	✓	✓	✓

Certificate Fields

Builtin Object Token:GeoTrust Global CA
Certificate
Version
Serial Number
Certificate Signature Algorithm
Issuer
Validity
Not Before

Field Value

8. Upload this certificate in Oracle Integration. See Upload a Certificate to Connect with External Services.

Send Message Operation Failure

If the `send message` operation fails with an error similar to the following, the `uploadType` query parameter is mapped with the wrong value.

```
<genericRestFault>
   <errorCode>REST_REQ_HDR_ERR</errorCode>
      <errorPath>
         <![CDATA[An error occurred while processing headers in the
target REST  endpoint.]]>
      </errorPath>
   <instance>
      <![CDATA[Target REST endpoint headers could not be set.[[The
values accepted for query parameter uploadType are [media,multipart,
resumable] but found 'media' instead.]]]]>
   </instance>
</genericRestFault>
```

The uploadType query parameter is hard-coded in the mapper and enclosed with quotes. The send message operation sends mail from the account used in the connection. It expects the value to be of MIME content. If the send message operation fails with an error similar to the following, the reason is that invalid MIME content is mapped to a raw element.

```
"error": {
 "errors": [
  {
   "domain": "global",
   "reason": "invalidArgument",
   "message": "Recipient address required"
  }
 ],
 "code": 400,
 "message": "Recipient address required"
}
```

Sample mail content is shown below:

```
From: sender@email.com
To: receiver@email.com
Subject: Mail Subject
MIME-Version: 1.0
Content-Type: text/plain; charset=utf-8
Content-Transfer-Encoding: 7bit

    Dear Sender,

        This is a sample mail sent using ICS Google Mail Adapter.

        Thank You!!

Regards,
Sender
```

The above mail content can be built using the XSLT mapper with the following steps.

1. Export the flow from Oracle Integration.
2. Manually edit the mapper XSLT with required values in the format using the sample provided below.
3. Save the XSLT and re-import the flow into Oracle Integration.

```
<xsl:template match="/" xml:id="id_11">
    <xsl:variable name="emailContent">
        <xsl:value-of select="concat('From: ','sender@email.com')"/>
        <xsl:text>&#xa;</xsl:text>
        <xsl:value-of select="concat('To: ','receiver@email.com')"/>
        <xsl:text>&#xa;</xsl:text>
        <xsl:value-of select="concat('Subject: ','Mail Subject')"/>
        <xsl:text>&#xa;</xsl:text>
        <xsl:value-of select="concat('MIME-Version: ','1.0')"/>
        <xsl:text>&#xa;</xsl:text>
        <xsl:value-of select="concat('Content-Type: ','text/plain;
charset=utf-8')"/>
        <xsl:text>&#xa;</xsl:text>
        <xsl:value-of select="concat('Content-Transfer-Encoding:
','7bit')"/>
        <xsl:text>&#xa;</xsl:text>
        <xsl:text>&#xa;</xsl:text>
        <xsl:value-of select="'    Dear Sender,'"/>
        <xsl:text>&#xa;</xsl:text>
        <xsl:text>&#xa;</xsl:text>
        <xsl:value-of select="'This is a sample mail sent using ICS
Google Mail Adapter.'"/>
        <xsl:text>&#xa;</xsl:text>
        <xsl:text>&#xa;</xsl:text>
        <xsl:value-of select="'    Thank You!!'"/>
        <xsl:text>&#xa;</xsl:text>
        <xsl:text>&#xa;</xsl:text>
        <xsl:value-of select="'Regards,'"/>
        <xsl:text>&#xa;</xsl:text>
        <xsl:value-of select="'Sender'"/>
    </xsl:variable>
    <nstrgmpr:sendMsg xml:id="id_12">
        <nstrgmpr:Messages.definitions.requestPayLoadForSendMsg
xml:id="id_18">
             <nstrgmpr:raw xml:id="id_19">
                  <xsl:value-of select="$emailContent"/>
             </nstrgmpr:raw>
        </nstrgmpr:Messages.definitions.requestPayLoadForSendMsg>
        <nstrgmpr:QueryParameters xml:id="id_16">
             <nstrgmpr:uploadType xml:id="id_17">media</
nstrgmpr:uploadType>
        </nstrgmpr:QueryParameters>
    </nstrgmpr:sendMsg>
</xsl:template>
```