Integrate with CA API Gateway

As a company that uses CA Live API Creator to build APIs for internal application development or integration with external partners and developers, you may want to:

- Expose and securely manage your APIs using the CA API Gateway as an API proxy.
- Reduce the cost of ownership to authenticate API users by using your available and configured identity provider, such as LDAP, Active Directory, or CA Single Sign On.
- Determine users' data access permissions and customize these group-to-role mappings to fit your environment by mapping your Gateway-retrieved user groups to Live API Creator roles.
- Allow developers to explore and use your APIs for development and integration purposes by exposing APIs from Live API Creator in CA API Management SaaS or CA API Developer Portal (the Portal).

To accomplish this, you can integrate Live API Creator with API Gateway and the Portal (either on-premise or SaaS).

CONTENTS

CREATE YOUR API PROJECT

MANAGE YOUR API PROJECT

API PROPERTIES

DEFINE CUSTOM REST RESOURCES

SPECIFY YOUR BUSINESS RULES

SECURITY

LANGUAGE EXAMPLES

DEBUG YOUR API PROJECT

TEST USING THE REST LAB

VIEW LOGGING INFORMATION

AUTH TOKENS

GENERATE A LIST OF API ISSUES

METRICS

RETRIEVE AN API KEY

DATA EXPLORER

TROUBLESHOOTING

CONTACT CA SUPPORT

INTEGRATE WITH CA API GATEWAY

PUBLISH AN API TO API GATEWAY

CONFIGURE

Integration Use Cases

Expose the API project that you created with Live API Creator for mobile/client developers to discover in an API Developer Portal and consume through API Gateway for the following use cases:

- Use API Gateway as an authentication provider.
- Map API Gateway-identity provider roles to Live API Creator roles.
- Control API traffic at run-time between API consumers and the API Server using API Gateway as a proxy.
- Expose APIs registered by Live API Creator in API Gateway from within the Portal.

Overall Integration Workflow

Use the following workflow to deploy and integrate Live API Creator, API Gateway, and API Developer Portal:

1. The API Gateway and Live API Creator administrators set up mutual authentication between API Server and API Gateway.
2. The API Gateway administrator installs the API Gateway - Live API Creator Integration solution kit.
3. The API Gateway and Live API Creator administrators publish an API from Live API Creator to the Gateway.
4. The API Gateway administrator configures the published API project.
5. The API Gateway administrator tests the consumption of the published API project.
6. (Optional) The API Gateway administrator exposes the

published API project from API Gateway to the Portal.

7. (Optional) The API consumer discovers the published APIs using the Portal and consumes them through API Gateway.

Verify Prerequisites

Ensure that you have completed the following prerequisites:

- You have created an API using Live API Creator.
- You have access to CA API Gateway version 9.0 or higher and the associated Policy Manager.
- (If you are integrating with the Portal) You have access to API Management SaaS or API Developer Portal, version 3.5 or higher.
- You have obtained the `LACGatewayIntegration.sskar` SKAR file for the API Gateway - Live API Creator Integration solution kit. You can obtain this file from the `<Live API Creator Installation folder>/samples/Gateway` folder.
- You have installed an API testing tool (Postman, cURL, SoapUI, etc.) if you plan to manually test the consumption of a published API project.

Set Up Mutual Authentication between API Server and API Gateway

API Gateway requires that you set up mutual authentication between API Server and API Gateway based on a public key infrastructure (PKI). Mutual authentication provides last-mile security and trust between API Gateway and Live API Creator.

Use the following process to set up mutual authentication:

1. The Gateway administrator creates a public/private key pair to be used by API Server.
   Note: The Jetty package contains a default key pair which is described in the README.txt.
2. The Gateway administrator adds the API Server public key to API Gateway keystore.

3. The Gateway administrator exports the Gateway public key.
4. The Live API Creator administrator imports the public/private key pair to API Server's keystore.
   Note: The Jetty package contains a default key pair which is described in the README.txt.
5. The Live API Creator administrator adds API Gateway's public key to API Server's keystore.
6. The Live API Creator administrator configures the API Server for SSL and HTTPS.

Create API Server's Public/Private Key Pair

You can create the public and private keys using API Gateway or alternative tools, such as OpenSSL. The following procedure describes how to create the pair using API Gateway.

1. Open the Policy Manager and connect to the Gateway.
2. Select Tasks, Manage Private Keys.
   The Manage Private Keys window opens.
3. Click Create.
   The Create Private Key window opens.
4. Enter an alias name for the API Server's public/private key pair (for example, lacssl) and then click Create.

API Server's public/private key pair is created and displays in the list on the Manage Private Keys window.

Export API Server's Private Key

API Server's keystore requires the API Server's public/private key pair. Export the key so that you can add it to the API Server keystore. API Gateway Server's private key is a PKCS#12 file.

1. In the Policy Manager, with the Manage Private Keys window open, select your API Server's public/private key pair, click Properties.
   The Private Key Properties window opens.
2. Click Export Key.
   The Enter Export Passphrase window opens.

3. Enter the password you want to use to protect your private key, and then click OK.
   The Save As dialog opens.
4. Enter a file name for this private key (for example, `lacssl.p12`), and then click Save.
5. Close the Private Key Properties window by clicking Cancel.

The API Server's private key is exported.

Export API Server's Public Key

Complete this procedure if you will be configuring mutual trust between API Server and multiple API Gateway servers.

1. In the Policy Manager, with the Manage Private Keys window open, select API Server's private/public key pair and click Properties.
   The Private Key Properties window opens.
2. Click View Certificate.
   The Certificate Properties window opens.
3. Click Export. The Save certificate dialog opens.
4. Enter a file name for this public key (for example, `lacssl.pem`), select PEM as the file format, and then click Save.
5. Click Close to close the Certificate Properties window.
6. Close the Private Key Properties window by clicking Cancel.

API Server's public key is exported.

Export API Gateway's Public Key

Enable API Server to authenticate API Gateway based on API Gateway's client certificate by exporting API Gateway's certificate. The certificate contains the public key.

1. In the Policy Manager, with the Manage Private Keys window open, select the API Gateway Server's private/public key pair and click Properties. The Private Key Properties window opens.

2. Click View Certificate. The Certificate Properties window opens.
3. Click Export. The Save certificate dialog opens.
4. Enter a file name (for example, `gatewayssl.pem`), select PEM as the file format, and then click Save.
5. Click Close to close the Certificate Properties window.
6. Click Cancel to close the Private Key Properties window.
7. Click Close to close the Manage Private Keys window.

API Gateway's public key (a second .pem file) is created and exported.

Configure API Gateway for Mutual Authentication

API Gateway requires the API Server's public key for SSL/HTTPS communication. Configure API Gateway to use the key by importing the API Server's certificate.

1. In the Policy Manager, select Tasks, Manage Certificates. The Manage Certificates window opens.
2. Click Add.
   The Add Certificate Wizard opens.
3. Complete one of the following:
   - If you created the API Server certificate on the same Gateway instance, select Import from Private Key's Certificate Chain, and then select the certificate (for example, `lacssl`).
   - If you created the API Server certificate on a different Gateway or using another certification tool, select Import from a File and browse to your previously exported certificate file (for example, `lacssl.pem`).
4. Click Next, and Next.
5. Select Outbound SSL Connections, and then click Next.
6. Select Certificate is a Trust Anchor, click Finish, and then click Close.

Configure API Server for Mutual Authentication

Configure API Server for SSL/HTTPS and mutual authentication with API Gateway by configuring the API Server keystore to use the API Server private and public keys, and API Gateway's public key. You can configure Jetty or Tomcat as the API Server's web servers. Use the following procedure based on the type of web server you use:

- Configure Jetty as the API Server web container.
- Configure Tomcat as the API Server web container.

Configure Jetty as the API Server Web Container

The Jetty package includes API Server with preconfigured SSL/HTTPS and a default keystore.

Use the following process to configure Jetty as the API Server web container:

1. Enable SSL and HTTPS modules.
2. (Optional) Change the default HTTPS ports.
3. Add API Gateway Server's public key to API Server's (Jetty) keystore.
4. Start Live API Creator.

Enable SSL and HTTPS Modules

1. Open the `<root Live API Creator installation directory>/CALiveAPICreator/start.ini` file.
2. Uncomment the `--module=https` and `--module=ssl` lines.
3. Save and close the file.

Change the Default HTTPS Ports

CA Live API Creator is configured to use the following HTTP/HTTPS ports:

- Port 8080 for API Creator web application access over HTTP
- Port 8083 for API Creator web application access over HTTPS
- Port 8081 for API Server communication with API Gateway

over HTTPS with mutual authentication enabled

You can change these default ports.

To change the default 8081 port, complete the following:

1. Open the `<root Live API Creator installation directory>/CALiveAPICreator/etc/jetty.xml` file.
2. Change the value for the `caGatewayLACPort` argument to a new preferred port number, save and close the `jetty.xml` file.
3. Open the `<root Live API Creator installation directory>/CALiveAPICreator/etc/jetty-https.xml` file.
4. Change the value for `https.port` property under httpsConnectorLAC connector to a new preferred port number, save and close the jetty-https.xml file

To change the default 8083 port, complete the following:

1. Open the `<root Live API Creator installation directory>/CALiveAPICreator/etc/jetty-https.xml` file.
2. Change the value for `https.port` property under httpsConnector connector to a new preferred port number, save and close the `jetty-https.xml` file.

Add API Gateway Server's Public Key to API Server's (Jetty) Keystore

1. Issue the following command:

```
keytool -importcert -file <your Gateway PEM file> -keystore <root LAC installation directory>/CALiveAPICreator/etc/keystore
```

   For example: `keytool -importcert -file /tmp/gateway1.pem -keystore /lac/CALiveAPICreator/etc/keystore)`
2. Enter the keystore password when prompted.
3. Trust the Gateway certificate by entering Yes.
   The certificate is successfully added to the Jetty keystore.

Start Live API Creator

From a command line, start Jetty by issuing the following command:

```
sh Start.sh
```

Jetty starts the three connectors on ports 8080, 8083, and 8081.

Configure Tomcat as the API Server Web Container

1. Add the API Server's private key to Tomcat's key store by completing the following:
    1. Issue the following command:

       ```
       keytool -importkeystore -srckeystore
       <your PKCS12 file> -srcstoretype
       PKCS12 -destkeystore <your Tomcat
       keystore>
       ```

       For example: `keytool -importkeystore -srckeystore /tmp/lacssl.p12 -srcstoretype PKCS12 -destkeystore /apache-tomcat-8.0.30/conf/keystore`
    2. Enter the destination keystore password when prompted. **Example:** Password1.
    3. Enter the source keystore password when prompted. **Example:** Password1.
       The API Server's private key is successfully imported into Tomcat's key store.
2. Add the Gateway server's public key to Tomcat's key store by issuing the following command:

   ```
   keytool -importcert -file <your Gateway PEM
   file> -keystore <your Tomcat keystore>
   ```

   For example: `keytool -importcert -file /tmp/gatewayssl.pem -keystore /apache-tomcat-8.0.30/conf/keystore`
3. Define a dedicated port for API Server and API Gateway to communicate by opening the `$CATALINA_BASE/conf/context.xml` file and adding the following environment entry:

```
<Environment name="caGatewayLACPort"
value="8081" type="java.lang.Integer"
override="false"/>
```

4. Configure an SSL/HTTPS connector by opening the `$CATALINA_BASE/conf/server.xml` file and adding the following XML fragment to the file:

```
<Connector

protocol="org.apache.coyote.http11.Http11NioP
port="8081" maxThreads="200"
scheme="https" secure="true"
SSLEnabled="true"
keystoreFile="conf/keystore"
keystorePass="Password1"
truststoreFile="conf/keystore"
truststorePass="Password1"
truststoreType="JKS"
clientAuth="true" sslProtocol="TLS"/>
```

5. Start Tomcat by issuing the `start.sh` script.

Install the Integration Solution Kit

The integration solution kit archive (SKAR) file serves as a reference-Gateway service to integrate API Server and API Gateway. Installing the solution kit creates the LiveAPICreator folder in API Gateway, and the LAC Publish Service service.

You can uninstall the solution kit when you no longer need it.

1. In the Policy Manager, click Tasks, Manage Solution Kits. The Manage Solution Kits window opens.
2. Click Install. The Solution Kit Installation Wizard opens.
3. Navigate to the SKAR file, click Open, click Next, then click Next again.
4. Verify that the content of the file has been imported and click Finish.
5. Verify that the solution kit has been added and then click Close.

Next Steps

Now that you have integrated Live API Creator with API Gateway, you can publish the API project to API Gateway. For more

information, see Publish an API Project to API Gateway.

SUBPAGES (4): CONFIGURE PUBLISHED API PROJECTS IN API GATEWAY CONSUME THE PUBLISHED API PROJECT IN API GATEWAY EXPOSE PUBLISHED API PROJECTS WITHIN API PORTAL PUBLISH AN API TO API GATEWAY