

 Search this site

GETTING STARTED

[VIDEOS](#)[WELCOME WALK-THROUGH](#)[BUSINESS TO BUSINESS EXAMPLE](#)[REACTIVE LOGIC TUTORIAL](#)[ARCHITECTURE](#)[QUICK REFERENCE](#)[TEAM DEVELOPMENT](#)[LANGUAGE EXAMPLES](#)[API PROJECT SAMPLES](#)

CA LIVE API CREATOR RELEASE NOTES

[SUPPORTED PLATFORMS](#)[THIRD-PARTY SOFTWARE](#)

INSTALL LIVE API CREATOR

[INSTALL AND USE THE ADMIN COMMAND LINE INTERFACE](#)[INSTALL AND USE THE COMMAND LINE INTERFACE](#)[Docs > Integrate with CA API Gateway >](#)

Publish an API to API Gateway

Publishing an API from Live API Creator (LAC) to API Gateway creates the API Gateway-managed service for that API.

An API is published to API Gateway using one of the following methods:

- The LAC administrator uses the LAC admin command line interface (CLI).
- The LAC developer uses the API Creator interface.
- The API Gateway administrator uses a cURL command in a command line.

API Server uses the LAC Publish Service service that was created when you installed the solution kit. When you publish an API from API Server, API Server uses this service to create a new API Gateway-managed service. This service represents the published API project. API Server adds this new service to the [LiveAPICreator/LAC Projects](#) folder in API Gateway.

Complete the following for every API project you want to publish to API Gateway.

CONTENTS

- [1 Verify Prerequisites](#)
- [2 Publish your API Project](#)
 - [2.1 Publish using Admin CLI](#)

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

2.2 Publish using Live API Creator

2.3 Publish using a cURL Command

3 Confirm that the API Gateway-Managed Service is Created

4 Next Steps

Verify Prerequisites

Ensure that you have completed the following prerequisites:

- You have set up mutual authentication between API Server and API Gateway.
- (For publishing using cURL commands) You have obtained the Swagger documentation URL of the API project you published to API Gateway and have downloaded the returned JSON response as a JSON file, for example, `demo.json`. For example: `https://liveapicreator.mycompany.com:8080/rest/c/auth=demo_full:1`.
- (For publishing using LAC admin CLI) You have installed the following:
 - **Node.js**. For more information about downloading Node.js, see the [Node.js site](#).
 - **The admin CLI**. For more information about installing the admin CLI, see [Install and Use the Admin Command Line Interface](#).

Publish your API Project

You can publish your API project using the following methods:

- Using the admin CLI.
- Using the API Creator interface.
- Using a cURL command.

Publish using Admin CLI

1. Log in to the API Server as an administrator using the

PUBLISHED API
PROJECTS IN API
GATEWAY

CONSUME THE
PUBLISHED API
PROJECT IN API
GATEWAY

EXPOSE PUBLISHED
API PROJECTS WITHIN
API PORTAL

admin CLI by issuing the following command:

```
lacadmin login -u <admin user name> -p
<password> <URL for the API Server
(Jetty or Tomcat)> -a <alias for the API
Server>
```

For example:

```
acadmin login -u admin -p
Password1 http://localhost:8080 -a
MyAPIServer
```

2. List and use the API project to publish by issuing the following commands:

```
lacadmin project list
lacadmin project use --url_name <API
project url name>
```

3. Publish the API project's Swagger doc to your API Gateway:

```
lacadmin gateway publish --username <API
Gateway admin user name> --password <API
Gateway admin password> --hostname
<protocol>://<API Gateway host name>:
<API Gateway port>/lacman/1.0/publish --
url_name <urlfragment> --apiversion
<version>
```

For example:

```
lacadmin gateway publish --username
admin --password Password1 --hostname
https://gateway1:8443/lacman/1.0/publish
--url_name demo --apiversion v1
```

4. Confirm that your API project has successfully published with API Gateway response 'OK'.

The API Gateway-managed service is created.

Publish using Live API Creator

You can publish the API Project's Swagger 2.0 documentation to the configured Gateway using LAC.

1. Log in to Live API Creator. If the Welcome screen opens, close it.
2. On the Home screen, click CA API Gateway.

3. Complete the following fields and then click Publish:

Gateway Name

Name this API Gateway connection.

API

Select your API from the drop-down.

URL

Enter `<protocol>://<API Gateway host name>:<API Gateway port>/lacman/1.0/publish`.

Example: `https://gateway1:8443/lacman/1.0/publish`.

API Version

Select your API version or accept the default.

Username

Enter the Gateway Admin user name.

Example: admin

Password

Enter the Gateway admin password.

Example: Password1.

A record of this API Gateway connection is added to the list of Gateways in LAC, the API project is published to API Gateway, and the API Gateway-managed service is created. In Policy Manager, the API project is created under the `LiveAPICreator/LAC Projects` folder.

Publish using a cURL Command

1. Issue the following command:

```
curl -kis -H "Content-Type: application/json" -X PUT -d @<your downloaded file> -u <API Gateway admin user>
```

```
name>:<API Gateway admin user password>  
<protocol>://<API Gateway host name>:<API  
Gateway port>/lacman/1.0/publish
```

For example:

```
curl -kis -H "Content-Type:  
application/json" -X PUT -d @demo.json -u  
admin:7layer  
https://gateway1:8443/lacman/1.0/publish
```

2. Confirm that your API project is successfully published with status code 200.

The API Gateway-managed service is created.

Confirm that the API Gateway-Managed Service is Created

1. In the Policy Manager, click Refresh.
2. Expand the LiveAPICreator/LAC Projects folder and confirm that the API Gateway-managed service is created. The name of the service is the name of the API project.

Next Steps

Now that you have published the API project to API Gateway, configure the published API project. For more information, see [Configure Published API Projects in API Gateway](#).