# Gateway and Live API Creator (LAC) Integration

## Description

This tutorial shows you how to integrate your gateway and a Live API Creator instance including how to apply your Live API Creator (LAC) license.  It will also show how to publish your LAC API project and have this service appear in policy manager on the gateway.  This tutorial will review how to retrieve the swagger for your LAC created service in order to test this in SoapUI.  It will also review the necessary user/role mappings for authentication of your API project when integrated with the gateway along with several other helpful hints.

## Prerequisites

### Environment – minimums:

1. CA API Gateway version 9.1 or later (should also work with 9.0)
2. CA API Management Policy Manager with a version matching the gateway.
3. CA Live API Creator version 3.0 or later.
4. The CA LAC Integration gateway Solution Kit found in the Jetty package.
5. SoapUI (use the free soapUI version 5.2.1 or later.)

### Environment – additional considerations if NOT using the presales combined image:

1. Mutual authentication must be established between the API Server (LAC) and the API Gateway.  Note that this has been done for you in the combined image using the default Jetty key.  Please refer to the LAC documentation for instructions.

### Environment – additional considerations if you ARE using the presales combined image:

1. Note that the Jetty default http port of 8080 has been changed to 8181 to avoid a conflict with the gateway in the combined image.  Please refer to the LAC documentation for further detail. This has been done for you in the combined image and explains the use of port 8181 in several screen shots used in this tutorial.
2. Note also that the change to the LAC start.ini file to enable SSL and HTTPS has also been done for you in the combined image.

### Tutorials or what you should know before you start:

1. Tutorials - Getting Started *or equivalent Policy Manager knowledge*.
2. Tutorial 1 - Deploy Tutorial Services *or equivalent Policy Manager familiarity*.
3. Tutorial 2 - Test Tutorial SOAP Service *or equivalent SoapUI familiarity*.
4. Live API Creator – knowledge to start the software and create an API Project.

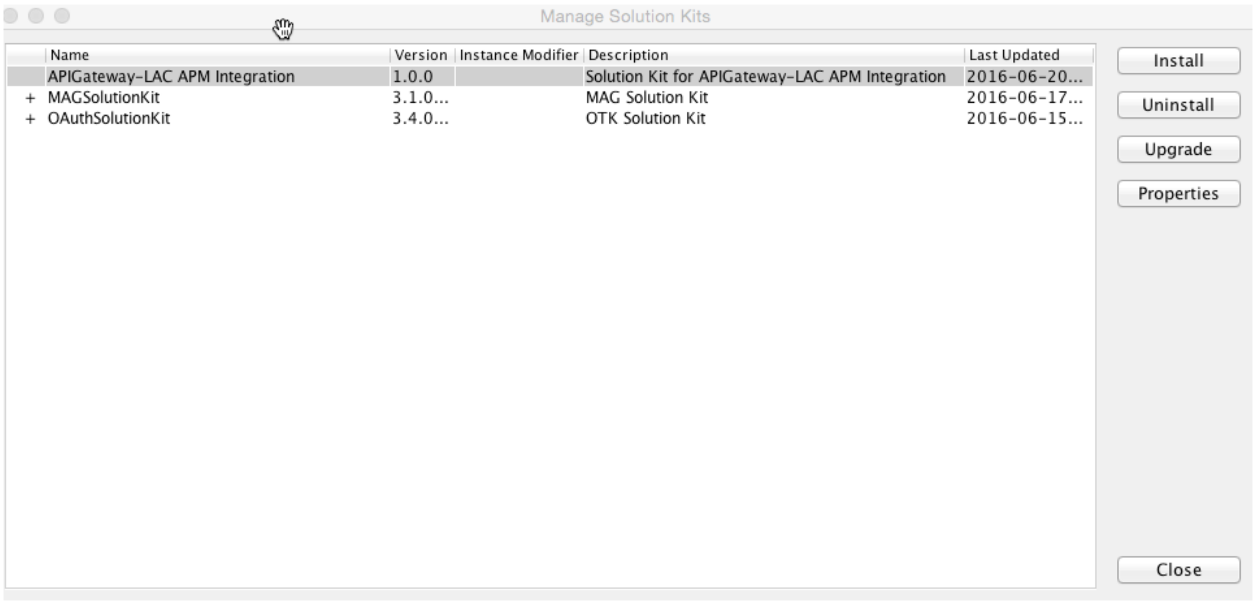
## Tutorial Steps Overview:

Gateway – Add/Verify solution kit to the gateway for Live API Creator
Live API Creator – Add/Verify license file
Live API Creator – Publish API and configure Gateway Settings
Live API Creator – Setup Roles
Live API Creator – How to retrieve the Swagger definition in order to test
Gateway – Customize default role mappings in policy to match LAC
Helpful Hints – which is which and policy customization considerations

## Tutorial Steps Detail:

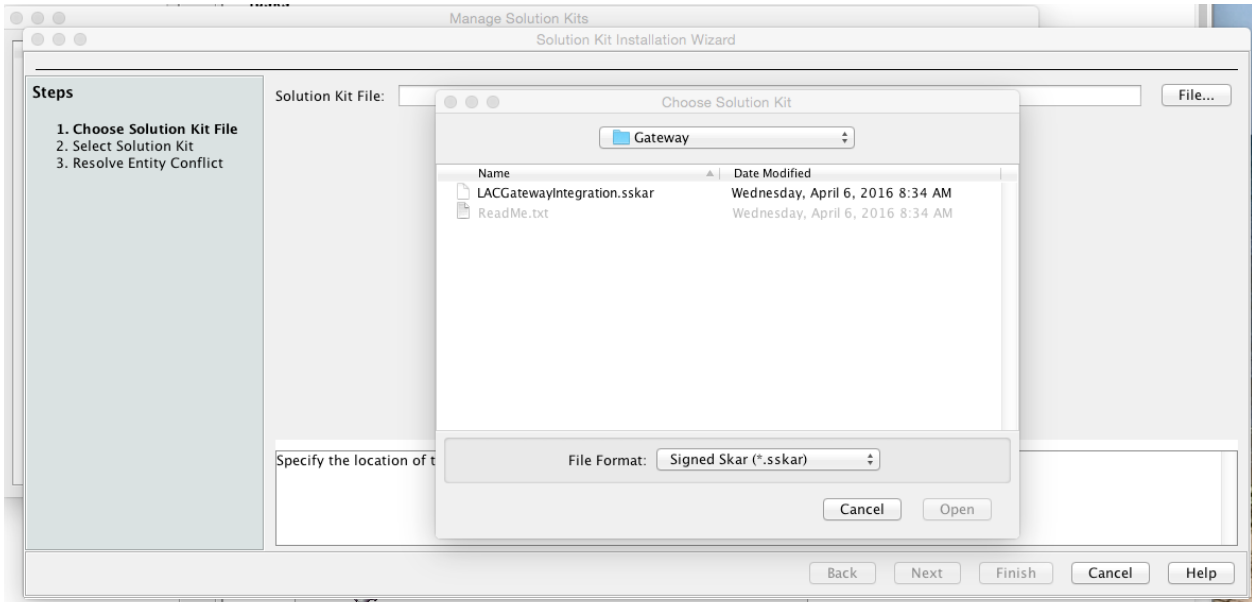
### Gateway – Add solution kit to the gateway for Live API Creator:

1. Connect to your gateway using Policy Manager (see Tutorials - Getting Started).
2. In Policy Manager, select the ***Tasks/Extensions and Add-ons/Manage Solution Kits*** menu item.
3. In the Manage Solution Kits dialog window, verify that the "APIGateway-LAC…" solution kit appears. (Note, this may already be in place if using the combined image.) If not, click ***Install***.



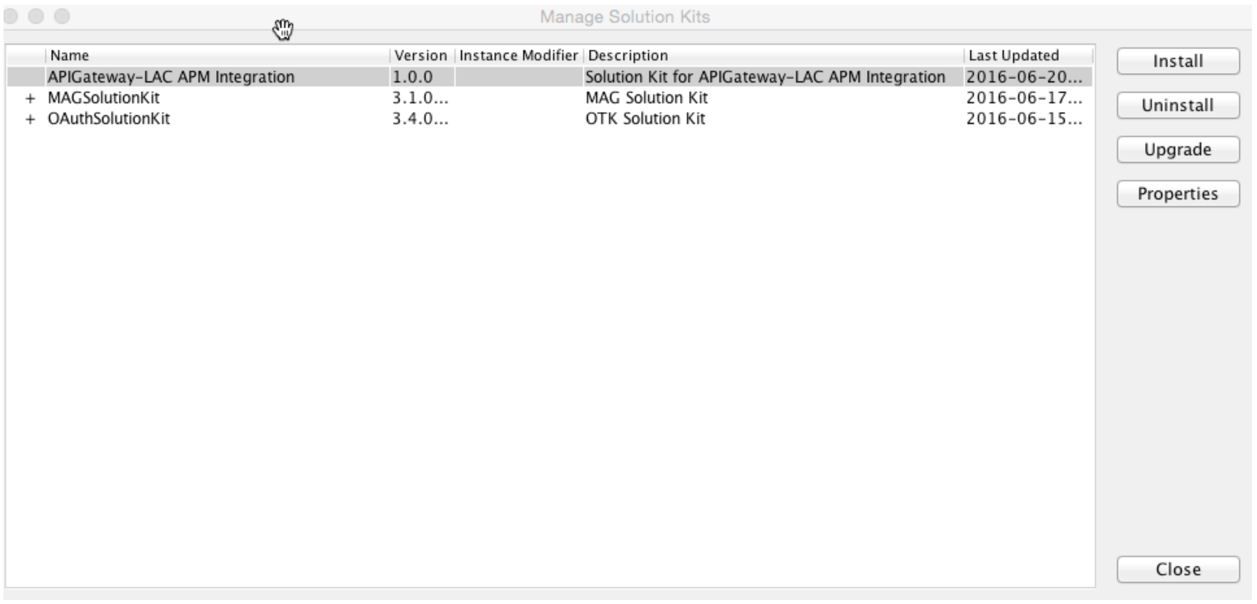
| Name | Version | Instance Modifier | Description | Last Updated |
|---|---|---|---|---|
| APIGateway-LAC APM Integration | 1.0.0 | | Solution Kit for APIGateway-LAC APM Integration | 2016-06-20… |
| + MAGSolutionKit | 3.1.0… | | MAG Solution Kit | 2016-06-17… |
| + OAuthSolutionKit | 3.4.0… | | OTK Solution Kit | 2016-06-15… |

4. From the Solution Kit Installation Wizard, click ***File***.
5. From the Choose Solution Kit window, browse to the location of the necessary file (LACGatewayIntegration.sskar), select it, and click ***Open***. Typically, the path will be <name_of_jetty_package_here>/Samples/Gateway/LACGatewayIntegration.sskar. See screen shot below:



| Name | Date Modified |
|---|---|
| LACGatewayIntegration.sskar | Wednesday, April 6, 2016 8:34 AM |
| ReadMe.txt | Wednesday, April 6, 2016 8:34 AM |

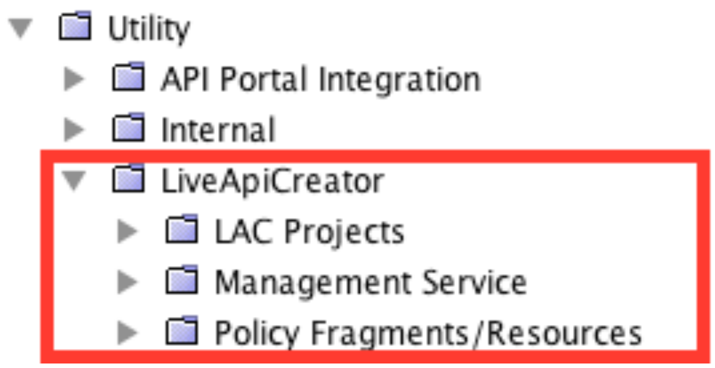
6. Click ***Next*** to resolve any conflicts, then click ***Finish***. Your solution kit will show in the Manage Solution Kits window.



| | Name | Version | Instance Modifier | Description | Last Updated |
|---|---|---|---|---|---|
| | APIGateway-LAC APM Integration | 1.0.0 | | Solution Kit for APIGateway-LAC APM Integration | 2016-06-20... |
| + | MAGSolutionKit | 3.1.0... | | MAG Solution Kit | 2016-06-17... |
| + | OAuthSolutionKit | 3.4.0... | | OTK Solution Kit | 2016-06-15... |

7. Click ***Close***. The LAC integration solution kit has been successfully applied to the gateway.

8. A new folder "**LiveAPICreator**" will be visible in the service and policy tree which will contain the policies for your LAC projects and the necessary integration services. Note, your higher level folders may vary from what is shown here.

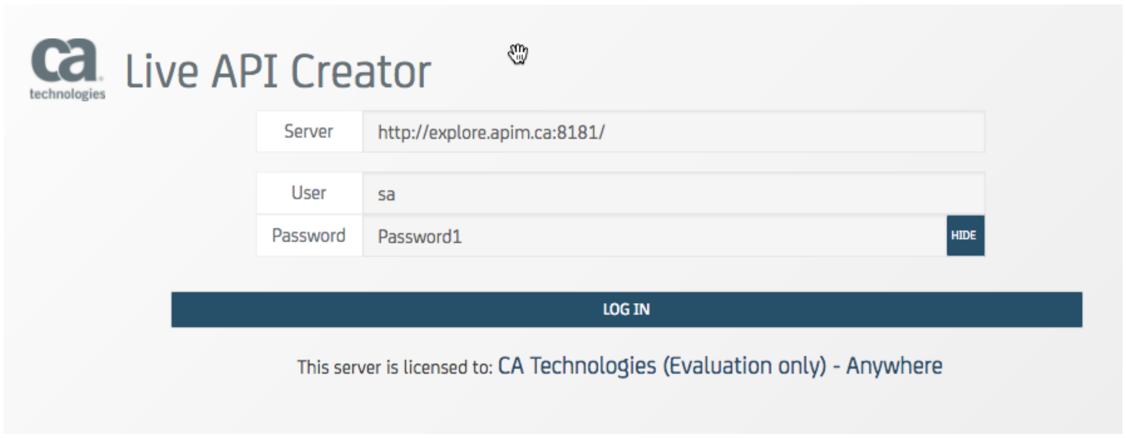


- Utility
  - API Portal Integration
  - Internal
  - LiveApiCreator
    - LAC Projects
    - Management Service
    - Policy Fragments/Resources

9. You are done with this first step on the gateway.

## Live API Creator – Add/Verify license file:

1. Connect to your own Live API Creator instance, or for the combined image use this URL: [explore.apim.ca:8181/APICreator/#/](explore.apim.ca:8181/APICreator/#/) using the credentials for User as "sa" and Password as "Password1" click ***Login***. Note that port 8181 is being used in place of the jetty default port 8080 to avoid a conflict with port 8080 when used by the gateway in the combined image. Refer to page 1, Environment Prerequisites in this document.

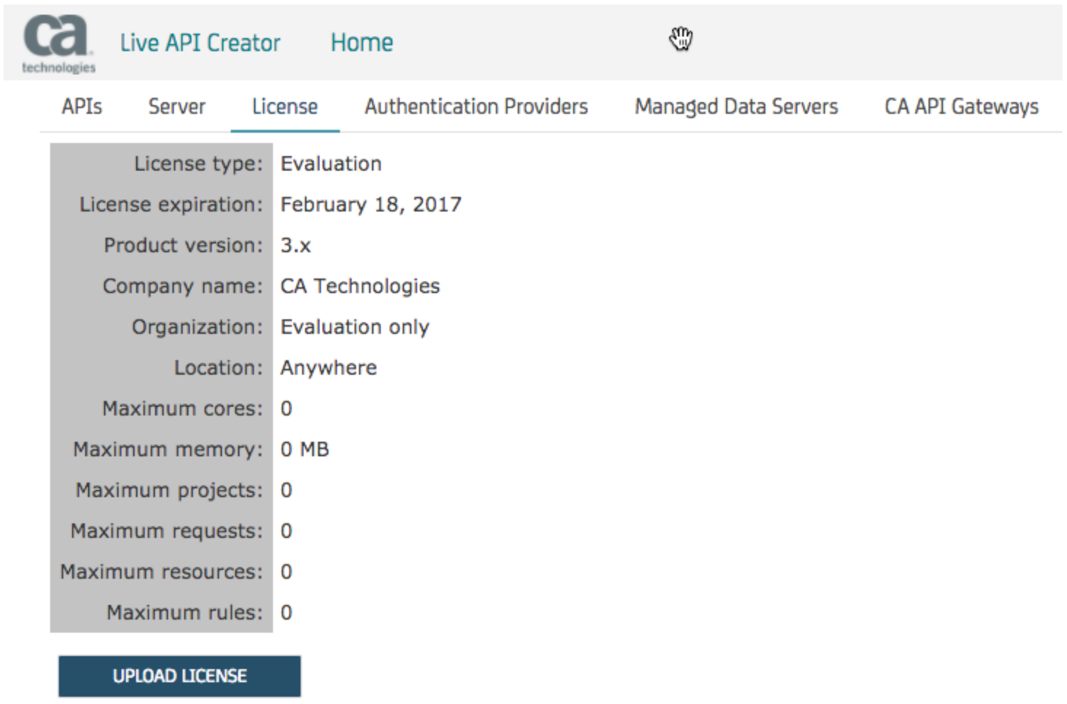


Live API Creator

| | |
|---|---|
| Server | http://explore.apim.ca:8181/ |
| User | sa |
| Password | Password1 |

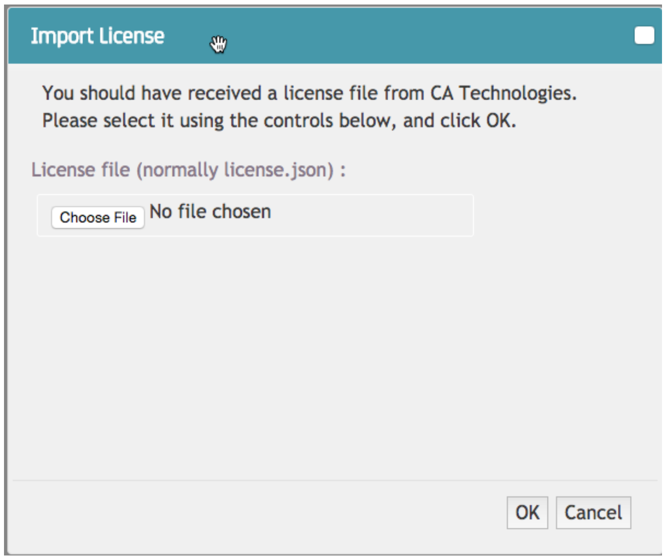
This server is licensed to: CA Technologies (Evaluation only) - Anywhere

2. You will get a warning about the only valid use of "sa" which can be ignored for now. Click ***Close*** on upper right corner of the welcome screen to reveal the Home page.
3. Click the ***License*** tab to reveal the status of the current license.



| | |
|---|---|
| License type: | Evaluation |
| License expiration: | February 18, 2017 |
| Product version: | 3.x |
| Company name: | CA Technologies |
| Organization: | Evaluation only |
| Location: | Anywhere |
| Maximum cores: | 0 |
| Maximum memory: | 0 MB |
| Maximum projects: | 0 |
| Maximum requests: | 0 |
| Maximum resources: | 0 |
| Maximum rules: | 0 |

4. If the license has expired or is nearing expiration, click ***Upload License*** to upload and apply a new license file. Note, when the license has already expired you will receive an immediate warning when attempting to log in and will be prompted to upload the license file at that time. Remember to use the user "sa."
5. Click ***Choose File***, select your license file, Click ***Open.*** Note, license files could be named something similar to "EvalLicense_YearMMDD.txt".
6. Click ***OK.***



You should have received a license file from CA Technologies.
Please select it using the controls below, and click OK.

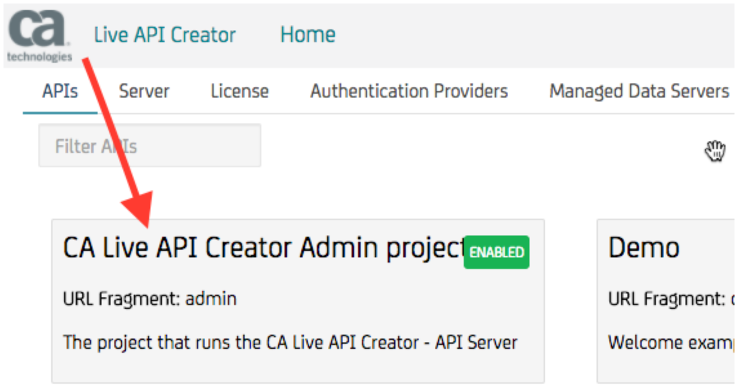
License file (normally license.json) :

7. The updated license expiration date will appear in the license tab as shown earlier. There is no need to restart the server.
8. Click the small "***gear" icon*** on the top right corner of the page and click ***Logoff***. For all other activities in this tutorial the regular "admin" user is sufficient.
9. You are finished with checking/uploading your LAC license file.

## Live API Creator – Publish API and Configure Gateway Settings:

1. ***Log in*** as admin. Close the welcome page if necessary.
2. Be sure you have already created your API project and it exists on your home page. (This is beyond the scope of this tutorial.) Note, if you see the "CA Live API Creator Admin project" as in the example below, you are logged in as the user "sa." Please log off and log in as "admin."

   **WRONG USER**: You should NOT see the CA Live API Creator Admin project.



CA Live API Creator Admin project

URL Fragment: admin

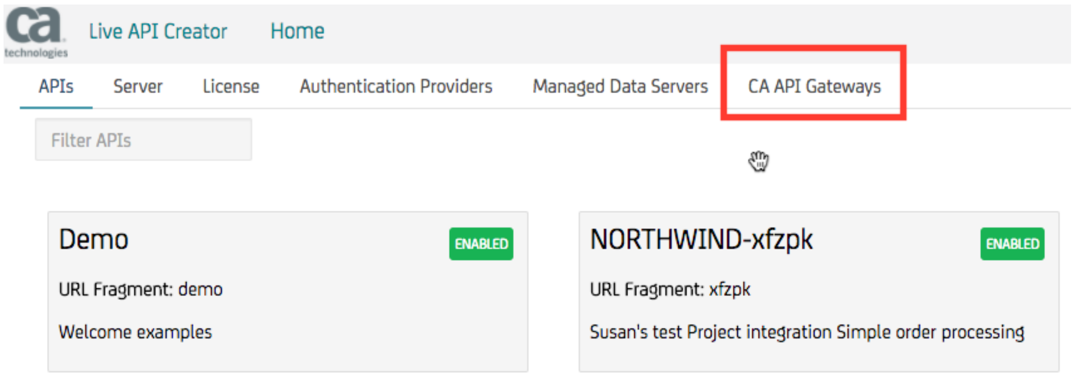
The project that runs the CA Live API Creator - API Server

Demo

URL Fragment:

Welcome exam

3. From the ***Home*** page as admin, click on the ***CA API Gateways*** tab.



Demo

URL Fragment: demo

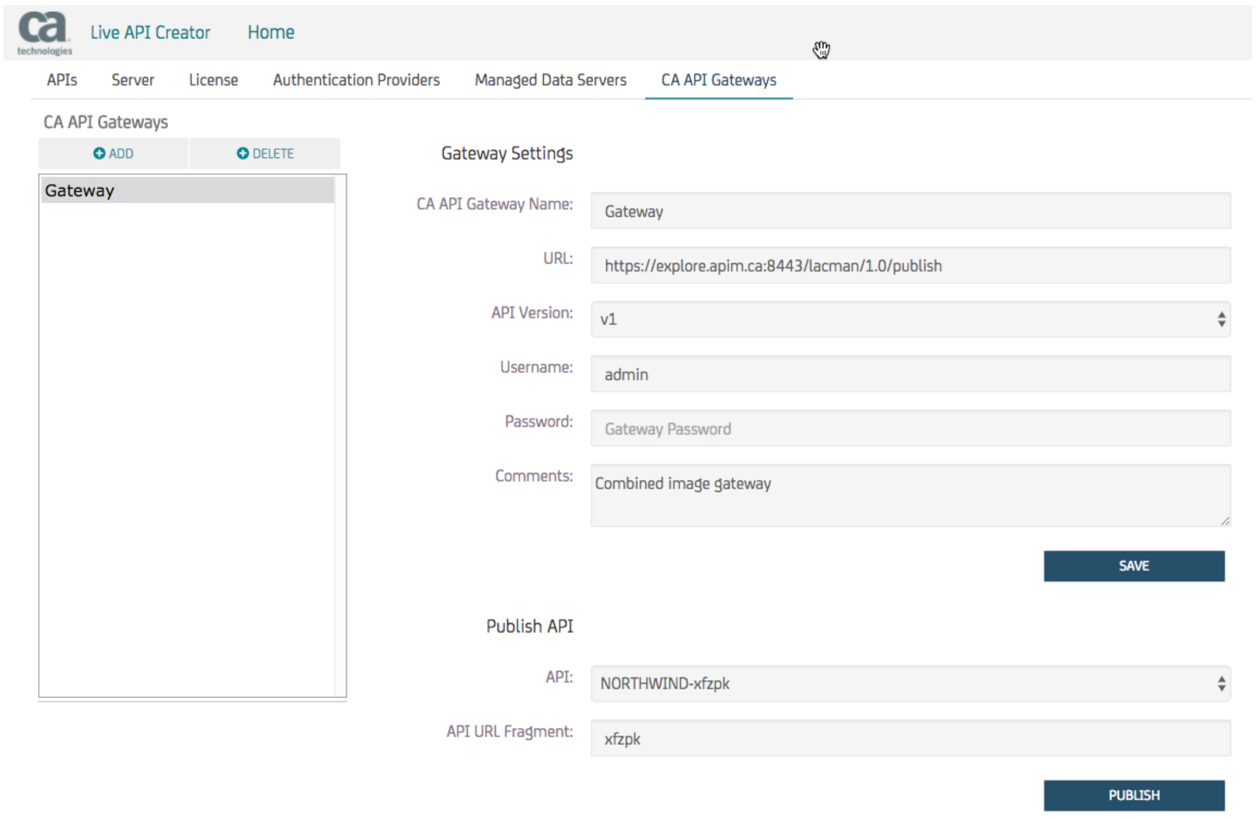
Welcome examples

NORTHWIND-xfzpk

URL Fragment: xfzpk

Susan's test Project integration Simple order processing

4. Enter your gateway settings. These are a name for your gateway, the URL (gateway, port, and integration service), the API version default will toggle based on your API choice in the section below.  Enter the admin user and password credentials for the gateway.  Comments are advised but optional.  Click ***Save***.



CA API Gateways

Gateway

Gateway Settings

CA API Gateway Name: Gateway

URL: https://explore.apim.ca:8443/lacman/1.0/publish

API Version: v1

Username: admin

Password: Gateway Password

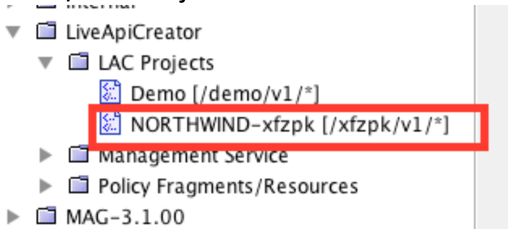
Comments: Combined image gateway

Publish API

API: NORTHWIND-xfzpk

API URL Fragment: xfzpk

5. Next, select from the API drop down list, your new LAC API (project name) and click ***publish***. In this example it is “NORTHWIND-xfzpk” as shown above.
6. You are finished publishing to the gateway.  Here is a recap of what happened: A record of this API Gateway connection is added to the list of Gateways in API Creator and the API is published to the API Gateway. The LAC API Server creates a new API Gateway-managed service in API Gateway and adds this new service to the LiveAPICreator/LAC Projects folder in Policy Manager for the gateway.
7. ***Refresh*** the **gateway Policy Manager** to see the new policy added to the “Live API Creator/LAC Projects” folder.



LiveApiCreator

LAC Projects

Demo [/demo/v1/*]

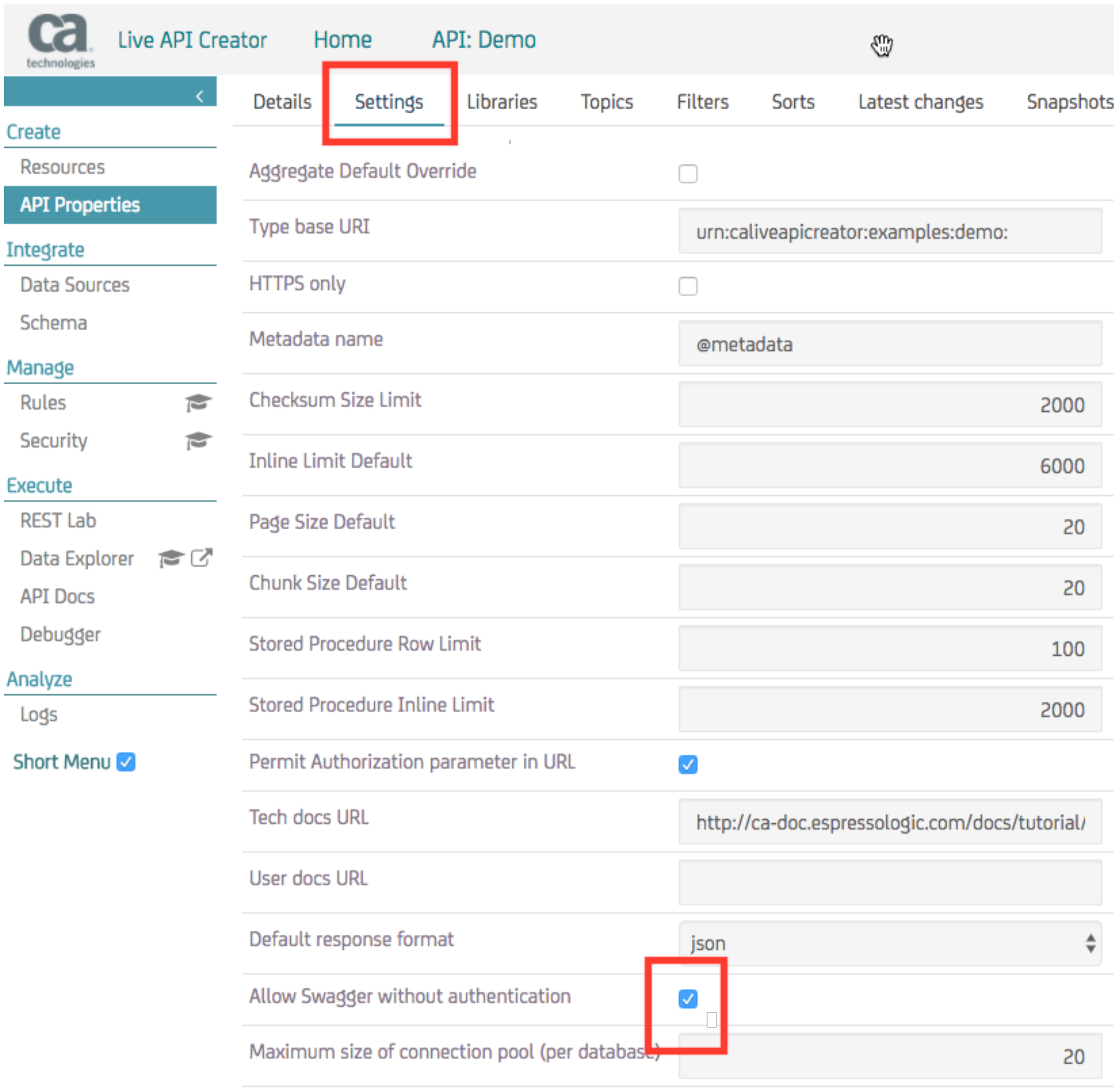
NORTHWIND-xfzpk [/xfzpk/v1/*]

Management Service

Policy Fragments/Resources

MAG-3.1.00

8. Click ***Home*** in LAC and select your API project you just published to the gateway. Click ***API Properties*** on the left menu, click ***Settings*** tab at the top of the panel. Click “Allow Swagger without Authentication” to enable. This will enable the retrieval of the swagger for the policy service.
9. Scroll down and click ***Save***.

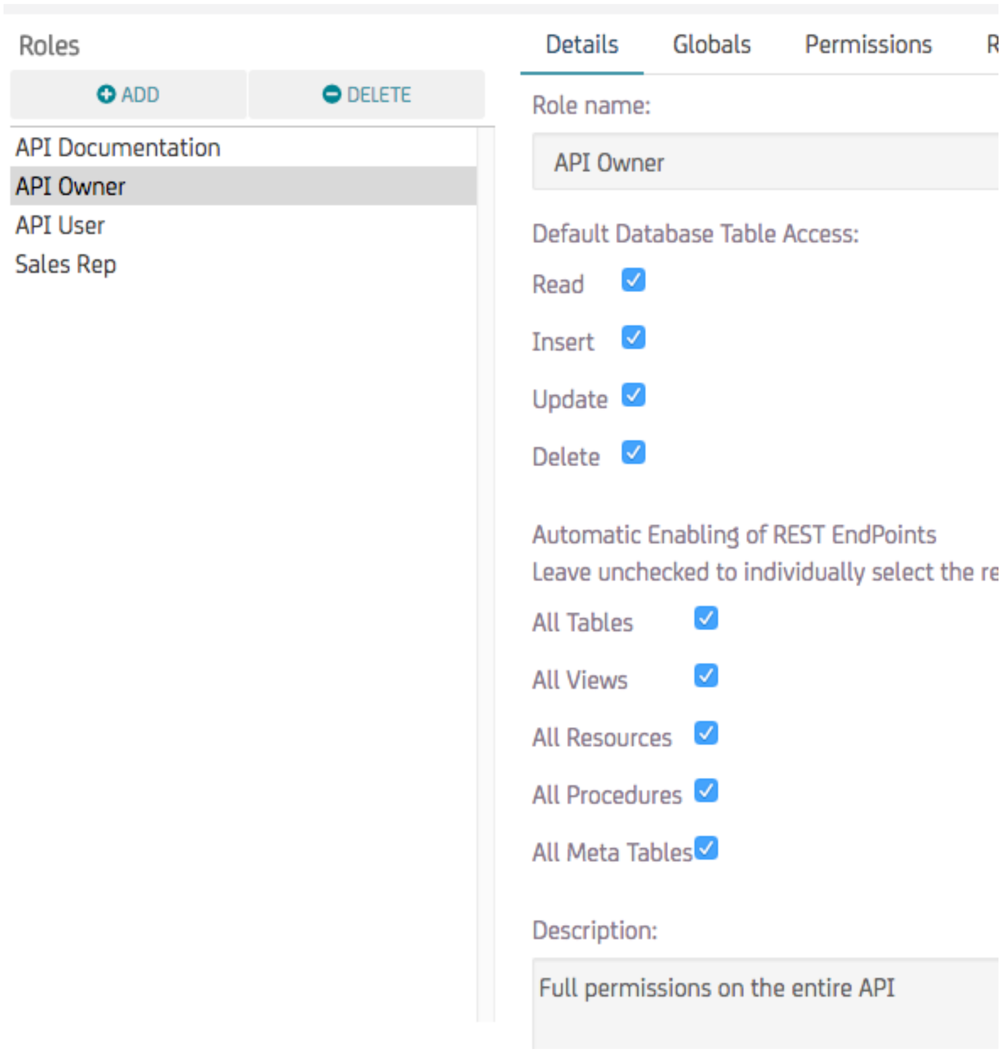


| | |
|---|---|
| Aggregate Default Override | ☐ |
| Type base URI | urn:caliveapicreator:examples:demo: |
| HTTPS only | ☐ |
| Metadata name | @metadata |
| Checksum Size Limit | 2000 |
| Inline Limit Default | 6000 |
| Page Size Default | 20 |
| Chunk Size Default | 20 |
| Stored Procedure Row Limit | 100 |
| Stored Procedure Inline Limit | 2000 |
| Permit Authorization parameter in URL | ✓ |
| Tech docs URL | http://ca-doc.espressologic.com/docs/tutorial/ |
| User docs URL | |
| Default response format | json |
| Allow Swagger without authentication | ✓ |
| Maximum size of connection pool (per database) | 20 |

## Live API Creator – Setup Roles:

You have two approaches here. On the LAC side you can setup roles to match the defaults in the solution kit or on the gateway side you can change the policy logic to match the roles you already have in LAC. Below are instructions for setting up the LAC roles. **Important: you must establish authorized roles for each API project.**

1. Click on ***Roles*** on the LAC menu on the left for your API project. You may need to ***uncheck Short Menu*** to see the Roles menu option.
2. Click ***Add*** and type "**API Owner**" as the name with ***full permissions*** (check all boxes). Supply a description.
3. Click ***Save***.



Role name:

API Owner

Default Database Table Access:

Read ✓

Insert ✓

Update ✓

Delete ✓

Automatic Enabling of REST EndPoints

Leave unchecked to individually select the re

All Tables ✓

All Views ✓

All Resources ✓

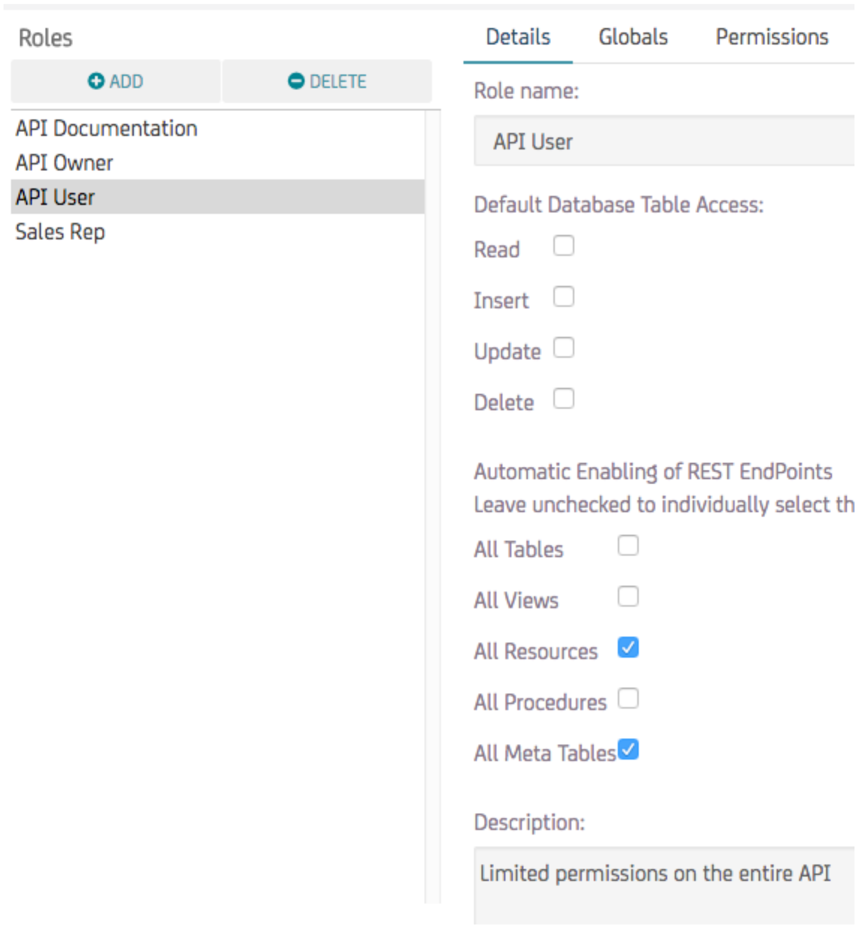
All Procedures ✓

All Meta Tables ✓

Description:

Full permissions on the entire API

4. Click ***Add*** and type "**API User**" as the name with limited permissions by only enabling: ***all resources***, ***all meta tables***. Supply a description for this role.
5. Click ***Save***.



Role name:

API User

Default Database Table Access:

Read ☐

Insert ☐

Update ☐

Delete ☐

Automatic Enabling of REST EndPoints

Leave unchecked to individually select th

All Tables ☐

All Views ☐

All Resources ✓

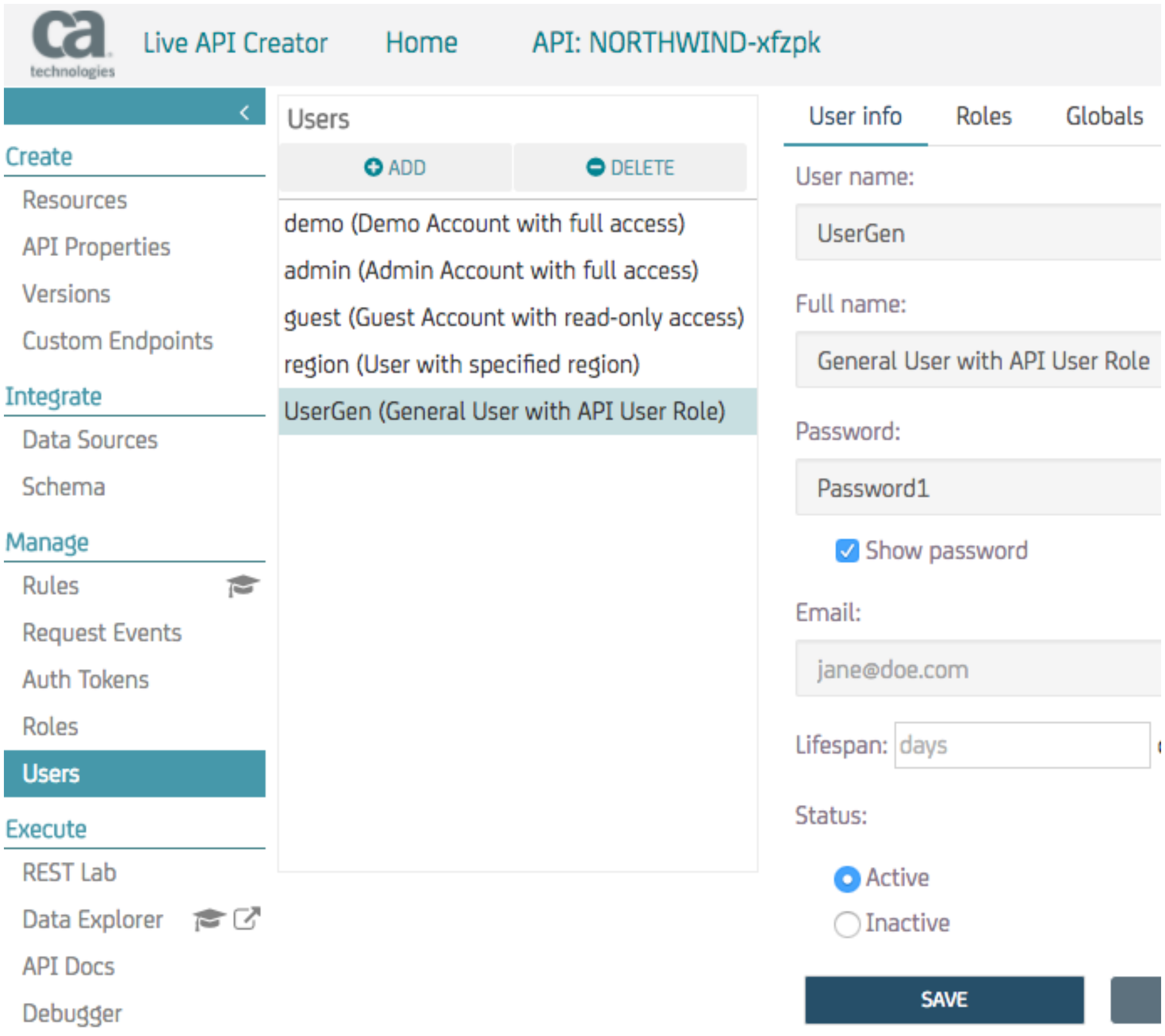
All Procedures ☐

All Meta Tables ✓

Description:

Limited permissions on the entire API

6. Create a user and assign the API User role to that new user. Click ***Users*** from left menu, Click ***Add***. Change name to "UserGen" for example. Type in the full name, password, email, and number of days for lifespan. Click ***Save***. See image below.



Users

demo (Demo Account with full access)
admin (Admin Account with full access)
guest (Guest Account with read-only access)
region (User with specified region)
UserGen (General User with API User Role)

User name: UserGen

Full name: General User with API User Role

Password: Password1

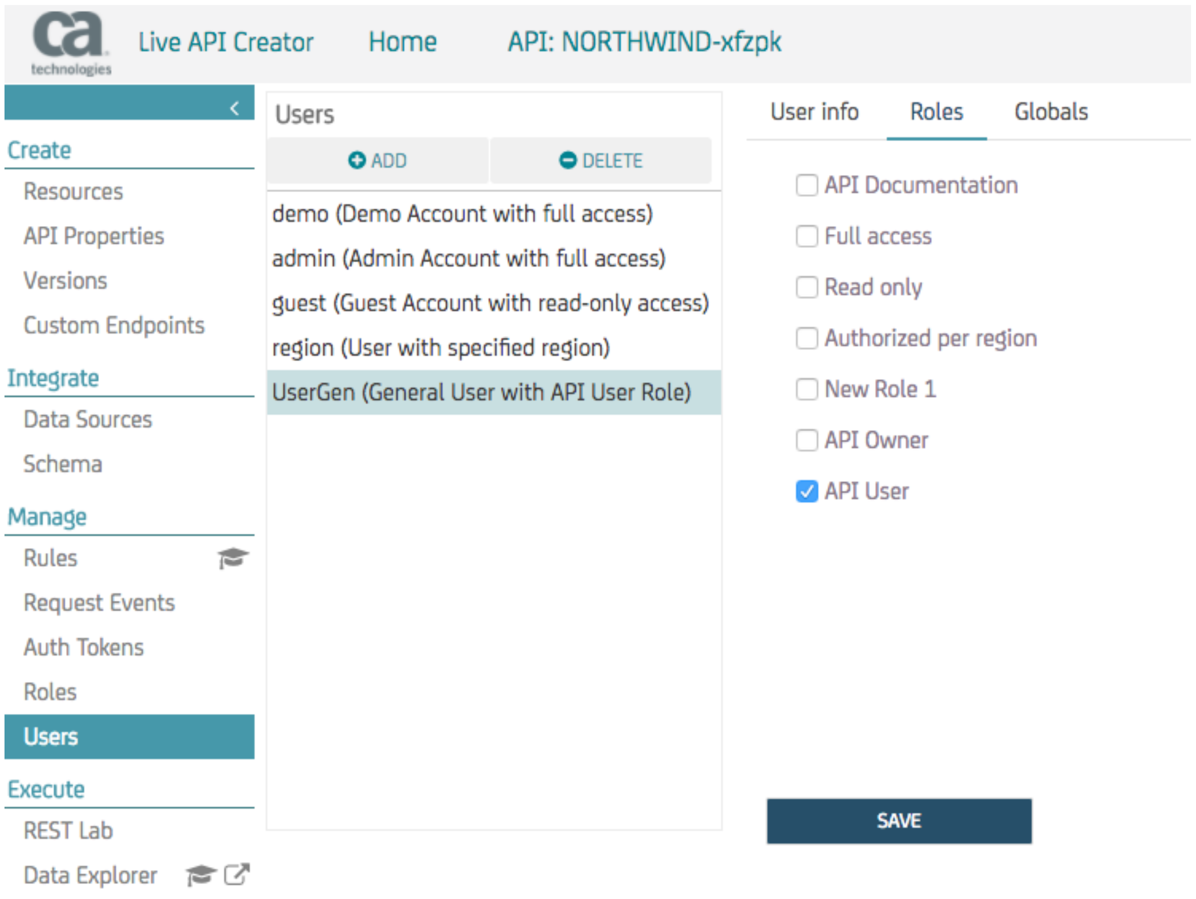
[✓] Show password

Email: jane@doe.com

Lifespan: days

Status: [● Active] Inactive

7. Click the ***Roles*** tab. Enable the role created above (“API User” role ) for this new user and ***Save***. See image below.



Users

- demo (Demo Account with full access)
- admin (Admin Account with full access)
- guest (Guest Account with read-only access)
- region (User with specified region)
- UserGen (General User with API User Role)

Roles:

- [ ] API Documentation
- [ ] Full access
- [ ] Read only
- [ ] Authorized per region
- [ ] New Role 1
- [ ] API Owner
- [✓] API User

8. You can now test using the Rest lab within Live API Creator OR by retrieving the Swagger definition and using SoapUI.

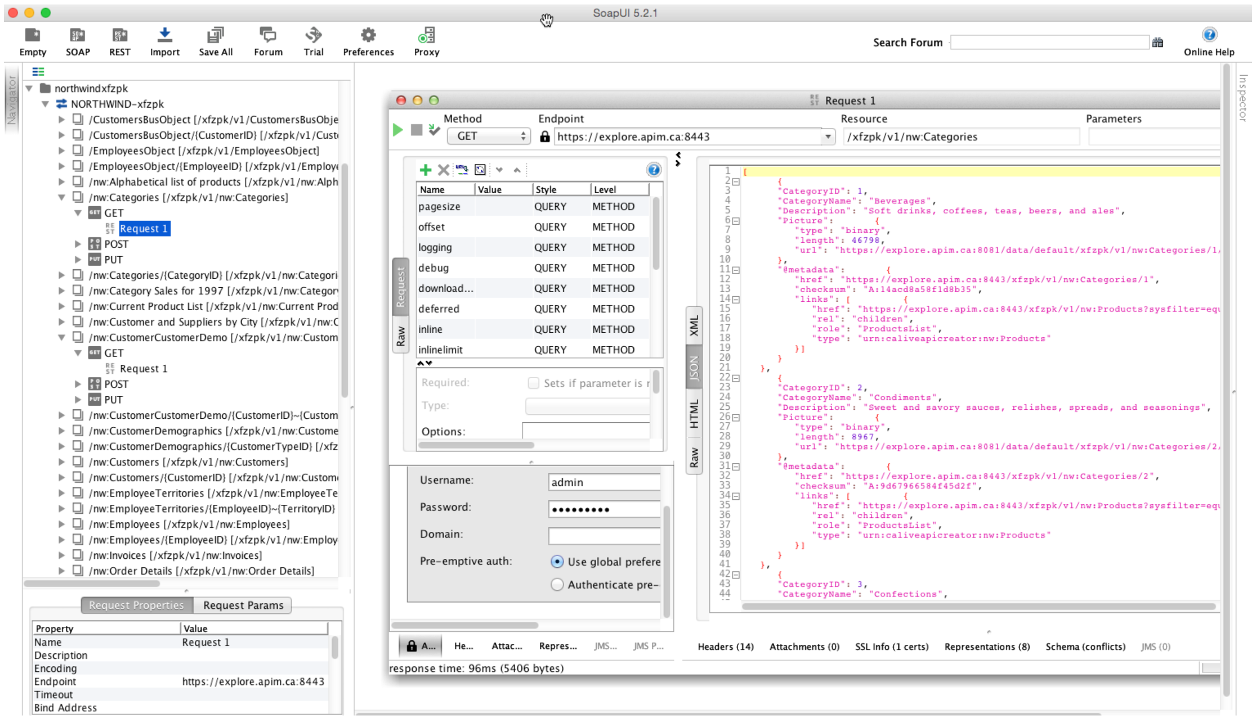
## Live API Creator – How to retrieve the Swagger definition in order to test from SoapUI:

1. Paste the following into a browser and go:

   https://explore.apim.ca:8443/<uriname>/v1/@docs
   for which the service name on the gateway Policy Manager looks like /<uriname>/v1/*.

2. Enter the gateway admin user and password when prompted.
3. When the page appears, click ***view page source*** or try step 4 below first.
4. Click ***File, Save as*** and place on your desktop as "northwind<uriname>.json" file.
5. Go into SoapUI, create a new REST project for your LAC API.
6. Click ***Project***, click ***Import Swagger*** and use the json file saved to your desktop.

7. Once imported, select an API from the list and open a Get, Request 1. Hint: Use an API that will return a short list such as categories and not the entire customer file.
8. Click ***Auth*** tab, bottom left and add basic with gateway credentials user = admin, password = CAdemo123, and domain left blank assuming you are using the combined image.
9. ***Submit*** the request. Change the results view from xml to Json.
10. You are finished with testing.



```
northwindxfzpk
  NORTHWIND-xfzpk
    /CustomersBusObject [/xfzpk/v1/CustomersBusObje
    /CustomersBusObject/{CustomerID} [/xfzpk/v1/Custo
    /EmployeesObject [/xfzpk/v1/EmployeesObject]
    /EmployeesObject/{EmployeeID} [/xfzpk/v1/Employe
    /nw:Alphabetical list of products [/xfzpk/v1/nw:Alph
    /nw:Categories [/xfzpk/v1/nw:Categories]
      GET
        Request 1
      POST
      PUT
    /nw:Categories/{CategoryID} [/xfzpk/v1/nw:Categori
    /nw:Category Sales for 1997 [/xfzpk/v1/nw:Categor
    /nw:Current Product List [/xfzpk/v1/nw:Current Prod
    /nw:Customer and Suppliers by City [/xfzpk/v1/nw:C
    /nw:CustomerCustomerDemo [/xfzpk/v1/nw:Custom
      GET
        Request 1
      POST
      PUT
    /nw:CustomerCustomerDemo/{CustomerID}~{Custom
    /nw:CustomerDemographics [/xfzpk/v1/nw:Custome
    /nw:CustomerDemographics/{CustomerTypeID} [/xfz
    /nw:Customers [/xfzpk/v1/nw:Customers]
    /nw:Customers/{CustomerID} [/xfzpk/v1/nw:Custom
    /nw:EmployeeTerritories [/xfzpk/v1/nw:EmployeeTe
    /nw:EmployeeTerritories/{EmployeeID}~{TerritoryID}
    /nw:Employees [/xfzpk/v1/nw:Employees]
    /nw:Employees/{EmployeeID} [/xfzpk/v1/nw:Employ
    /nw:Invoices [/xfzpk/v1/nw:Invoices]
    /nw:Order Details [/xfzpk/v1/nw:Order Details]
```

Request Properties | Request Params

| Property | Value |
|---|---|
| Name | Request 1 |
| Description | |
| Encoding | |
| Endpoint | https://explore.apim.ca:8443 |
| Timeout | |
| Bind Address | |

Request 1

| Method | Endpoint | Resource | Parameters |
|---|---|---|---|
| GET | https://explore.apim.ca:8443 | /xfzpk/v1/nw:Categories | |

| Name | Value | Style | Level |
|---|---|---|---|
| pagesize | | QUERY | METHOD |
| offset | | QUERY | METHOD |
| logging | | QUERY | METHOD |
| debug | | QUERY | METHOD |
| download... | | QUERY | METHOD |
| deferred | | QUERY | METHOD |
| inline | | QUERY | METHOD |
| inlinelimit | | QUERY | METHOD |

Required: Sets if parameter is r
Type:
Options:

Username: admin
Password: •••••••••
Domain:
Pre-emptive auth: Use global prefere / Authenticate pre-

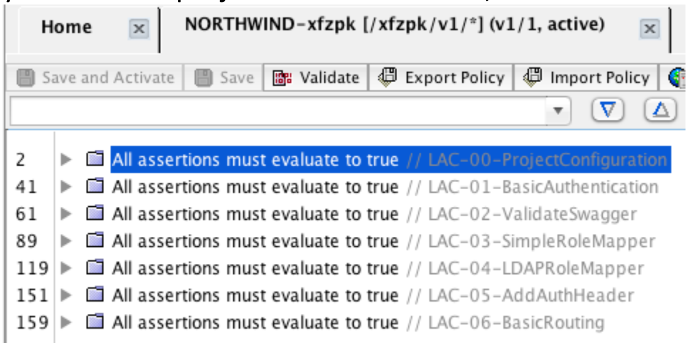
```
[
    {
    "CategoryID": 1,
    "CategoryName": "Beverages",
    "Description": "Soft drinks, coffees, teas, beers, and ales",
    "Picture":       {
      "type": "binary",
      "length": 46798,
      "url": "https://explore.apim.ca:8081/data/default/xfzpk/v1/nw:Categories/1,
    },
    "@metadata":       {
      "href": "https://explore.apim.ca:8443/xfzpk/v1/nw:Categories/1",
      "checksum": "A:14acd8a58f1d8b35",
      "links": [        {
        "href": "https://explore.apim.ca:8443/xfzpk/v1/nw:Products?sysfilter=equ
        "rel": "children",
        "role": "ProductsList",
        "type": "urn:caliveapicreator:nw:Products"
      }]
    }
  },
    {
    "CategoryID": 2,
    "CategoryName": "Condiments",
    "Description": "Sweet and savory sauces, relishes, spreads, and seasonings",
    "Picture":       {
      "type": "binary",
      "length": 8967,
      "url": "https://explore.apim.ca:8081/data/default/xfzpk/v1/nw:Categories/2,
    },
    "@metadata":       {
      "href": "https://explore.apim.ca:8443/xfzpk/v1/nw:Categories/2",
      "checksum": "A:9d67966584f45d2f",
      "links": [        {
        "href": "https://explore.apim.ca:8443/xfzpk/v1/nw:Products?sysfilter=equ
        "rel": "children",
        "role": "ProductsList",
        "type": "urn:caliveapicreator:nw:Products"
      }]
    }
  },
    {
    "CategoryID": 3,
    "CategoryName": "Confections",
```

A... He... Attac... Repres... JMS... JMS P... Headers (14) Attachments (0) SSL Info (1 certs) Representations (8) Schema (conflicts) JMS (0)

response time: 96ms (5406 bytes)

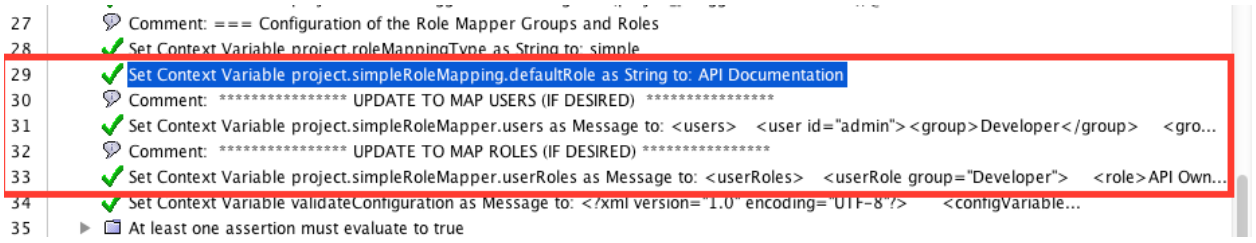
## Gateway – Customize default role mappings in policy to match LAC:

Alternatively, if you have roles already created in LAC for your API Project that you prefer to use, here are the minimum places which need to be changed in the policy logic on the gateway for the proxy. You will find these lines in the first section of policy logic that corresponds to your LAC API project. Section LAC-00, line 2:



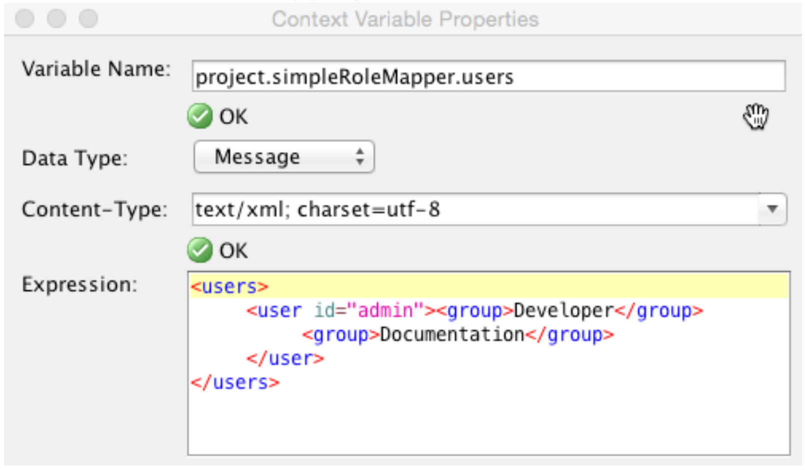
```
2    All assertions must evaluate to true // LAC-00-ProjectConfiguration
41   All assertions must evaluate to true // LAC-01-BasicAuthentication
61   All assertions must evaluate to true // LAC-02-ValidateSwagger
89   All assertions must evaluate to true // LAC-03-SimpleRoleMapper
119  All assertions must evaluate to true // LAC-04-LDAPRoleMapper
151  All assertions must evaluate to true // LAC-05-AddAuthHeader
159  All assertions must evaluate to true // LAC-06-BasicRouting
```

line 29 sets the default



```
27  Comment: === Configuration of the Role Mapper Groups and Roles
28  Set Context Variable project.roleMappingType as String to: simple
29  Set Context Variable project.simpleRoleMapping.defaultRole as String to: API Documentation
30  Comment: **************** UPDATE TO MAP USERS (IF DESIRED) ****************
31  Set Context Variable project.simpleRoleMapper.users as Message to: <users>   <user id="admin"><group>Developer</group>   <gro...
32  Comment: **************** UPDATE TO MAP ROLES (IF DESIRED) ****************
33  Set Context Variable project.simpleRoleMapper.userRoles as Message to: <userRoles>   <userRole group="Developer">   <role>API Own...
34  Set Context Variable validateConfiguration as Message to: <?xml version="1.0" encoding="UTF-8"?>   <configVariable...
35  At least one assertion must evaluate to true
```

line 31 establishes groups but notice the group "Support" is not included. You may wish to add it here for the role mapping below.



Variable Name: project.simpleRoleMapper.users

OK

Data Type: Message

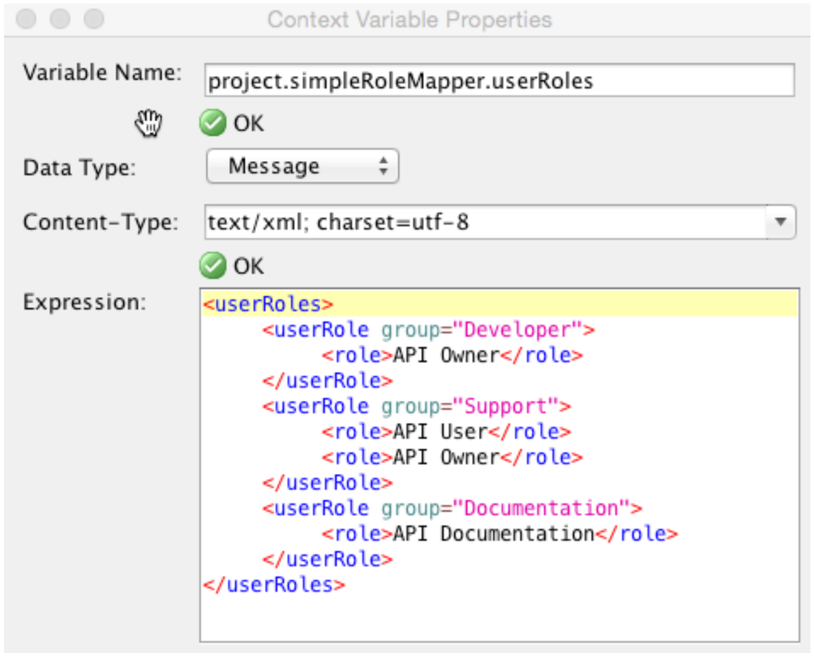
Content-Type: text/xml; charset=utf-8

OK

Expression:

```
<users>
    <user id="admin"><group>Developer</group>
        <group>Documentation</group>
    </user>
</users>
```

line 33, note "API Owner" and "API User" we created for our LAC API Project in previous steps.



Variable Name: project.simpleRoleMapper.userRoles

OK

Data Type: Message

Content-Type: text/xml; charset=utf-8

OK

Expression:

```
<userRoles>
    <userRole group="Developer">
        <role>API Owner</role>
    </userRole>
    <userRole group="Support">
        <role>API User</role>
        <role>API Owner</role>
    </userRole>
    <userRole group="Documentation">
        <role>API Documentation</role>
    </userRole>
</userRoles>
```

## Helpful Hints – which is which and policy customization considerations:

**Policy customization and configuration**:
If you expand the fragments/resources subfolder in the gateway policy manager, you will see at least six fragments. The lacman service will copy and insert these into each published LAC API policy on the gateway. As such, if you prefer to modify the default fragment for all future published policies, this would be the place to make the customization as opposed to the individual policy as shown above.

When not using the combined image, pay particular attention to the default host names as these may need to be changed in the policy or in the policy fragment used by lacman in order to route properly.

**Which is which:**
When using the combined image, note the following syntax to be sure you are calling the endpoint you expect when testing:

Client call which goes to the gateway:
https://explore.apim.ca:8443/xfzpk/v1/nw:categories

gateway URL and resolution path:
https:/explore.apim.ca:8443/xfzpk/v1/*

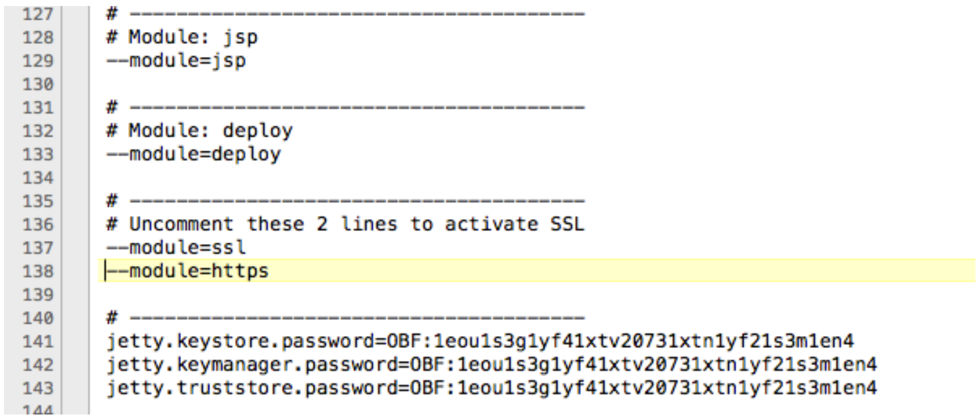
LAC restlab:
http://explore.apim.ca:8181/rest/default/xfzpk/v1/nw:categories

Note: policy on gateway will insert the "rest/default/" to determine URL for service on the LAC API server. Also remember the standard port 8080 on the LAC API server was changed to 8181 for the combined image.

**Modification to the LAC Start.ini file to support SSL:**

Hint: when using the combined image the LAC start.ini file can be found in this path:
/opt/LAC/CALiveAPICreator/start.ini

The image below shows the two lines that need to be uncommented to enable SSL and HTTPS modules as per the instructions. This has been done for you in the combined image.



```
# ---------------------------------------
# Module: jsp
--module=jsp

# ---------------------------------------
# Module: deploy
--module=deploy

# ---------------------------------------
# Uncomment these 2 lines to activate SSL
--module=ssl
--module=https

# ---------------------------------------
jetty.keystore.password=OBF:1eou1s3g1yf41xtv20731xtn1yf21s3m1en4
jetty.keymanager.password=OBF:1eou1s3g1yf41xtv20731xtn1yf21s3m1en4
jetty.truststore.password=OBF:1eou1s3g1yf41xtv20731xtn1yf21s3m1en4
```