**Getting Started with JBOSS APIman**

Table of Contents

[Overview of APIman 2](#_Toc16278004)

[Key Features 2](#_Toc16278005)

[Core components 2](#_Toc16278006)

[Installing APIman in WildFly 10 3](#_Toc16278007)

[Hardware and software requirements 3](#_Toc16278008)

[Overview of Echo API 4](#_Toc16278009)

[Invoking an Echo service 4](#_Toc16278010)

[Configuring an API Provider 5](#_Toc16278011)

[Creating a new plan 7](#_Toc16278012)

[Adding a new policy 8](#_Toc16278013)

[Defining an API 9](#_Toc16278014)

[Configuring an API Consumer Organization 13](#_Toc16278015)

[Creating a new organization 13](#_Toc16278016)

[Creating a new application 13](#_Toc16278017)

[Creating a contract 15](#_Toc16278018)

[Registering the application with the API Gateway 15](#_Toc16278019)

[Accessing the API through the API Gateway 16](#_Toc16278020)

[Testing a user defined policy 19](#_Toc16278021)

[Glossary 19](#_Toc16278022)

# Overview of APIman

JBoss APIman is an open source Application Programming Interface (API) manager, which aides in effectively creating, controlling and monitoring APIs through its set of subroutines or policies. Managing APIs in an effective manner is vital as APIs serve as tools for developing various types of software applications and ensuring safe data transfer across cloud environments.

## Key Features

Some of the key features of APIman are listed below,

* Enables governing APIs through policy-based runtime requests
* Allows managing and configuring the APIs using a REST API or a separate UI
* Provides a fully asynchronous runtime engine, which can run equally in both a standard Java EE environment and newer asynchronous runtimes like Vert.X
* Allows API developers to build and implement their own customized policies in the API manager with its extensible plug in-based architecture. In addition, based on the requirements, the architecture allows implementing the APIman components separately without creating a demand for building the core system

## Core components

The two main components of APIman are listed below

API Manager

The APIman’s web user interface enables API providers to easily configure the policies that mainly govern access to APIs and limit the rate at which the users can access the APIs. In addition, the interface allows the providers to use these policies across different APIs and control role-based user access and API versioning in a business environment.

API Gateway

The API gateway acts as a point of entry between the APIs and the defined group of microservices. The main function of the gateway is to apply the policies that are configured in the API manager. These policies act as runtime rules when requests are made to access APIs.

# Installing APIman in WildFly 10

Follow the procedures provided below to download, install and run the APIman in the WildFly 10 environment.

## Hardware and software requirements

Table 1 Hardware and software requirements to install APIman

|  |  |
| --- | --- |
| Software | Java 1.8 or newer |
| Gateway | Vert.X |
| Server | WildFly 10.1.0 Final |
| APIman overlay distribution | apiman 1.2.9 Final |

Download

It is recommended to download **WildFly 10.1.0 Final** and [Download **apiman 1.2.9.Final**](http://downloads.jboss.org/apiman/1.2.9.Final/apiman-distro-wildfly10-1.2.9.Final-overlay.zip) to install APIman.

Launch the command line and run the following code to download **WildFly 10.1.0**.

curl [http://download.jboss.org/wildfly/10.1.0.Final/wildfly-10.1.0.Final.zip -o wildfly-10.1.0.Final.zip](http://download.jboss.org/wildfly/10.1.0.Final/wildfly-10.1.0.Final.zip%20-o%20wildfly-10.1.0.Final.zip)

Launch the command line and run the following code to download **apiman 1.2.9.Final**

curl [http://downloads.jboss.org/apiman/1.2.9.Final/apiman-distro-wildfly10-1.2.9.Final-overlay.zip -o apiman-distro-wildfly10-1.2.9.Final-overlay.zip](http://downloads.jboss.org/apiman/1.2.9.Final/apiman-distro-wildfly10-1.2.9.Final-overlay.zip%20-o%20apiman-distro-wildfly10-1.2.9.Final-overlay.zip)

Install

Follow the procedure provided below to install APIman using the **WildFly 10.1.0** and **apiman 1.5.1 Final** overlay

1. Unzip both **WildFly 10.1.0** Zip file and **apiman 1.2.9 Final** overlay inside the **WildFly** directory

Syntax

unzip wildfly-10.1.0.Final.zip

unzip -o apiman-distro-wildfly10-1.2.9.Final-overlay.zip -d wildfly-10.1.0.Final

Run

1. Run **WildFly 10.1.0** using the standalone-apiman.xml configuration

Syntax

cd jboss-eap\*./bin/standalone.sh -c standalone-apiman.xml

The **APIman** user interface opens

1. Log in to the **APIman** user interface using the following login credentials

**User name:** admin

**Password:** admin 123

**Important:** It is recommended to change the default admin password after logging in to the Interface.

# Overview of Echo API

The Echo API is a service used by the APIs accessed by public (for SAAS). The Echo simply accepts any event and returns information (path, request parameters, headers, etc.) on the request made.

## Invoking an Echo service

Any request made to the echo service will respond with a 200 response code and a JSON response body that includes all of the meta-data sent to the echo service in the request. In general, the request method, path, and headers are all wrapped into a JSON object and returned in the body of the response.

Here is an example of a simple WAR that implements an "echo" REST web service. The WAR is useful for testing and performing a demo using APIman.

Go to

<http://localhost:8080/apiman-echo/sample/path>

The response might be like

{ "method" : "GET", "resource" : "/apiman-echo/sample/path", "length" : -1, "uri" : "/apiman-echo/sample/path", "headers" : { "Accept-Language" : "en-US,en;q=0.5", "Host" : "localhost:8080", "Accept-Encoding" : "gzip, deflate", "User-Agent" : "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:24.0) Gecko/20100101 Firefox/24.0", "Accept" : "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8", "Connection" : "keep-alive" } }

Deploy

To deploy this demo to a running WildFly 10 server, follow the command provided below

**cd apiman-quickstarts/echo-service mvn clean install mvn wildfly:deploy**

Run Locally

You can also run the project locally using the Jetty Web container and access the REST Echo Service at this address

[**http://localhost:9999/apiman-echo/sample/path**](http://localhost:9999/apiman-echo/sample/path)

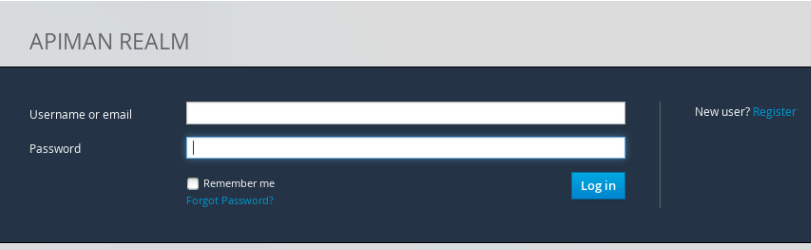
# Configuring an API Provider

Generally, a new API provider user is created for a before created an API provider organization. Follow the procedure provided below to create API provider users

1. Login to the **API REALM** home page

Note: If you have already logged in to the **Admin** page, log out and log in again.

The **Login** dialog box appears



1. Select the **New user? Register** option and fill the required fields.

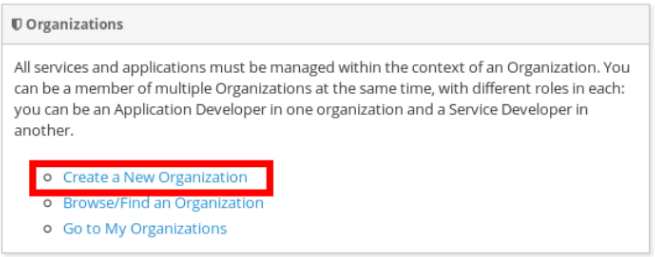


1. Click **Register** and log out.
2. Repeat steps from 1 to 3 to create a register a new application developer user.



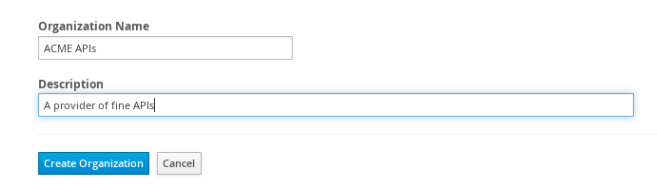
To create a API provider organization, follow the steps provided below

1. Log in to the API as the **apiProvider** user.
2. Choose **Create a new Organization** from the **Organizations** window.



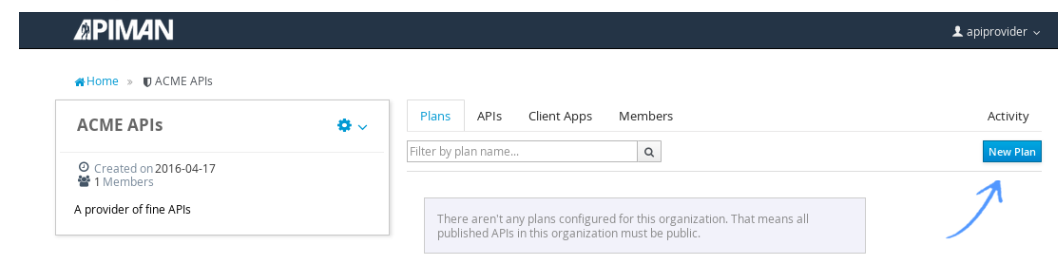
The **New Organization** window appears.

1. Select a name and description for the organization. For example, you can provide the following



1. Click the **Create Organization** button**.**

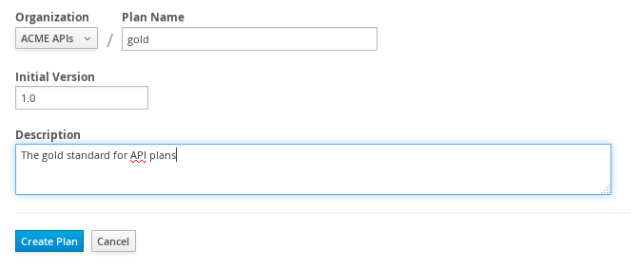
The new organization appears.



## Creating a new plan

Creating a new plan within an organization enables assigning a set of policies to the organization. To create a new plan, follow the steps provided below

1. Enter the **Organization** name, **Plan Name**, **Initial Version** and the **Description** of the plan in the appropriate fields of the **New Plan** window



1. Click **Create.**

## Adding a new policy

APIman supports several policy types related to security, limiting, modification and many others. To add a new policy, follow the procedure provided below

1. Navigate to the **Plan Policies** window.
2. Describe your new policy in the **Policy Description** field. An example policy is provided below

Authorization policy: The policy enables only users with specific user roles (For example: Admin) to use the API resources. The policy may also be used to

1. Click **Add Policy** button to add your new policy

Sample configuration

"rules" : [

{

"pathPattern": "/admin/.\*",

"verb": "\*",

"role": "admin"

},

{

"pathPattern": "/.\*",

"verb": "GET",

"role": "user"

}

],

"multiMatch": true,

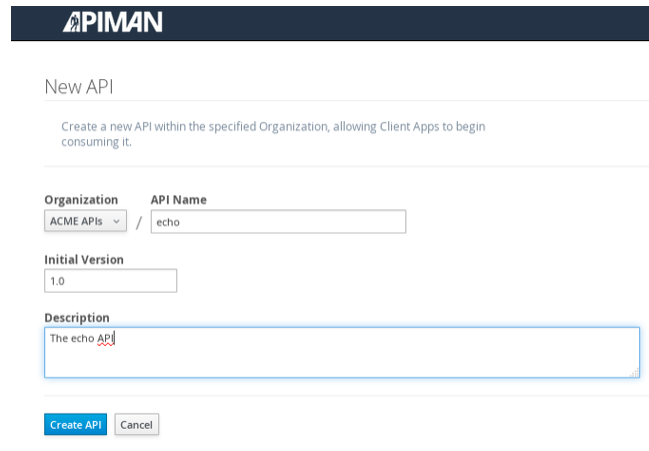
"requestUnmatched": false

}

# Defining an API

Define your API by providing an appropriate name for your API. This helps the providers and consumers to locate your API easily in the API manager.

1. Enter **echo** as the name of the API in the **API Name** field.
2. Enter the initial version of the API in the **Initial Version** field.
3. Add the description of the API in the **Description** field.

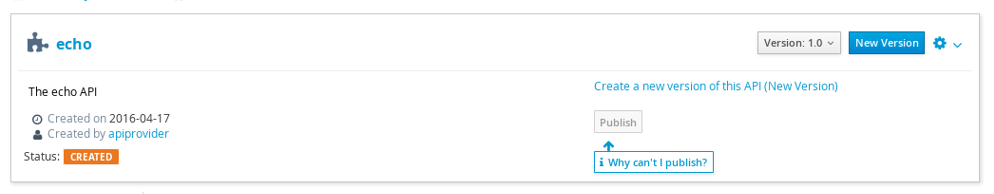


1. Click **Create API**

**Publishing an API**

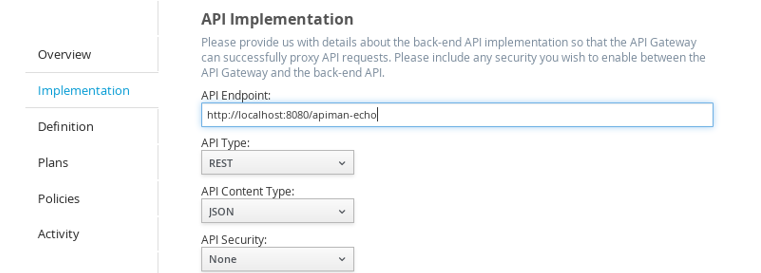
It is mandatory to publish the API created as the API Gateway acts as a proxy for the specific API and it must know the API’s actual URL.

Example URL is: [**http://localhost:8080/apiman-echo**](http://localhost:8080/apiman-echo)

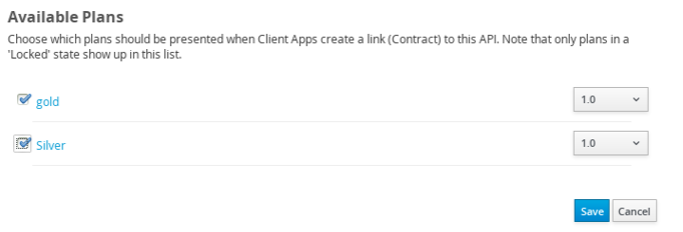


A screen asking for the details for the back-end API implementation appears.

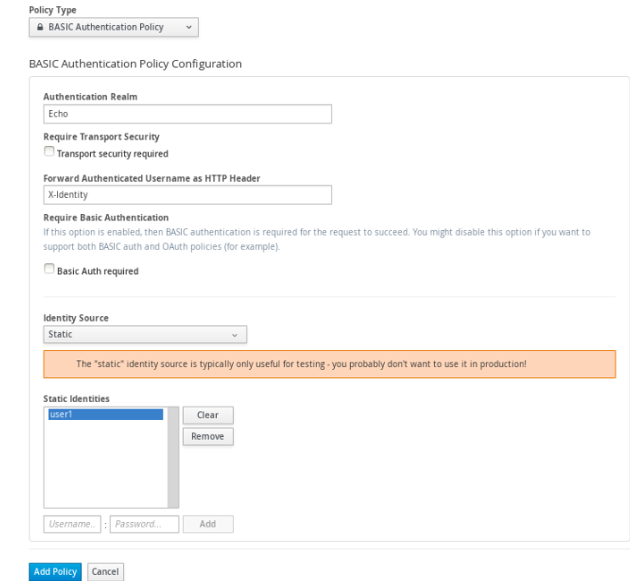
1. Fill the appropriate back-end API implementation details in the **API Implementation** screen.



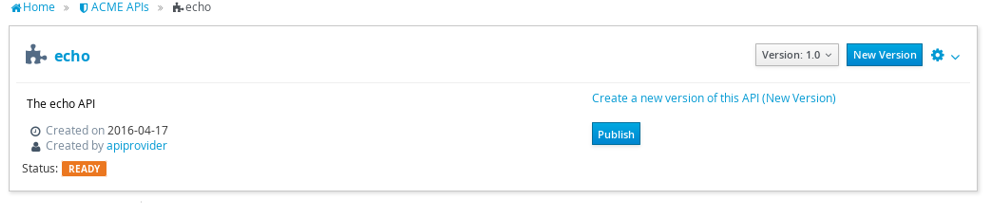
1. Choose the appropriate plan from the **Available Plans**



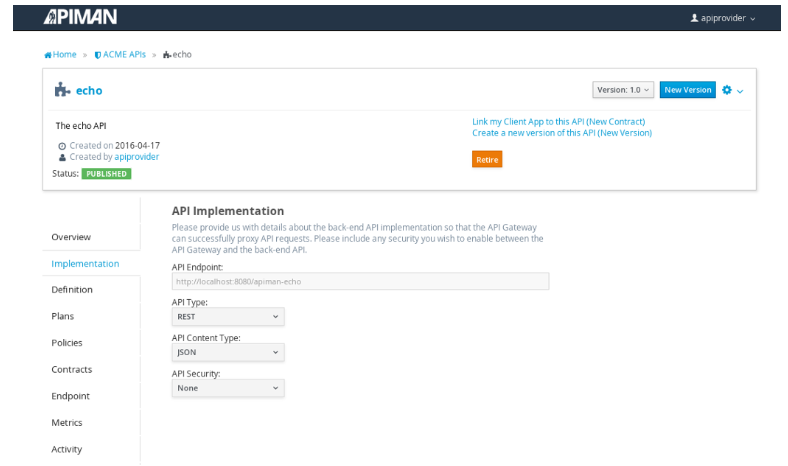
1. Add the policy type as **Basic authentication policy** and click **Add policy.**



1. Click Publish to publish the API to the API Gateway.



1. View the published API

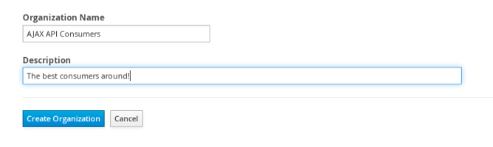


# Configuring an API Consumer Organization

## Creating a new organization

Follow the steps provided below to create a new organization

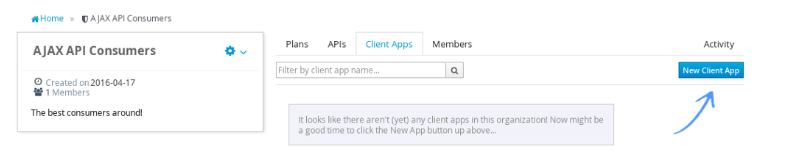
1. Log in to the **APIman** interface as **appdev** user.
2. Add the **Organization Name** and the **Description** of the new organization in the appropriate fields.
3. Click **Create Organization**.



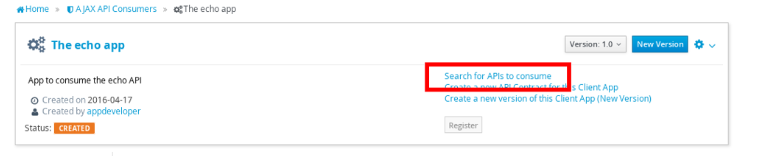
## Creating a new application

To create a new application, follow the steps provided below

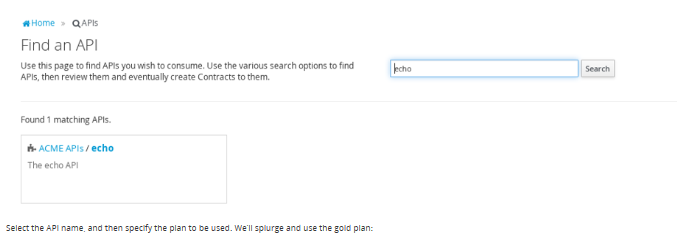
1. Click the **Client Apps** tab
2. Click **New Client App** to create a new client.



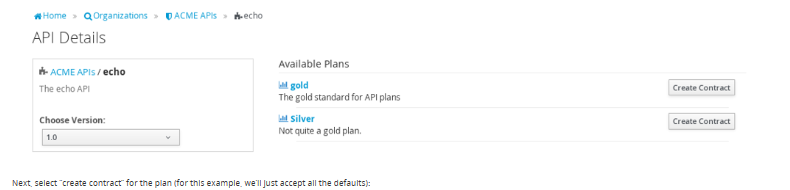
1. Click **Search for the APIs to consume** to search for the API to be used by the application



1. Type the name of the API as **echo** in the Search field of the **Find an API** screen

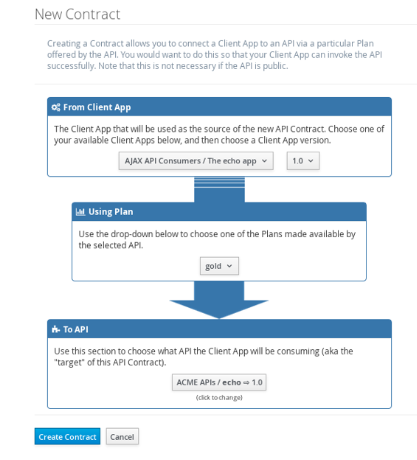


1. Select the name of the API as **echo** and the plan as **Gold** in the **API details** screen



## Creating a contract

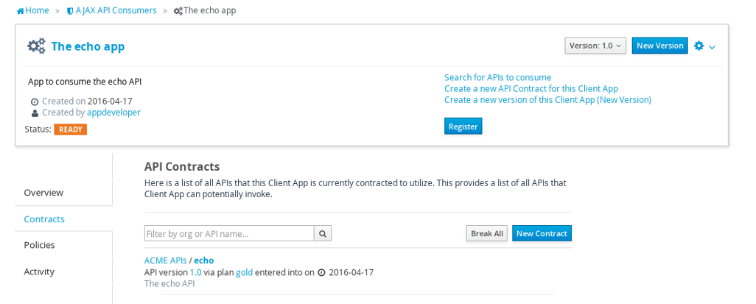
1. Add the contract details of the **Gold** plan in the **New Contract** page

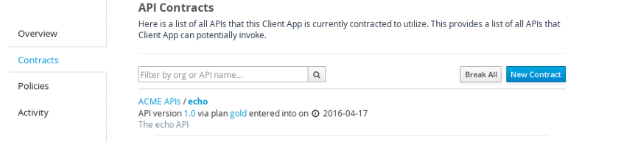


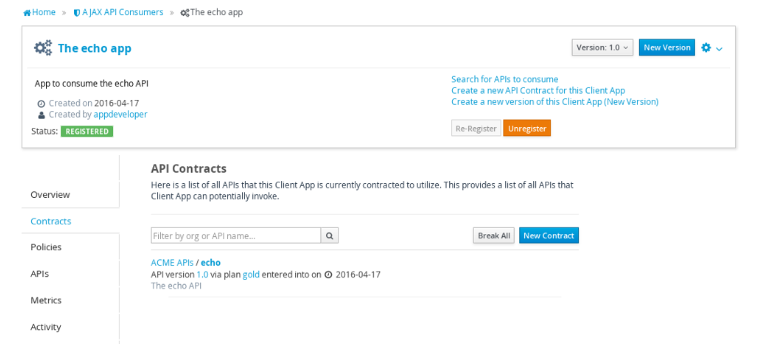
1. Click **Create Contract**

## Registering the application with the API Gateway

1. Click **Register** in **The echo app** screen



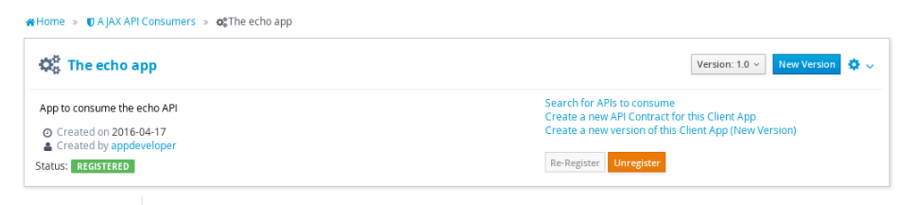




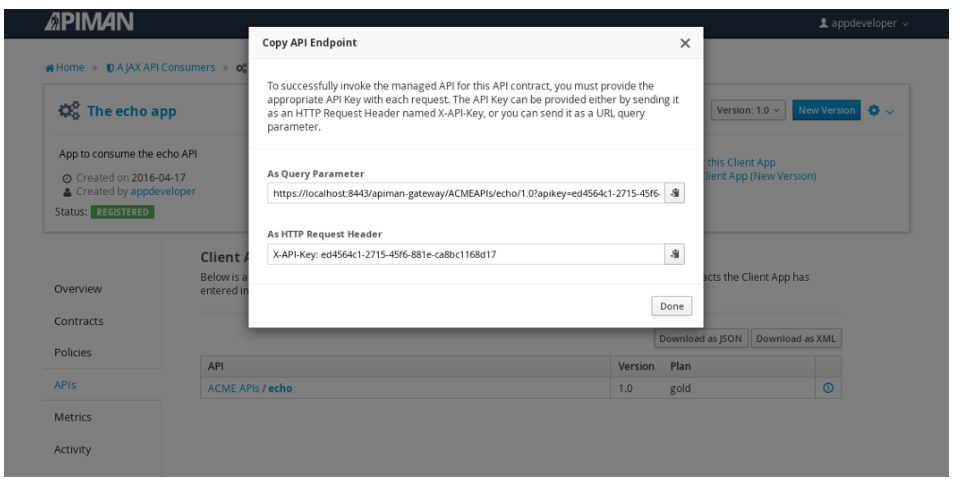
## Accessing the API through the API Gateway

1. Select the API and then click on the **“i”** character on the right side of the API name in the **Apps** tab of the **APIman** user interface

The API Key and the API’s HTTP endpoint in the API Gateway are displayed.





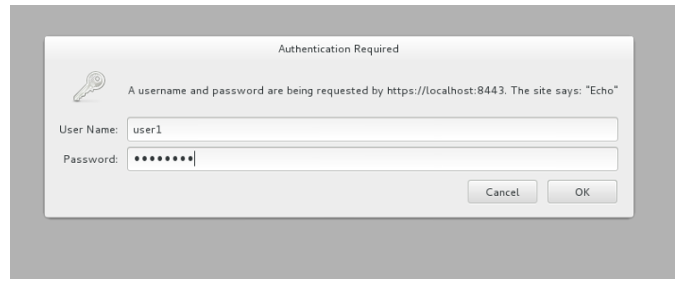


1. Provide the API Key through an HTTP Header (X-API-Key) or a URL query parameter with each request to access the API through the API Gateway



1. Copy the URL into the clipboard.
2. Paste the URL for the managed API in a new browser or a tab.

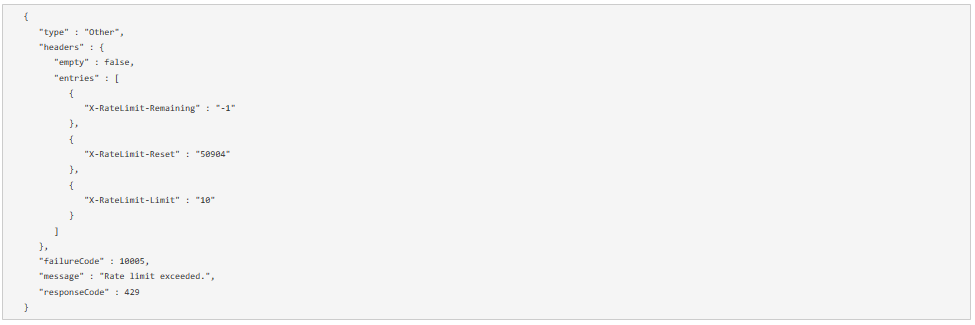
The authentication policies are applied at the runtime and a new dialog box displays



1. Enter the username and password (user1/password) that we defined while we created the authentication policy to access the API.

A Get request appears in the API along with a successful response





# Testing a user defined policy

Follow the steps provided below to test the authorization policy.

1. Log in to the APIman interface.
2. Enter the login credentials.
3. Enter Developer in the User Role field

The interface displays a message You are not an authorized user.

1. Replace the User Role field as Admin.

The interface allows you to access the application.

# Glossary

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |