



Apiman

Table of Contents

Overview	1
What is Apiman?	1
Why JBoss Apiman is needed?	1
Installing the Apiman in Wildfly10	3
Prerequisites.....	3
Installing WildFly 10.....	3
Starting the Server	3
Accessing the Web UI	4
Logging to the Apiman administration	4
Creating a user for an API provider	4
Creating an organization.....	5
Creating a plan for an organization	6
Creating a policy to plan	7
Add an API	9
Configuring user for an API consumer	12
Creating an organization.....	13
Assigning a plan.....	13
Testing the usage policy.....	16
Basic response.....	16
Quota information	17

Overview

What is Apiman?

The current objective in enterprise software development is to create decoupled client applications and use APIs to associate them. This helps to reuse the functionalities across different applications and business units.

The key features of Apiman are as follow:

- Configuring centralized governance policy
- Tracking APIs and related consumers
- Sharing and identifying APIs with ease
- Leveraging common policy configuration across different APIs.

The Apiman is based on three elements:

- An **Organization**, where all the plans and policies of an Apiman are stored. It can match the entire organization or a section of it.
- A **Plan** explains all the services according to the given policies. It helps to define various Apiman services.
- A **Policy** is an action executed by Apiman gateway when any service request is issued. In the data model policies are at the last level and the basis for building higher level elements.

Why JBoss Apiman is needed?

The purpose of JBoss Apiman is to provide an easy to use and powerful API Manager that makes the API Gateway fast, small, and low overhead to execute standard API management functionality.

Apiman fulfills API providers' and consumers' needs by implementing:

- **API Manager** - The API Manager provides an easy way for API providers to use a web UI to define API contracts for their APIs, apply these contracts across multiple APIs, and control role-based user access and API versioning. These contracts can govern access to APIs and limits on the rate at which consumers can access APIs. The same UI enables API consumers to easily locate and access APIs.

- **API Gateway** - The gateway applies the API contract policies of API Management by enforcing at runtime the rules defined in the contracts and tracking the API consumers' use of the APIs for every request made to the APIs. The way that the API Gateway works is that the consumer of the API accesses the API through a URL that designates the API Gateway as a proxy for the API. If the policies defined to govern access to the API, the API Gateway then proxies requests to the API's backend API implementation.

Installing the Apiman in Wildfly10

Prerequisites

To install Apiman, the following software components must be installed on your system:

- Java (version 8)
- Git
- Maven

Installing WildFly 10

Perform the following steps to install WildFly 10:

1. Download the [WildFly 10.1.0.Final](#) and unpack the file into the directory in which you want to run the server.
2. Download the [Apiman 1.5.5.Final](#) overlay for WildFly 10 and unpack the file into the WildFly directory.

Starting the Server

The apiman overlay contains the following:

- apiman binaries (several WAR files)
- apiman-specific WildFly 9 configuration (standalone-apiman.xml)
- apiman RDBMS datasource (h2)
- pre-configured admin user with password admin123!
- pre-configured h2 database for the API Manager (populated with default values)
- embedded Elasticsearch instance for metrics and gateway storage

Use the following command to start the server:

```
cd 10.1.0.Final
./bin/standalone.sh -c standalone-apiman.xml
```

Accessing the Web UI

Logging to the Apiman administration

1. Open the [Apiman UI](#) in your browser.
2. Login with username as **admin** and password as **admin123!**.
3. Click **Log in**.



Change the password after the first login.

API Management page opens.

Creating a user for an API provider

1. To create a user for an API provider, logout from the admin account in the API Manager UI.

The login dialog opens.

A screenshot of the 'LOG IN TO APIMAN' dialog box. The dialog has a light gray header with the title 'LOG IN TO APIMAN'. Below the header, there are two input fields: 'Username or email' and 'Password'. To the right of these fields, there is a link that says 'New user? Register'. At the bottom of the dialog, there are two buttons: 'Cancel' and 'Log in'.

2. Select **New user** to register the API provider user.

REGISTER WITH **APIMAN**

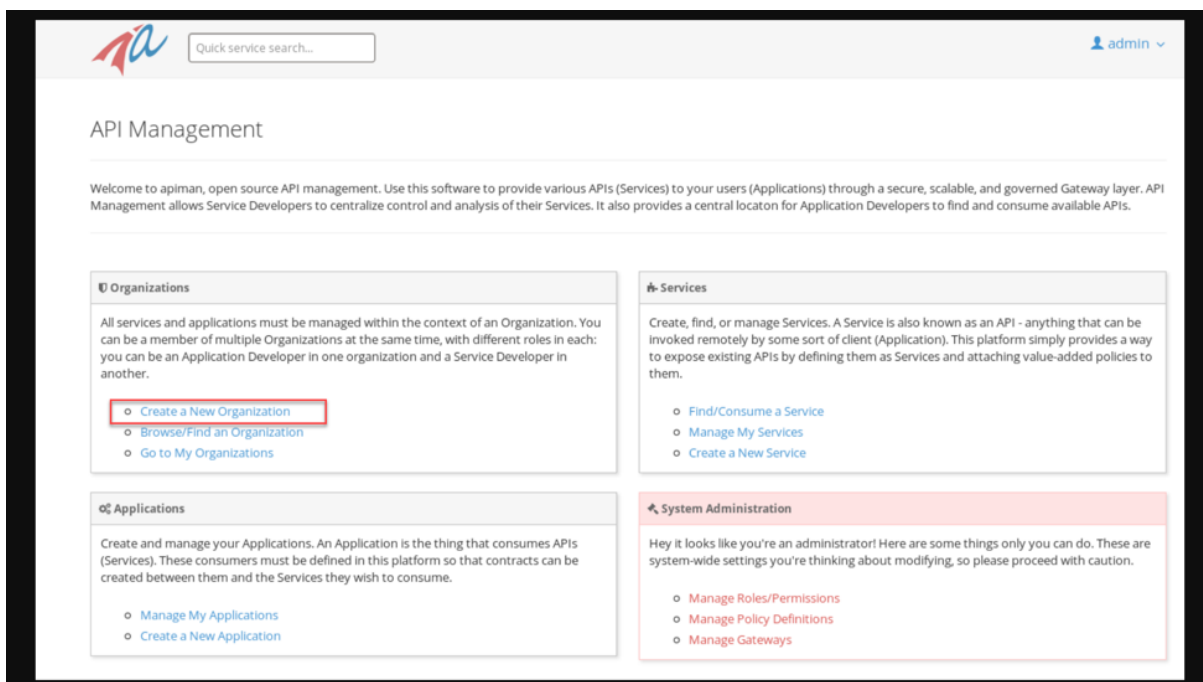
Username	<input type="text" value="servprov"/>
First name	<input type="text" value="Service"/>
Last name	<input type="text" value="Provider"/>
Email	<input type="text" value="servprov@example.com"/>
Password	<input type="password" value="....."/>
Confirm password	<input type="password" value="....."/>

[« Back to Login](#)

3. Fill in the required field and then click **Register**.

Creating an organization

1. On the API Management page, select **Create a New Organization**.

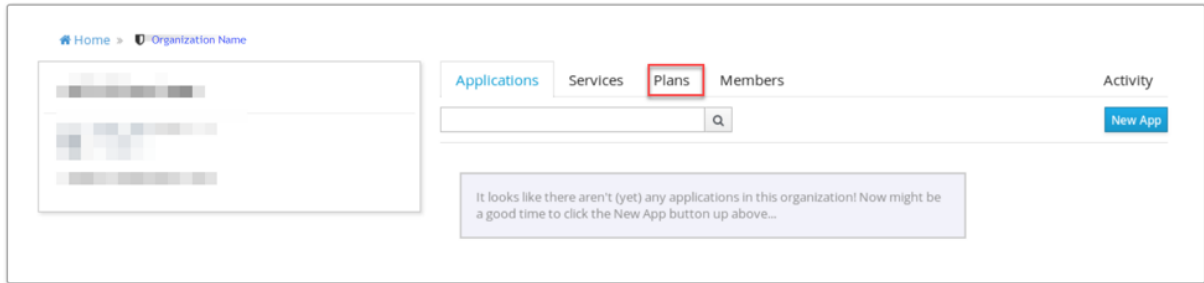


2. Fill in an **Organization Name** and **Description** fields.

3. Click **Create Organization** to open your new organization page.

Creating a plan for an organization

1. Select the **Plans** tab to create a new plan.



2. On the New Plan page, fill out the required details.

A screenshot of the 'New Plan' form in the Apiman web UI. The form has a title 'New Plan' and a subtitle 'Create a new Plan within the specified Organization, allowing you to assign groups of Policies to Services.' The form contains the following fields:

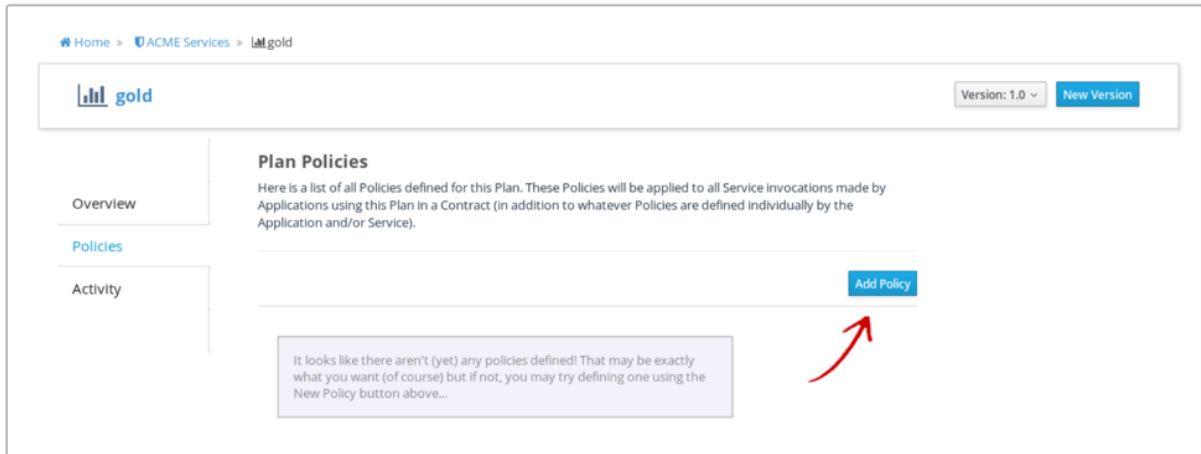
- Organization**: A dropdown menu showing 'ACME Services'.
- Plan Name**: A text input field containing 'gold'.
- Initial Version**: A text input field containing '1.0'.
- Description**: A text input field containing 'The gold standard for service plans'.

At the bottom of the form, there are two buttons: 'Create Plan' (in blue) and 'Cancel' (in grey).

3. Click **Create Plan**.

Creating a policy to plan

1. Click **Add Policy** to add a policy to the plan.



2. On the Add Policy page, fill out the required details.

Add Policy

Adding a policy will allow its specific functionality to be applied to the service invocation as part of the overall Policy Chain.

Policy Type

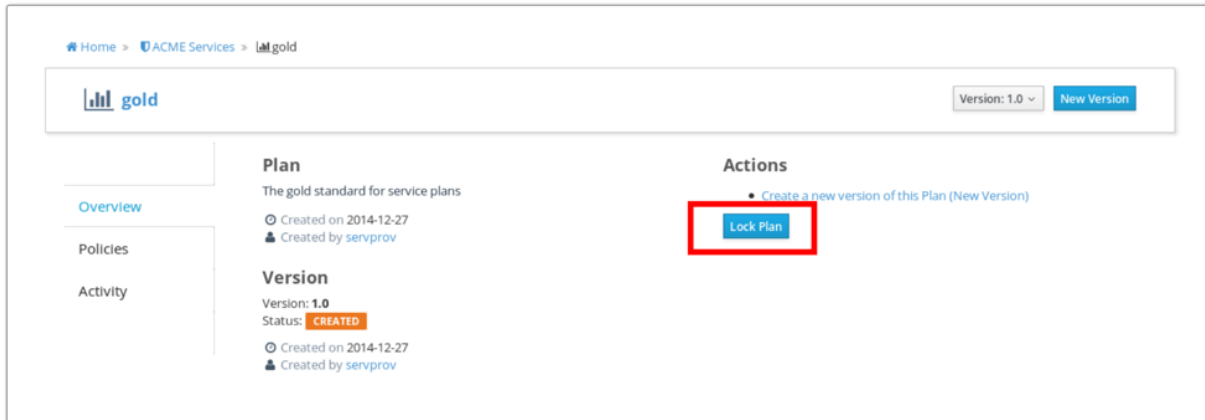
Rate Limiting Policy Configuration

I want to limit request rates to requests per per

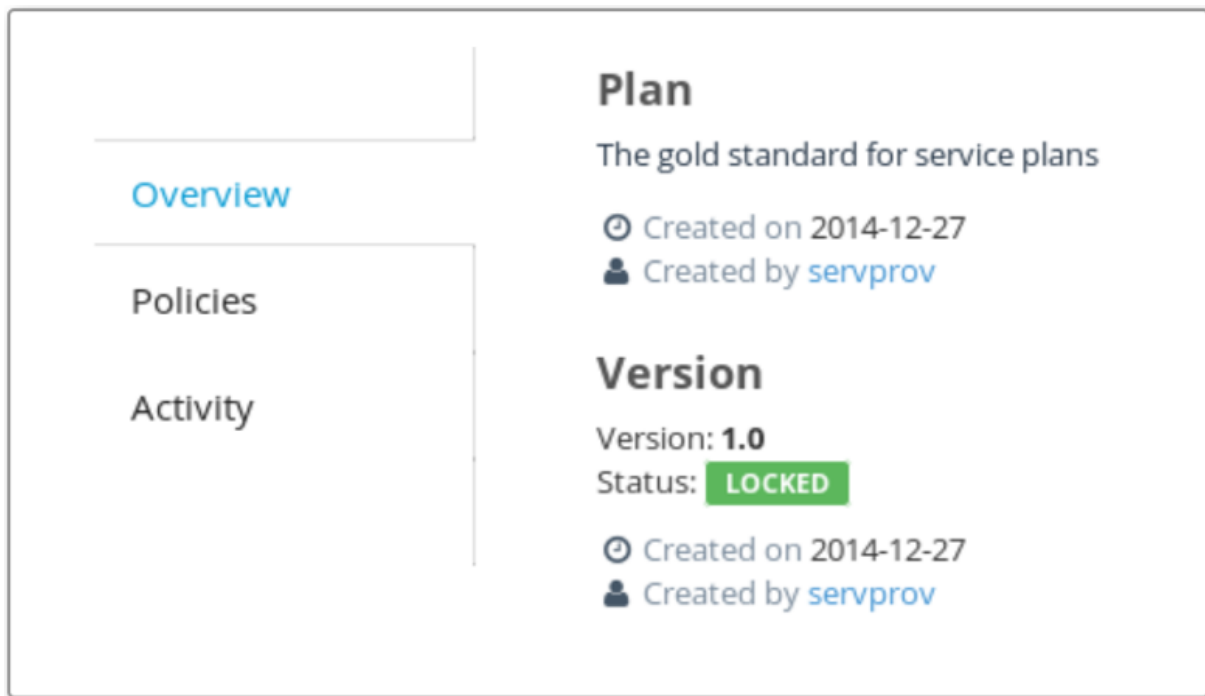
[Add Policy](#) [Cancel](#)

3. Click **Add Policy**.

4. After the Policy is created, click **Lock Plan** to lock the plan.

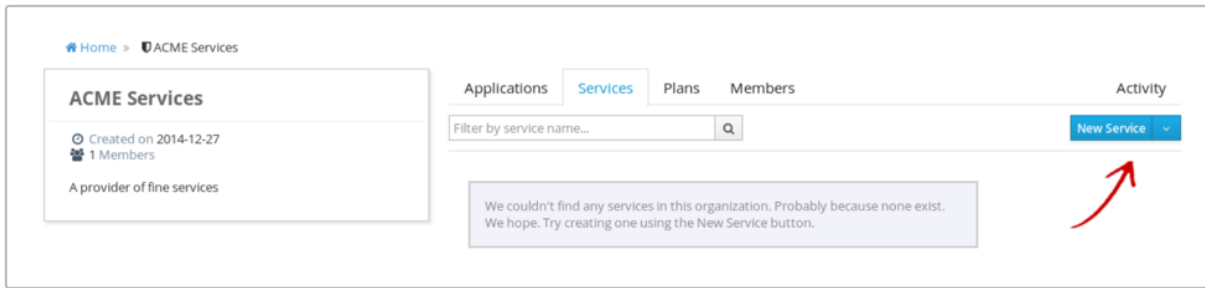


The locked plan opens as displayed:



Add an API

1. On the organization page, select the **Services** tab and click **New Service** to define an API.



2. On the New Service page, fill out the required details.

The screenshot shows the 'New Service' page in the Apiman web UI. The page has a heading 'New Service' and a subheading 'Create a new Service within the specified Organization, allowing Applications to begin consuming it.' Below this is a form with the following fields:

- Organization:** A dropdown menu showing 'ACME Services'.
- Service Name:** A text input field containing 'echo'.
- Initial Version:** A text input field containing '1.0'.
- Description:** A text input field containing 'The echo service'.

At the bottom of the form are two buttons: 'Create Service' (highlighted in blue) and 'Cancel'.

3. Click **Create Service**.
4. Define API implementation:
 - a. On the **Implementation** tab, fill out the API Endpoint field as <http://localhost:9999/apiman-echo>

The screenshot shows the 'Service Implementation' page for the 'echo' service. The breadcrumb navigation is 'Home > ACME Services > echo'. The left sidebar contains links for Overview, Implementation (selected), Plans, Policies, Contracts, and Activity. The main content area has a 'Service Implementation' header. Below it, the 'API Endpoint' is 'http://localhost:8080/apiman-echo'. The 'API Type' is set to 'REST'. There are 'Save' and 'Cancel' buttons at the bottom.

b. Click **Save**.

5. Select the **Plans** tab, and then select a required plan.

The screenshot shows the 'Available Plans' page for the 'echo' service. The breadcrumb navigation is 'Home > ACME Services > echo'. The left sidebar contains links for Overview, Implementation, Plans (selected), Policies, Contracts, and Activity. The main content area has an 'Available Plans' header with the instruction 'Choose which plans should be presented when Applications create a link (Contract) to this Service.' Below this, there are two plans: 'gold' and 'silver', each with a checkbox and a version dropdown set to '1.0'. There are 'Save' and 'Cancel' buttons at the bottom.

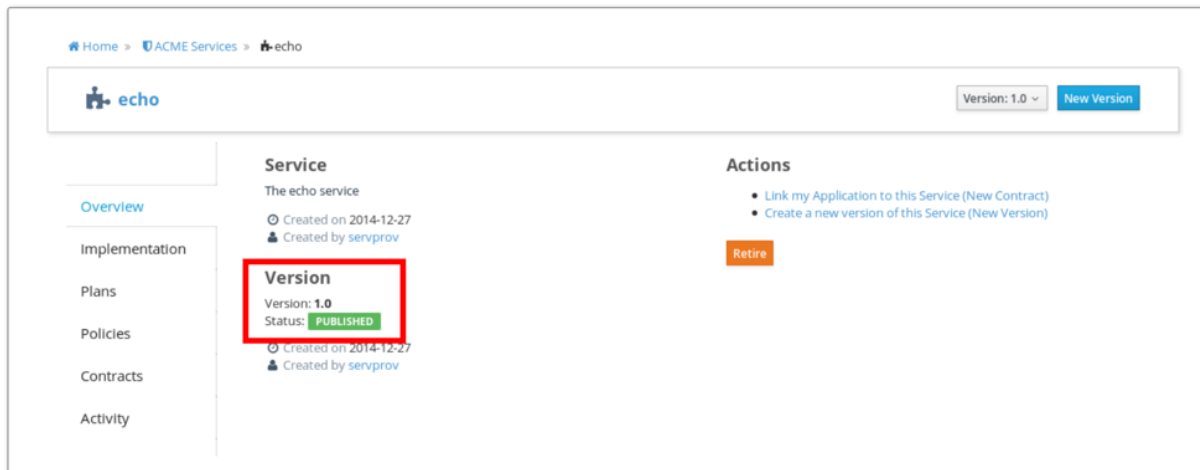
6. Click **Save**.

7. Select the **Policies** tab, and then define an authentication policy.

8. After the authentication policy is added, click **Publish** to publish the API to the API Gateway.

The screenshot shows the 'Overview' page for the 'echo' service. The breadcrumb navigation is 'Home > ACME Services > echo'. The left sidebar contains links for Overview (selected), Implementation, Plans, Policies, Contracts, and Activity. The main content area has a 'Service' section with the title 'The echo service', creation date 'Created on 2014-12-27', and creator 'Created by servprov'. Below this is a 'Version' section with 'Version: 1.0' and 'Status: CREATED' (highlighted with a red box). The 'Actions' section on the right contains two links: 'Link my Application to this Service (New Contract)' and 'Create a new version of this Service (New Version)', and a 'Publish' button (highlighted with a red box).

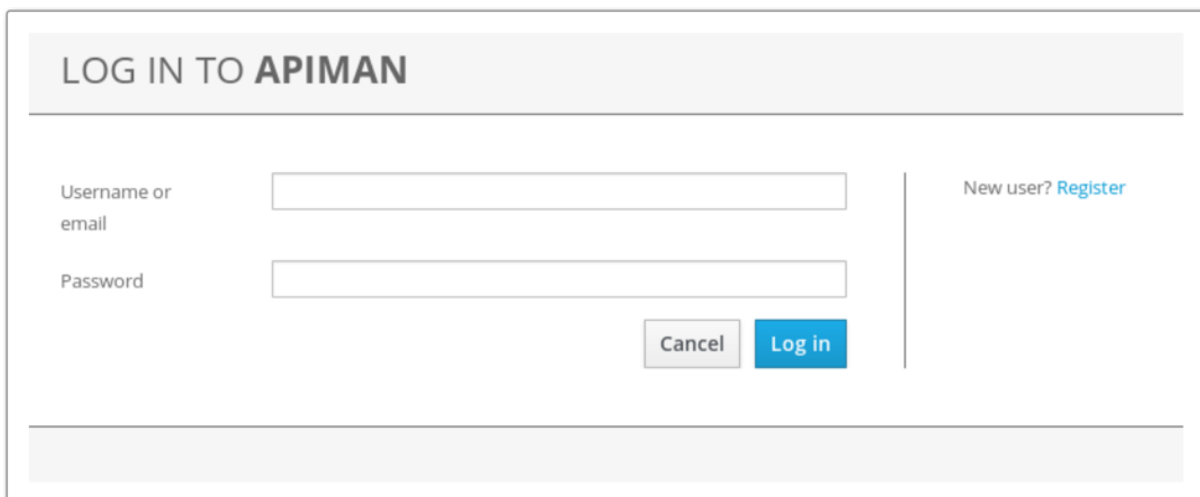
- The status of the API changes to **PUBLISHED**, which means that the created API is now available.



Configuring user for an API consumer

- To create a user for an API consumer, logout from the API Provider account in the API Manager UI.

The login dialog opens.



- Select **New user** to register a new client app developer user.

REGISTER WITH **APIMAN**

Username	<input type="text" value="appdev"/>
First name	<input type="text" value="Application"/>
Last name	<input type="text" value="Developer"/>
Email	<input type="text" value="appdev@example.com"/>
Password	<input type="password" value="....."/>
Confirm password	<input type="password" value="....."/>

[« Back to Login](#)[Register](#)

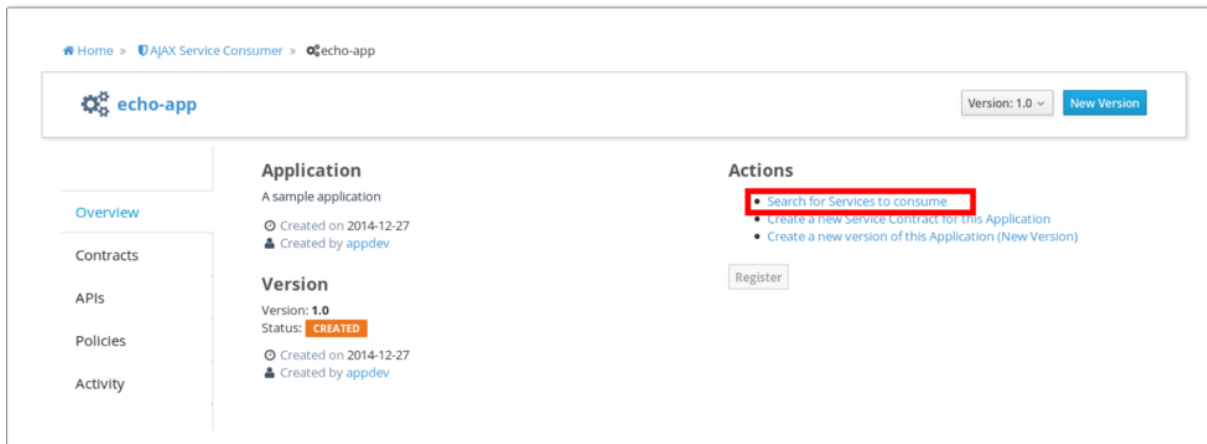
3. Fill in the required field and then click **Register**.

Creating an organization

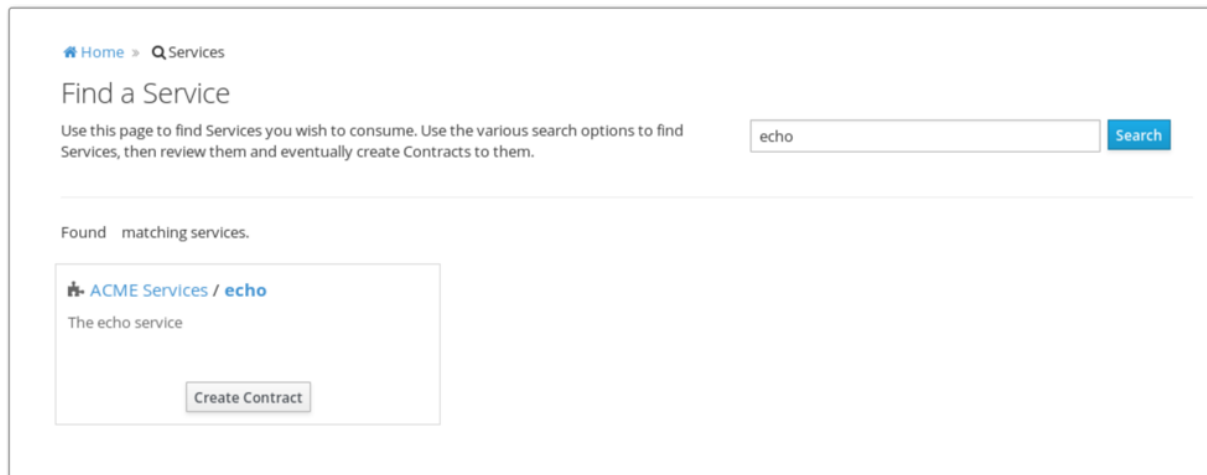
Create an organization for an API consumer. For more information about creating an organization, see [Creating an organization](#).

Assigning a plan

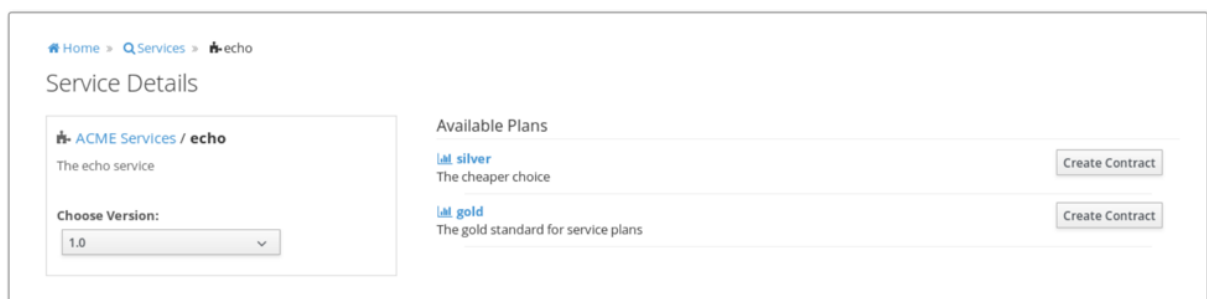
1. Click **Search for Services to consume** to search for the API to be used by the client app.



2. Type the API name on the **Search** box. Matching services appear on the bottom left of the UI.



3. Select the API name, and then specify the plan to use.



4. Click **Create Contract**.
- New Contract page opens.
5. Fill the required details and then click **Create Contract**.
 6. Click **I Agree** to accept the contract terms.

New Contract

Creating a Contract allows you to connect an Application to a Service via a particular Plan offered by the Service. You would want to do this so that your Application can invoke the Service successfully. Note that this is not necessary if the Service is public.

Terms & Conditions

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec lobortis massa vitae pulvinar pretium. Ut suscipit pellentesque auctor. Nullam venenatis pulvinar scelerisque. Sed sit amet leo laoreet, consectetur massa eget, consequat lacus. Maecenas efficitur ornare nibh. Morbi accumsan magna quis felis commodo, quis accumsan turpis molestie. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Nam porta, risus id tempus finibus, justo ipsum facilisis lacus, sit amet dignissim nunc lacus id felis. Cras vestibulum dui vitae magna lacinia, ac scelerisque odio laoreet. Etiam tristique sed quam sit amet finibus. Nulla luctus felis a nunc vestibulum, in dictum sapien condimentum. Vestibulum convallis malesuada lobortis. Curabitur mollis tellus sit amet libero consectetur lobortis. Aliquam erat volutpat. Quisque vel tortor mauris. Ut ac egestas orci. Nunc mollis id nisl fringilla rhoncus. Integer bibendum rhoncus lorem. Nam commodo leo et turpis molestie, vitae tristique risus faucibus. Proin accumsan ex sit amet eros sollicitudin, et condimentum justo suscipit. Morbi ac facilisis urna. Phasellus quis dui et dui luctus fringilla eget vitae erat. Proin malesuada vestibulum convallis.

Sed eleifend non justo ut efficitur. Maecenas iaculis odio ut lacus scelerisque, et maximus nunc cursus. Vestibulum eu lorem quis ante egestas scelerisque. Etiam nisi massa, luctus non mauris vitae, sagittis aliquet turpis. In auctor posuere neque, sit amet aliquam lacus. Donec hendrerit pharetra nisl, nec imperdiet mi iaculis non. Phasellus bibendum vulputate metus, vel varius elit vestibulum eu. Phasellus fermentum dapibus est, sed convallis nulla ultricies ac. Maecenas venenatis, mauris eget lobortis aliquet, odio odio iaculis orci, et tristique nisi mauris quis lacus. Curabitur vehicula, nisl at vulputate efficitur, elit risus sollicitudin eros, nec molestie elit magna lobortis magna. Integer molestie lectus ac imperdiet consequat.

7. Click **Register**.



Apiman will generate a unique key, which a consumer can use to access the private API.

[Home](#) >
[AJAX Service Consumer](#) >
echo-app

Version: 1.0

Overview

Contracts

APIs

Policies

Activity

Application

A sample application

Created on 2014-12-27
Created by appdev

Version

Version: 1.0
Status: REGISTERED

Created on 2014-12-27
Created by appdev

Actions

- Search for Services to consume
- Create a new Service Contract for this Application
- Create a new version of this Application (New Version)

Unregister

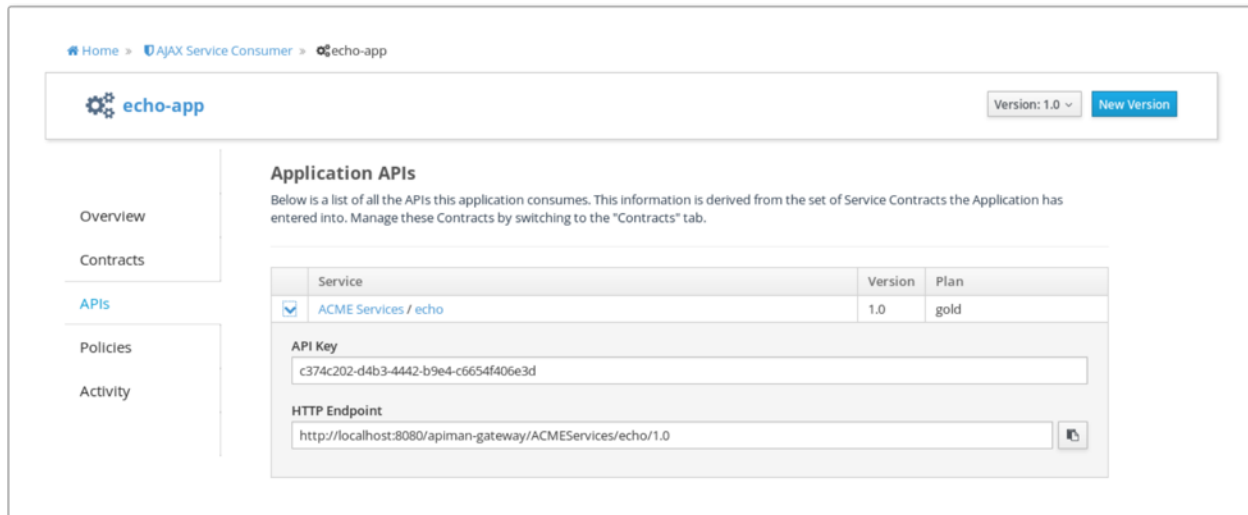
Issue 1.0

©2020

March 2020 | Page 15

Testing the usage policy

On the Consumer page, select the APIs tab, where you can copy the API key and the location of the HTTP endpoint.



Basic response

To check that you can get through to the Echo service, which was made private and is proxied by the Apiman, use the Apiman-provided endpoint and append a regular Echo service path, for example:

```
curl -k -H "X-API-Key: 81430d0e-8f62-46d4-822b-167eddaled7f" -X GET
https://localhost:8443/apiman-gateway/EchoOrganization/EchoAPI/1.0/hello
{
  "method" : "GET",
  "resource" : "/apiman-echo/hello",
  "uri" : "/apiman-echo/hello",
  "headers" : {
    "Accept" : "/*/*",
    "User-Agent" : "curl/7.55.1",
    "Connection" : "keep-alive",
    "Host" : "localhost:9999",
    "Accept-Encoding" : "gzip"
  },
  "bodyLength" : null,
  "bodySha1" : null
}
```

Quota information

To check the quota consumption, view the response headers, for example:

```
curl -k -H "X-API-Key: 81430d0e-8f62-46d4-822b-167eddaled7f" --head  
https://localhost:8443/apiman-gateway/EchoOrganization/EchoAPI/1.0/hello
```

```
HTTP/1.1 200 OK  
X-Powered-By: Undertow/1  
X-Quota-Remaining: 6  
Server: WildFly/10  
Server: Jetty(9.3.0.v20150612)  
X-Quota-Limit: 10  
Date: Thu, 26 Jul 2018 22:37:57 GMT  
X-Quota-Reset: 84122  
Connection: keep-alive  
Content-Type: application/json  
Content-Length: 320
```