

# Managing a Public API Configuring the API

Next, you are going to define your API as to be public and you will add restrictions, the so-called policies to your API.

## Follow Our Example User Story

David Stringer wants the API to be available to everyone. Therefore, he needs to make his API public and set a matching visibility.



Step 4: Testing and Consuming the API

## Choosing the API Type

### Good to Know

**Scheer PAS API Management** supports the creation and management of two different types of APIs: **public** APIs and **private** APIs.

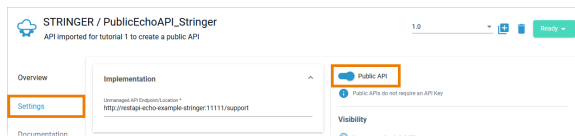
A public API can be consumed by everyone (assuming no additional security policy has been set). It is also very easy to consume a public API: You just need to know its public endpoint. Clients do not need to register for a public API: Neither a client nor a contract are necessary. Compared to a private API, a public API requires less configuration.

Refer to [API Management Guide > API Types](#) for more detailed information.

### On this Page:

- [Choosing the API Type](#)
- [Setting the Visibility](#)
- [Adding Policies to the API](#)
  - [1. Ignoring Certain Service Resources](#)
  - [2. Blocking Certain IP Addresses](#)
- [Publishing the API](#)

A newly imported API is automatically created as private API. To change the API type, enable the toggle button **Public API** in tab **Settings** and confirm:



### Related Documentation:

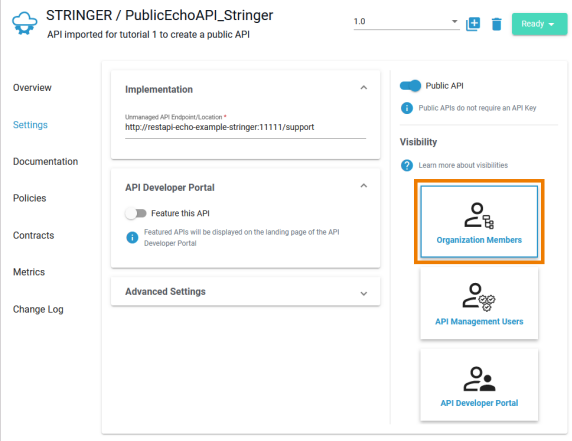
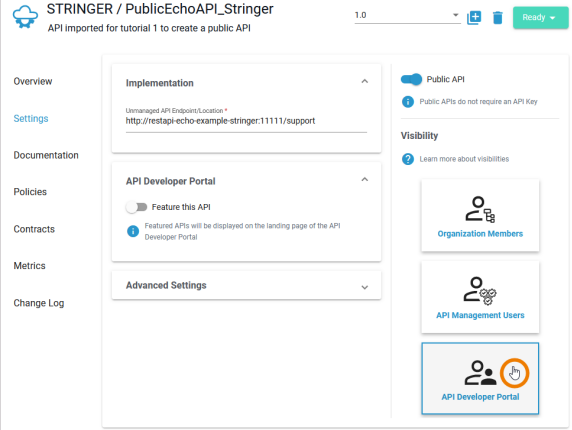
- [API Types: Public vs. Private](#)
- [The Concepts of API Management](#)
- [APIs](#)
  - [API Settings](#)
  - [Publishing an API](#)
- [Policies](#)
  - [Ignored Resources](#)
  - [IP Blocklist](#)

## Setting the Visibility

### Good to Know

The visibility concept of API Management defines which user groups can find the APIs in the Developer Portal. The visibility resides on top of the permission system as another security layer. Visibilities are applicable to public APIs and plans for private APIs. Three different visibilities are available: **Organization members** (default), **API Management users**, and **API Developer Portal users**.

Refer to [API Management Guide > The Concepts of API Management](#) for more details.

	<p>For a newly imported API, the most strict visibility is set by default: Only members of the same organization are allowed to see and use the API.</p> <p>But you want to enable everyone to find the API.</p>
	<p>Click <b>API Developer Portal</b> to make the API visible to all visitors of the API Developer Portal.</p>

## Adding Policies to the API

### Follow Our Example User Story

The **RESTAPI\_Echo\_Example** implements a basic REST service that returns a simple string and a timestamp on a GET request and returns the sent string on a POST request.

David Stringer wants you to implement the following restrictions to the public API:

1. Consumers of the API should only be able to perform the GET request.
2. David has observed requests from obscure IP addresses lately. He wants you to block API access for those addresses.

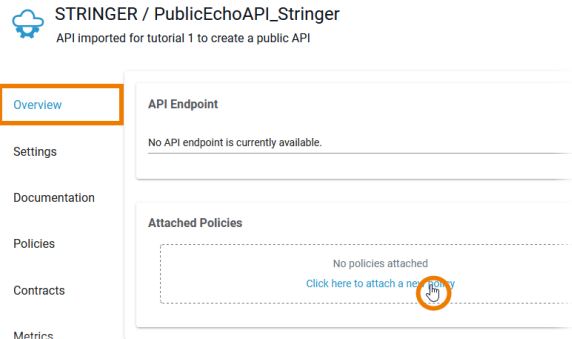
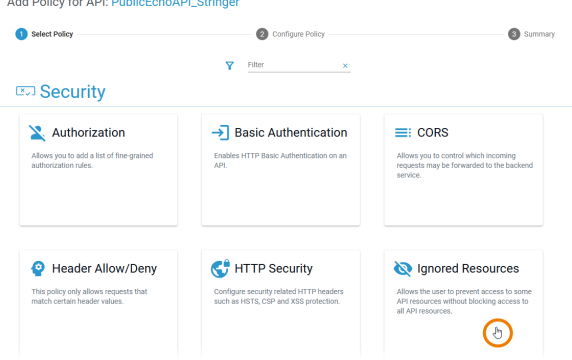
### Good to Know

A policy is a rule or a set of rules API Management uses to manage access to your APIs. Policies are applied to all API requests and represent a unit of work applied at runtime to the request by API Management.

Policies are applied through a policy chain: when a request to an API is made, API Management creates a chain of policies to be applied to that request. The policy chain is applied to the request in a fixed order: Client policies are applied first, then policies added to plans, and finally policies added to the API itself.

Refer to [API Management Guide > Policies](#) for more details.

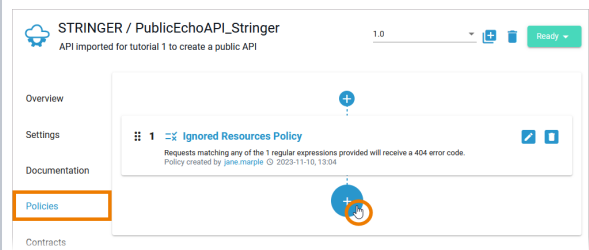

## 1. Ignoring Certain Service Resources

	<p>API restrictions are implemented via policies.</p> <p>In tab <b>Overview &gt; Attached Policies</b> use the link <b>Click here to attach a new policy</b>. This will open the policy wizard.</p> <p>For an overview on all policies provided in API Management refer to <a href="#">API Management Guide &gt; Policies</a>. Each policy and its configuration options are explained on dedicated pages.</p>
	<p>To restrict access to certain service resources you can use the <b>Ignored Resources Policy</b>.</p> <p>Select the <b>Ignored Resources Policy</b> from the list of policies.</p>

	<p>As soon as one of the policies has been selected, the configuration of this policy is displayed.</p> <p>Using the <b>Ignored Resources Policy</b>, you can specify defined resources to be ignored by API Management. Use the link <b>Click here to add a new entry</b> and insert the following:</p> <ul style="list-style-type: none"> <li>• <b>Path:</b> /HelloWorld</li> <li>• <b>Method:</b> POST</li> </ul> <p>This configuration will prohibit POST requests to the HelloWorld resource.</p> <p>Click <b>Next</b> to display the summary and <b>Save</b> to finally attach the policy.</p>
	<p>The <b>Ignored Resources Policy</b> now is displayed on the API's detail page &gt; tab <b>Policies</b>.</p>

## 2. Blocking Certain IP Addresses

To block certain IP addresses, you will now add the **IP Blocklist Policy**.

	<p>Click <b>Plus</b> to open the policy wizard again.</p>
	<p>Select the <b>IP Blocklist Policy</b>.</p> <p>You can use the filter to make it easier to find the policy in the overview.</p>

Add Policy for API: PublicEchoAPI\_Stringer

1 Select Policy 2 Configure Policy 3 Summary

**IP Blocklist**

**Basic Configuration**

Failure Response \*  
Authentication Failure (403)

Specify how the gateway should respond to a client if the request fails due to a violation of this policy.

☐ IP Address Rule

☐ 12.66.66.66

☐ 14.66.66.66

1 - 2 of 2 |< < > >|

Items per page: 5

**Valid IP Formats**

- Literal Address: 192.0.2.0 or 2001:08A:1
- CIDR Address Ranges: 192.0.2.0/24 or 2001:08A:1212::48
- Disjoint Address Ranges: 192.0.2.0-192.0.2.10
- Wildcards: 192.0.2.\*

**Advanced Configuration**

Cancel Search Next

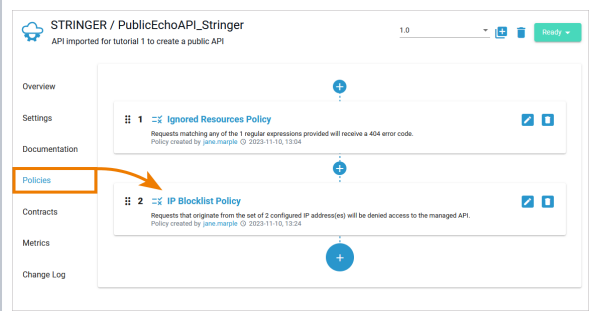
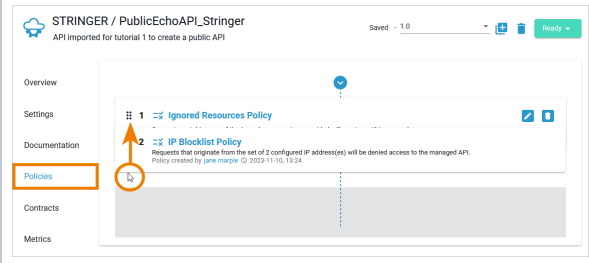
The **IP Blocklist Policy** allows to specify a list of IP addresses to be blocked and to define an error response. Configure the policy as follows:

- **Failure Response:** Authentication Failure (403)

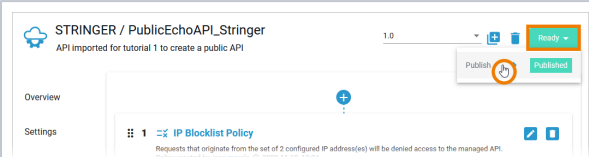
Use the link **Click here to add a new entry** and insert the following:








- **IP Address Rule:** 12.66.66.66
- **Add a new line**
- **IP Address Rule:** 14.66.66.66

Click **Next** to display the summary and **Save** to finally attach the policy.

	<p>Both policies are now displayed on the details page of the <a href="#">Public EchoAPI</a>.</p> <p>The order in which the policies appear in the user interface determines the order in which they will be applied at runtime.</p>
	<p>Drag the <b>IP Blocklist Policy</b> to the top of the list to ensure that this policy is applied first.</p>

## Publishing the API

	<p>Once you have completed the configuration of your API, you must publish it. Only published APIs can be consumed by customers.</p> <p>Since all mandatory configuration of your API has been finished, your API is <b>Ready</b> to be published.</p> <p>Click the status label and publish your API.</p>
---	--

<div><div><div><div><div></div><div>STRINGER / PublicEchoAPI_Stringer</div></div><div>API imported for tutorial 1 to create a public API</div></div><div><div>Save</div><div>1.0</div><div></div><div><div>Published</div><div></div></div></div><div><div>Overview</div><div>Settings</div><div>Documentation</div></div><div><div>API Endpoint</div><div>https://acme- gateway/STRINGER/PublicEchoAPI_Stringer/1.0</div><div></div></div><div><div>API Type</div><div> Public API</div><div> You do not need an API key to perform a request with this API</div></div></div></div>	<p>The new status of the API is displayed.</p> <p>Your API is ready for testing.</p>
---	--