Getting Started with API Management in Azure [APIM]

Darshana Mihiran Edirisinghe · Follow

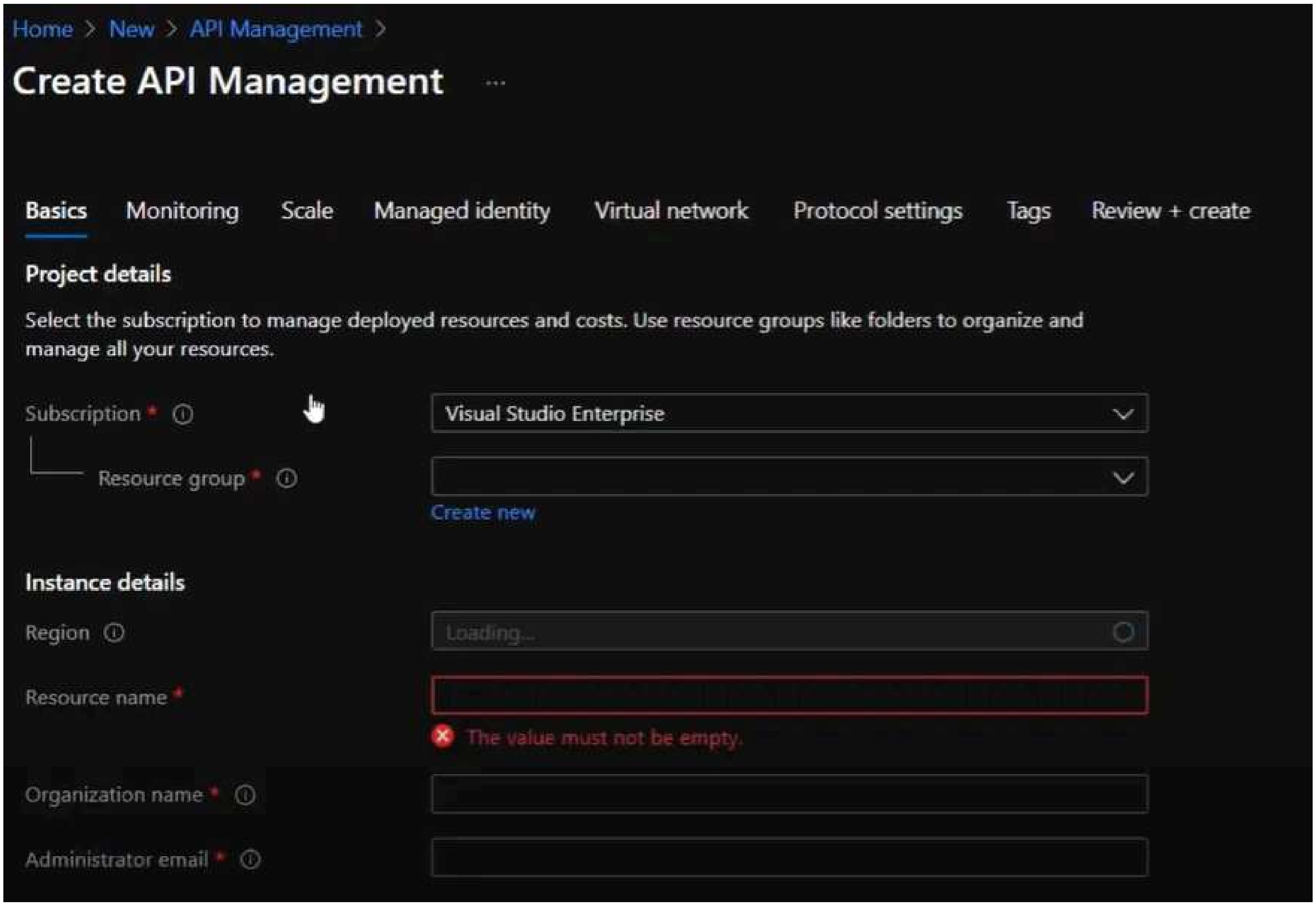
5 min read · Jun 19, 2024

17

This article will cover how to set up API Management (APIM) in Azure involves creating the APIM service, configuring network settings, and handling authentication. Start by creating an APIM service in the Azure Portal, integrating Application Insights for monitoring. Manage authentication and authorization with Azure AD by registering the application, creating app roles, and assigning users. Finally add inbound policies for authorization.

1. Create APIM Service

- Go the Azure Portal and click on create resource.
- Search for API Management and click on create.



Create API Management

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * : Visual Studio Enterprise

Resource group * :

Instance details

Region : Loading...

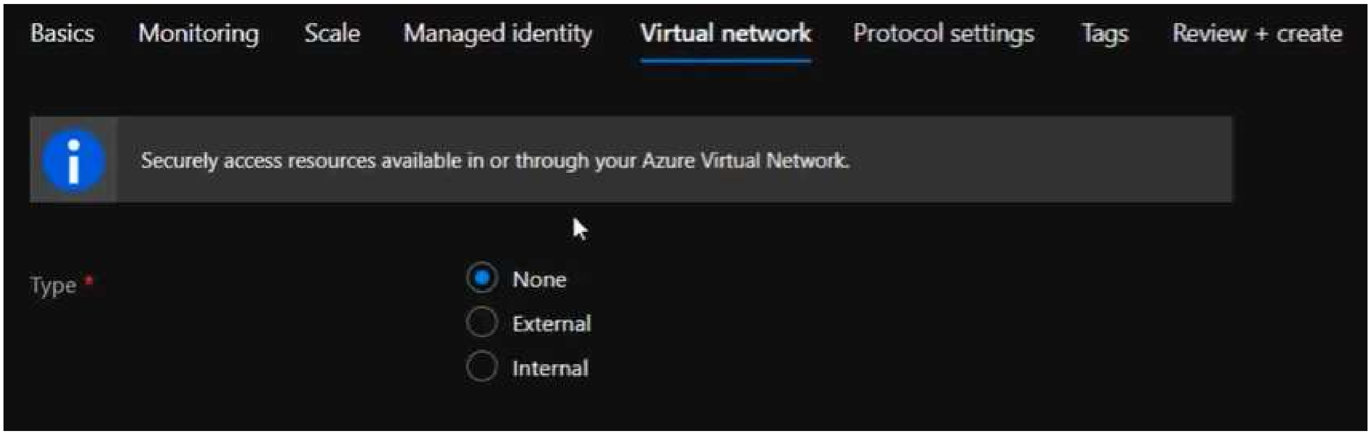
Resource name * :

The value must not be empty.

Organization name * :

Administrator email * :

- Configure the Application Insight for monitoring and logging.



Securely access resources available in or through your Azure Virtual Network.

Type *

● None
○ External
○ Internal

- **None:**
  The API Management service is not connected to any virtual network.
  It is publicly accessible over the internet.

- **External:**
  The API Management service is connected to a virtual network and is accessible from the internet through an external IP address.
  This setting allows you to expose your APIs to external clients while still leveraging the security and control provided by a virtual network.

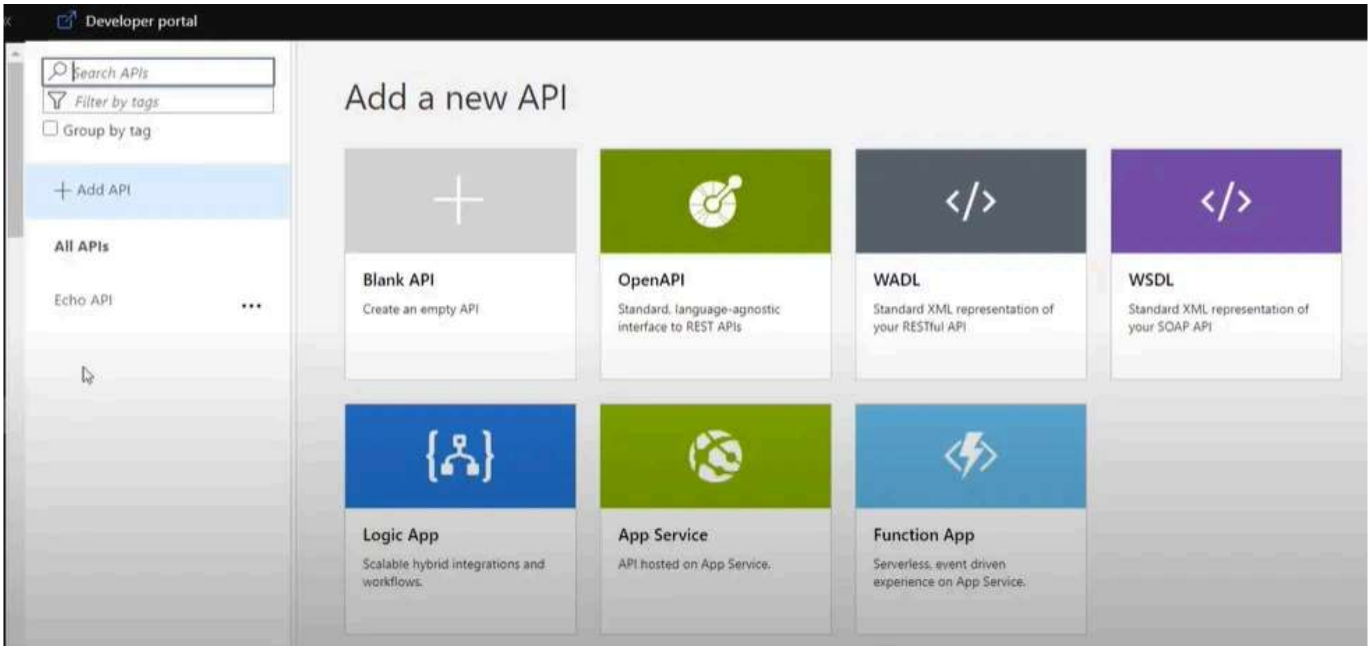
- **Internal:**
  The API Management service is connected to a virtual network but is only accessible from within that virtual network.
  This setting is used to restrict access to the API Management instance to internal clients only, providing enhanced security by preventing internet exposure.

To ensure that your APIs **can only be accessed through the API Gateway and are not directly accessible over the internet**, you should select the **"Internal"** option for your API Management instance.

> Internal: This setting ensures that your API Management instance is only accessible within the Azure Virtual Network. By doing so, you effectively prevent direct internet access to your APIs. This means that external clients must go through the API Gateway to access your APIs, ensuring that the APIs are only exposed through the controlled and secured gateway.

2. Import and Publish API

- Navigate to the API section from the left side menu.



Add a new API

Blank API
Create an empty API

OpenAPI
Standard, language-agnostic interface to REST APIs

WADL
Standard XML representation of your RESTful API

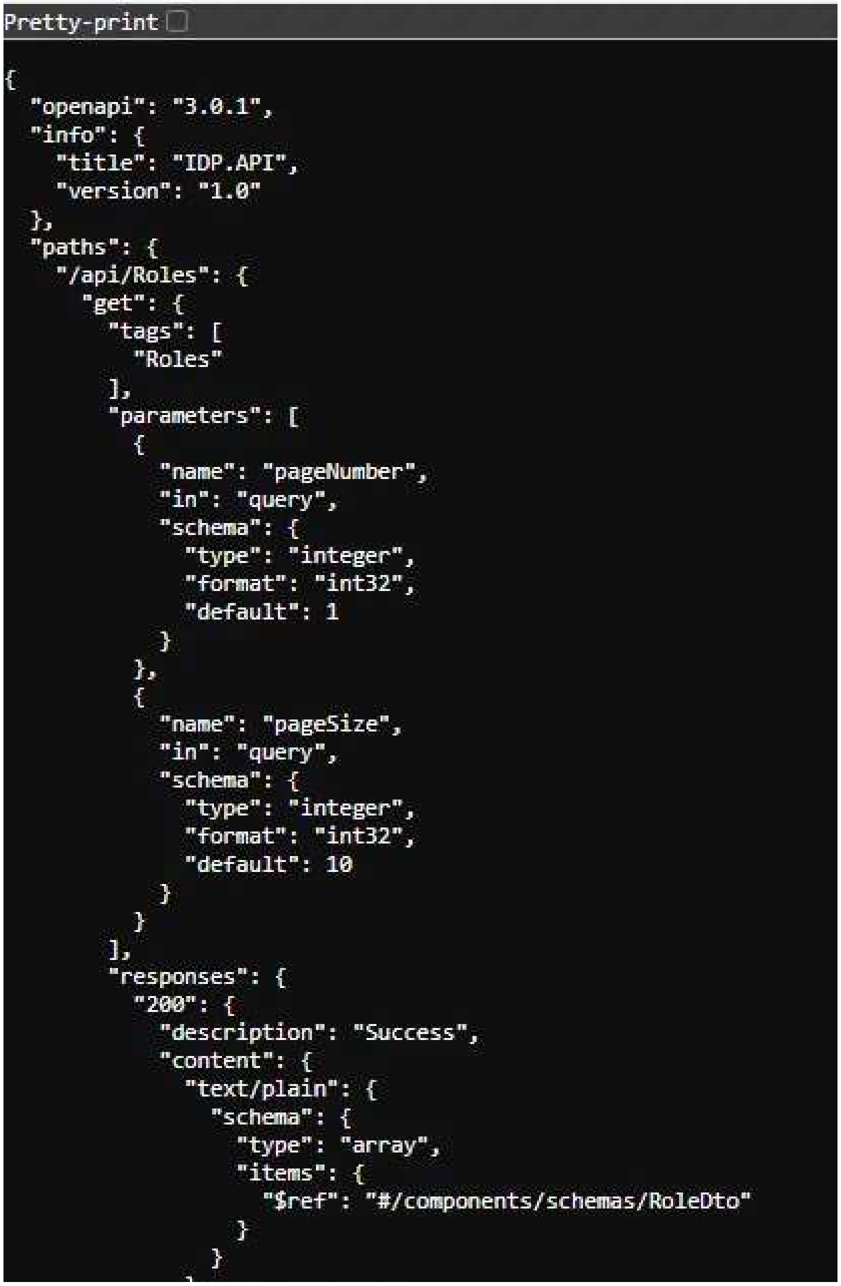
WSDL
Standard XML representation of your SOAP API

Logic App
Scalable hybrid integrations and workflows.

App Service
API hosted on App Service.

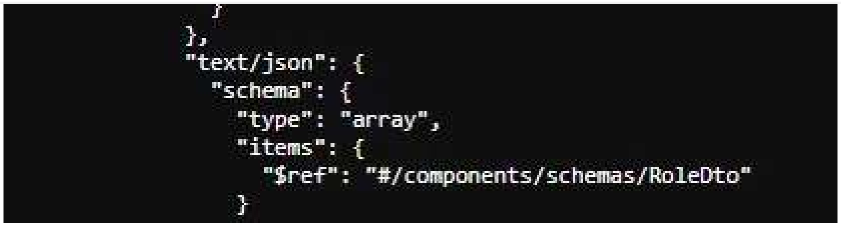
Function App
Serverless, event driven experience on App Service.

- Click on the OpenAPI.
- Copy the of the OpenAPI sepecification. If you have swagger, you can get it by appending `v1/swagger.json` to the api url.



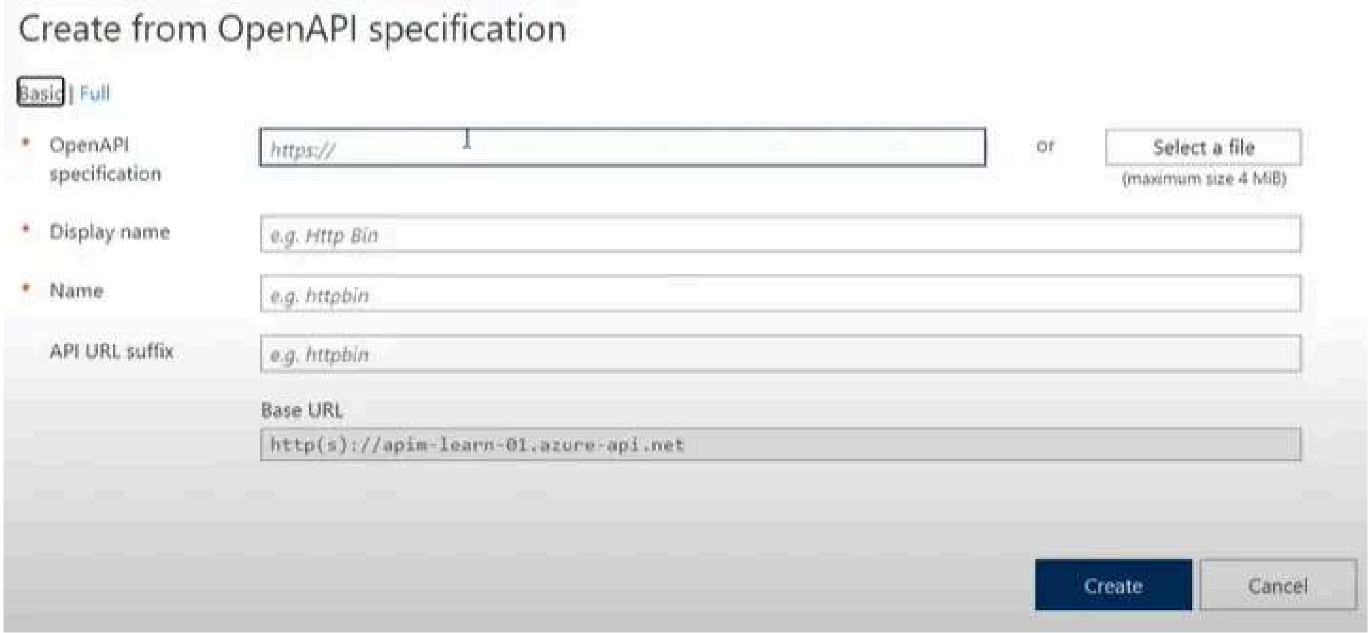
```
Pretty-print □

{
  "openapi": "3.0.1",
  "info": {
    "title": "IDP.API",
    "version": "1.0"
  },
  "paths": {
    "/api/Roles": {
      "get": {
        "tags": [
          "Roles"
        ],
        "parameters": [
          {
            "name": "pageNumber",
            "in": "query",
            "schema": {
              "type": "integer",
              "format": "int32",
              "default": 1
            }
          },
          {
            "name": "pageSize",
            "in": "query",
            "schema": {
              "type": "integer",
              "format": "int32",
              "default": 10
            }
          }
        ],
        "responses": {
          "200": {
            "description": "Success",
            "content": {
              "text/plain": {
                "schema": {
                  "type": "array",
                  "items": {
                    "$ref": "#/components/schemas/RoleDto"
                  }
                }
```



```
              },
              "text/json": {
                "schema": {
                  "type": "array",
                  "items": {
                    "$ref": "#/components/schemas/RoleDto"
                  }
```

- Fill the details.

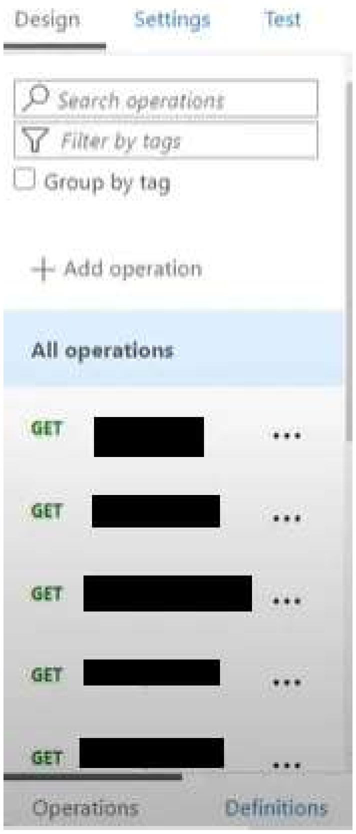


Create from OpenAPI specification

Basic | Full

- * OpenAPI specification: https:// or Select a file (maximum size 4 MiB)
- * Display name: e.g. Http Bin
- * Name: e.g. httpbin
- API URL suffix: e.g. httpbin
- Base URL: http(s)://apim-learn-01.azure-api.net

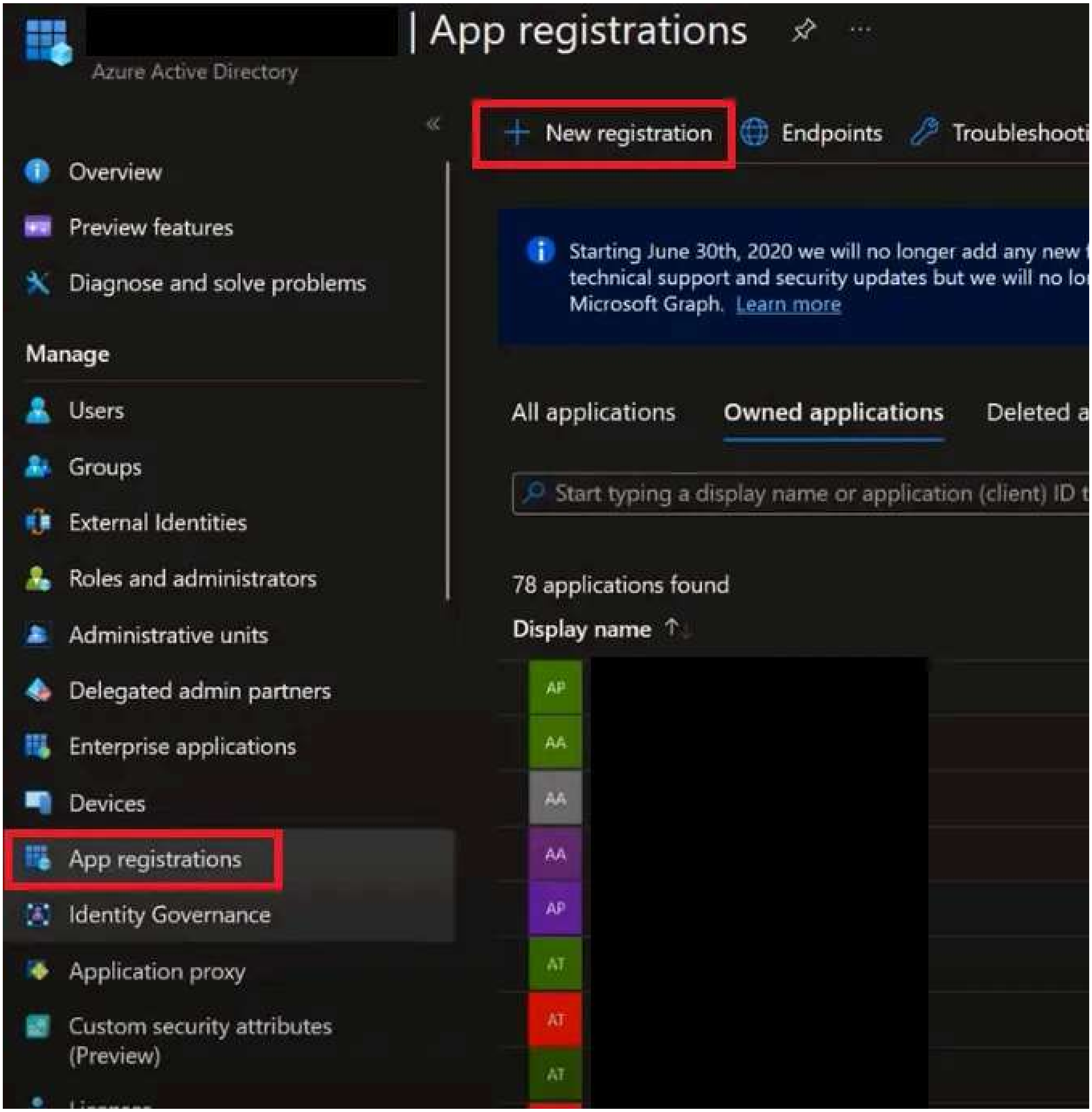
- Once you import the OpenAPI Specification, you will see the APIs listed as shown below.



- You can also test the endpoints using the Test menu.

Handle Authentication and Authorization through Azure AD / Entra ID

- Go to the Azure AD.
- Go to the App registration and select New registration.

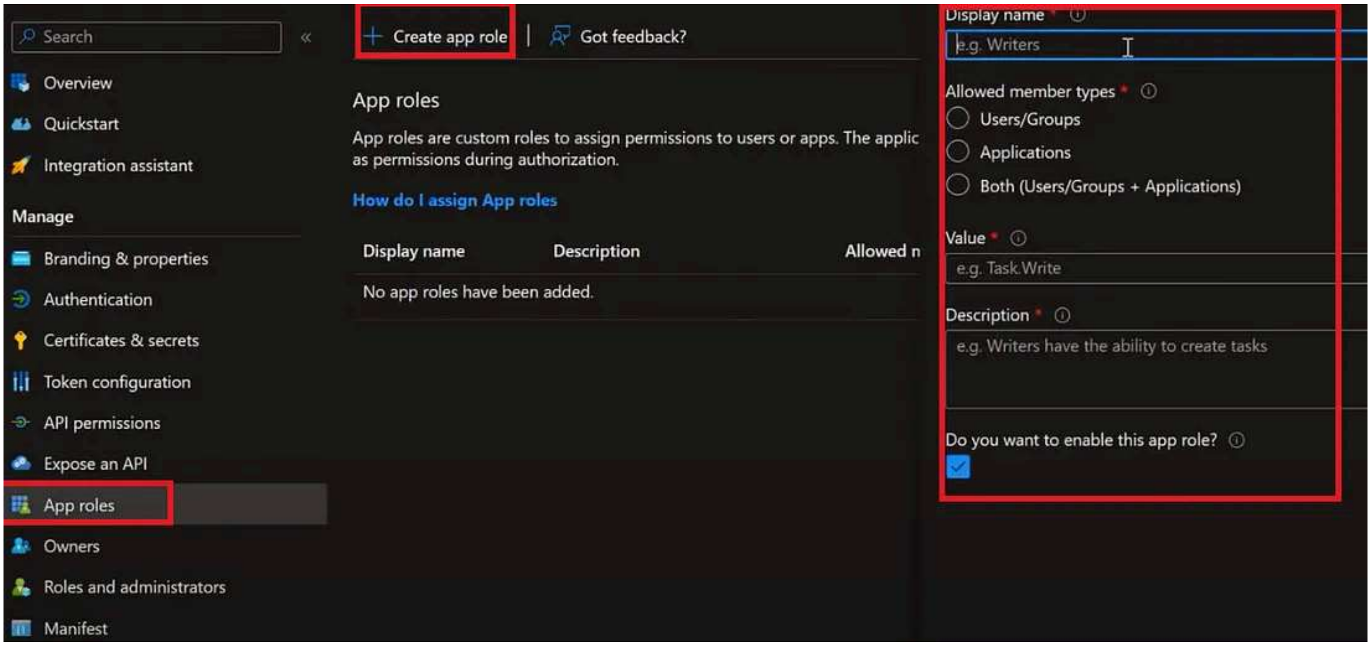


Starting June 30th, 2020 we will no longer add any new technical support and security updates but we will no lo Microsoft Graph. Learn more

78 applications found

[redacted]

- Create App roles for App registration. Select allowed member types as Both(Users/Groups and Applications)



App roles

App roles are custom roles to assign permissions to users or apps. The applic as permissions during authorization.

How do I assign App roles

Display name	Description	Allowed n
No app roles have been added.		

Display name * ⓘ
e.g. Writers

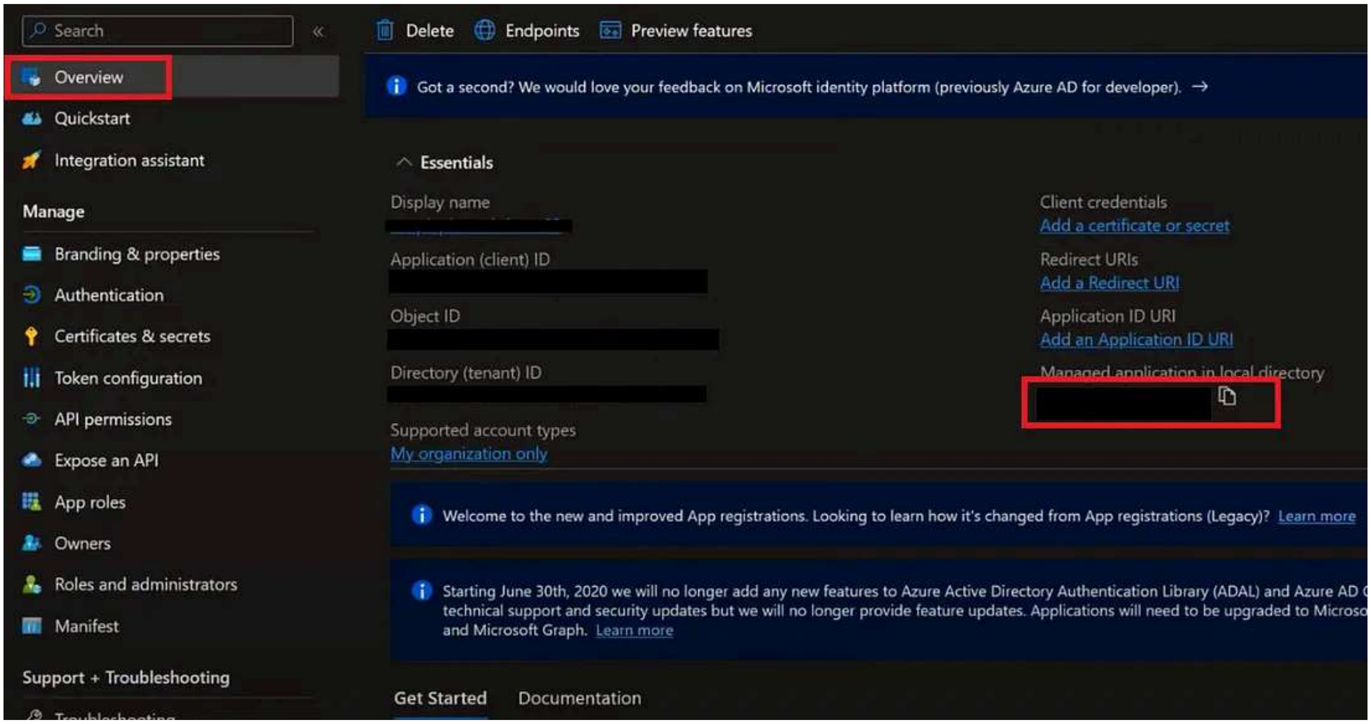
Allowed member types * ⓘ
○ Users/Groups
○ Applications
○ Both (Users/Groups + Applications)

Value * ⓘ
e.g. Task.Write

Description * ⓘ
e.g. Writers have the ability to create tasks

Do you want to enable this app role? ⓘ
☑

- Lets add users and groups to the App Roles.
- Navigate to the overview and select the service principal.



Got a second? We would love your feedback on Microsoft identity platform (previously Azure AD for developer). →

Essentials

Display name
[redacted]

Application (client) ID
[redacted]

Object ID
[redacted]

Directory (tenant) ID
[redacted]

Supported account types
My organization only

Client credentials
Add a certificate or secret

Redirect URIs
Add a Redirect URI

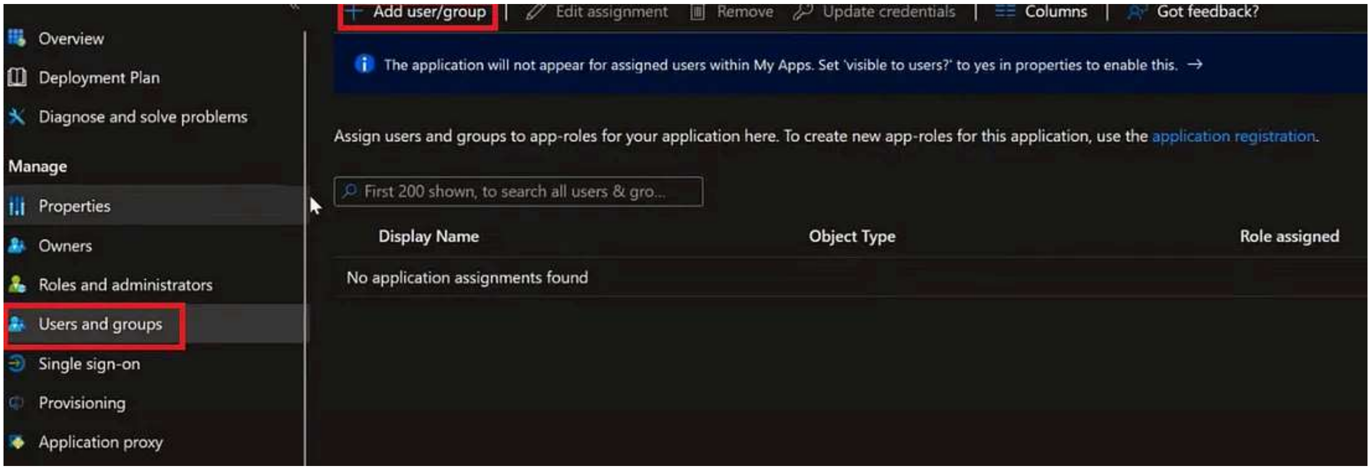
Application ID URI
Add an Application ID URI

Managed application in local directory
[redacted]

Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? Learn more

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microso and Microsoft Graph. Learn more

- Go the Users and Groups and select Add user/group.

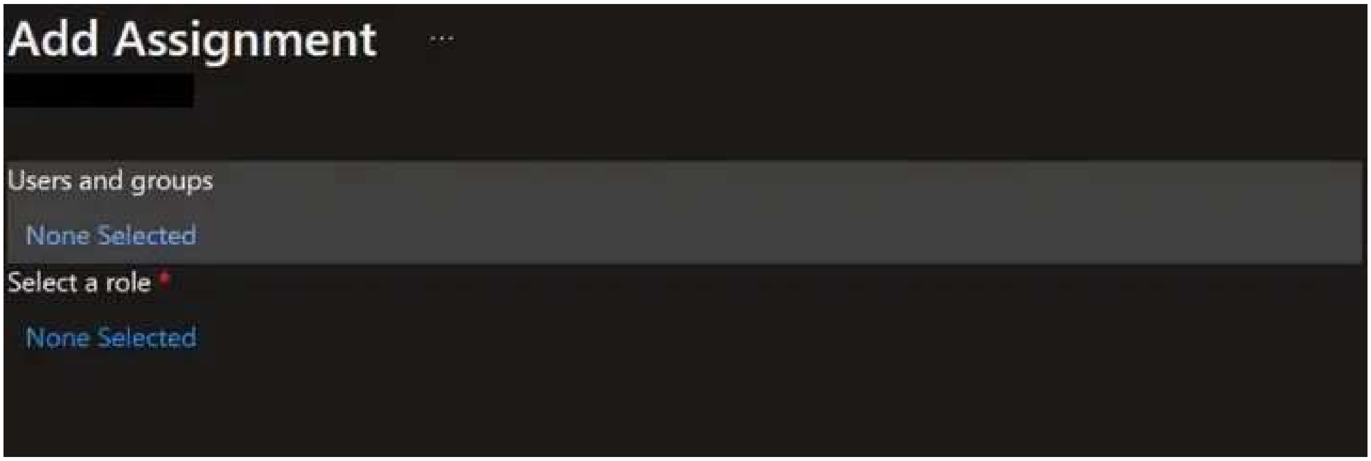


The application will not appear for assigned users within My Apps. Set 'visible to users?' to yes in properties to enable this. →

Assign users and groups to app-roles for your application here. To create new app-roles for this application, use the application registration.

Display Name	Object Type	Role assigned
No application assignments found		

- Select the user and the created role.



Add Assignment ...

[redacted]

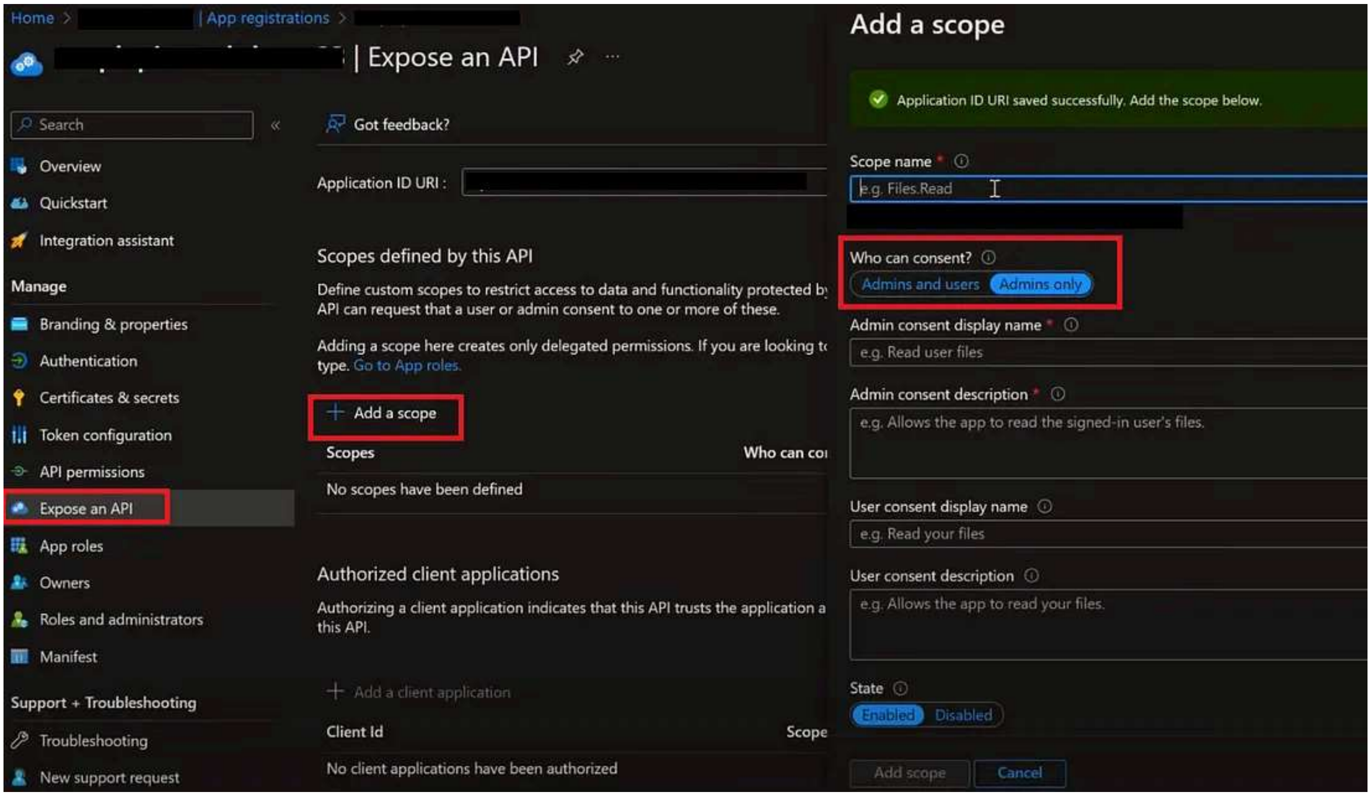
Users and groups

None Selected

Select a role *

None Selected

- Go back to the app registartion and select Expose and API. Select Add a scope.
- Add the application ID URI.



Application ID URI : [redacted]

Scopes defined by this API

Define custom scopes to restrict access to data and functionality protected by API can request that a user or admin consent to one or more of these.

Adding a scope here creates only delegated permissions. If you are looking to type. Go to App roles.

+ Add a scope

Scopes	Who can co
No scopes have been defined	

Authorized client applications

Authorizing a client application indicates that this API trusts the application a this API.

+ Add a client application

Client Id	Scope
No client applications have been authorized	

Add a scope

Application ID URI saved successfully. Add the scope below.

Scope name * : e.g. Files.Read

[redacted]

Who can consent? : Admins and users | Admins only

Admin consent display name * : e.g. Read user files

Admin consent description * : e.g. Allows the app to read the signed-in user's files.

User consent display name : e.g. Read your files

User consent description : e.g. Allows the app to read your files.

State : Enabled | Disabled

Add scope | Cancel

- Select Admin and Users for consent option.

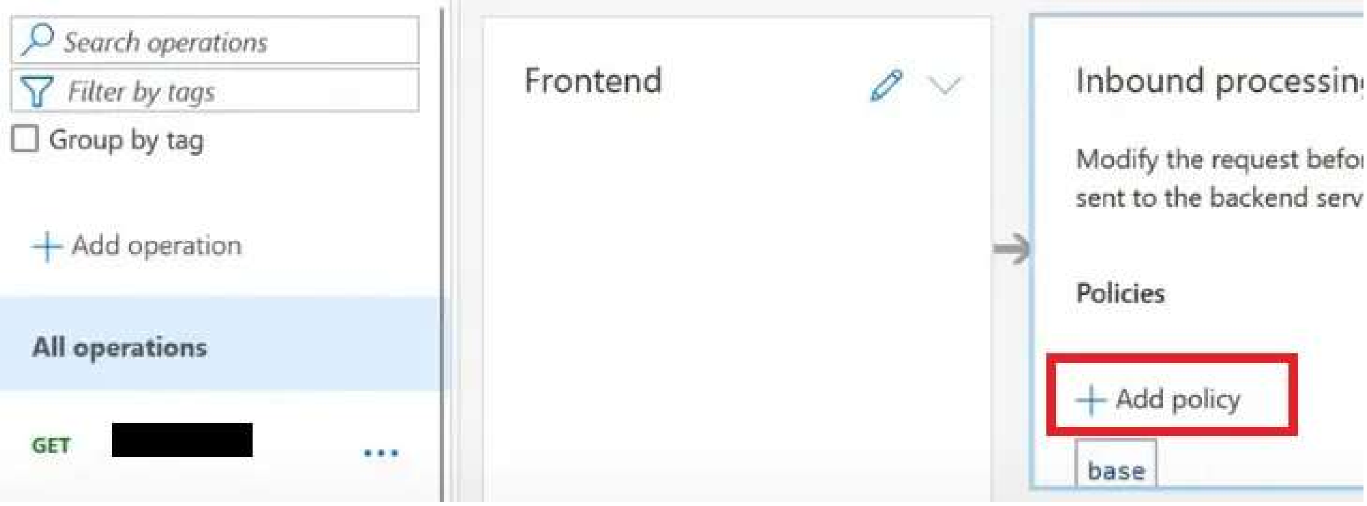
> If you add two app roles to a user, you will find both roles in the claims when you generate the access token.

- Let’s use Azure CLI to generate the token.

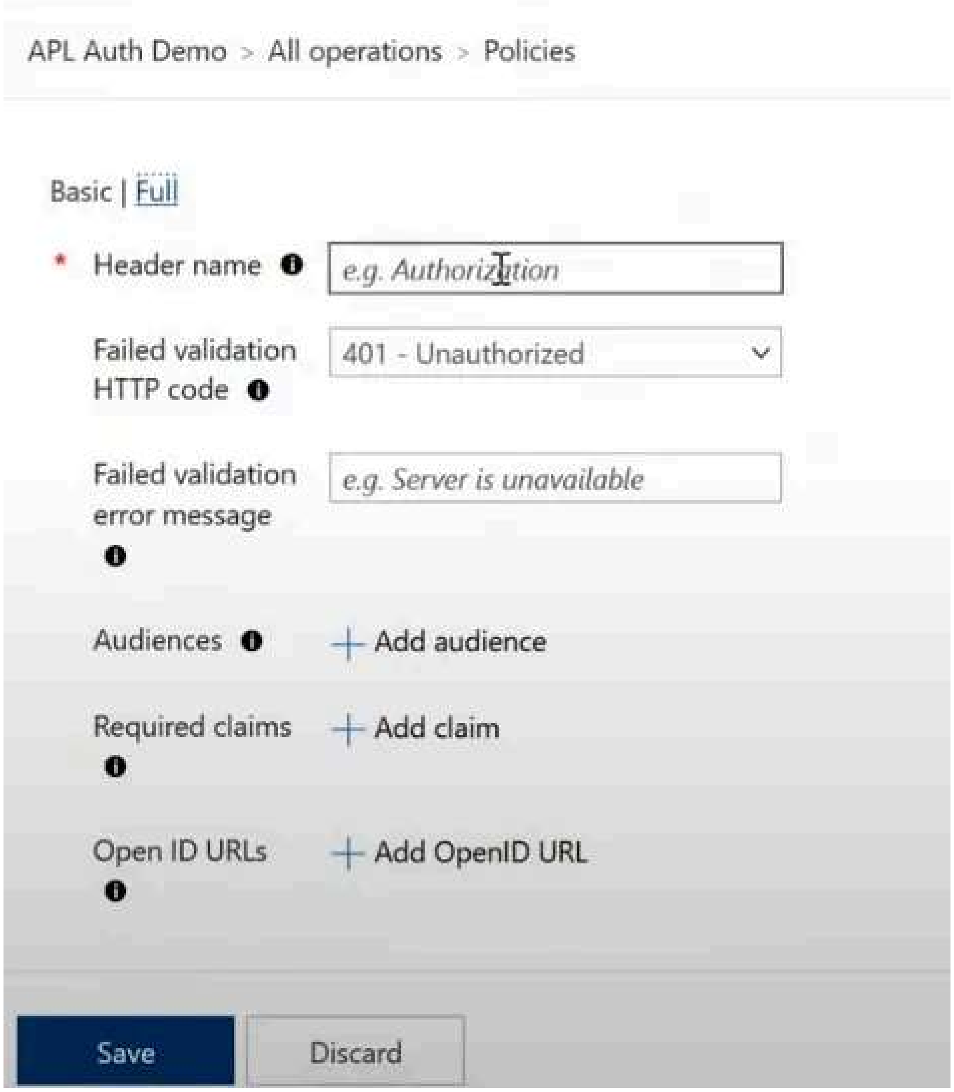
```
az account get-access-token — resource <application-id-uri>
```

Add Inbound Policy

- Go to the All APIs -> All operations and select the API. Then go to the design and click on Add policy.



- Add Header Name -> Authorization


APL Auth Demo > All operations > Policies

Basic | Full

* Header name: e.g. Authorization

Failed validation HTTP code: 401 - Unauthorized

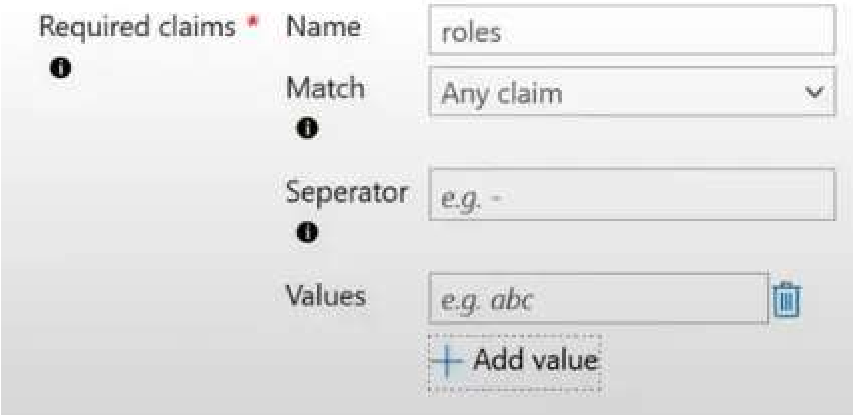
Failed validation error message: e.g. Server is unavailable

Audiences: Add audience

Required claims: Add claim

Open ID URLs: Add OpenID URL

- For claims: For values give the app role. and click on the save button.



Required claims *

Name: roles

Match: Any claim

Seperator: e.g. -

Values: e.g. abc

- Like this we can implement authorization for all operations or specific API endpoint.

Set Up Caching Policies

- Select the API for which you want to enable caching.
- Go to the “Design” tab of the API.
- Click on “All operations” or select a specific operation where you want to add caching.
- Click on “Inbound processing” and then “Add policy”.
- Select the “Cache” policy from the list.

```
<inbound>
  <cache-lookup vary-by-developer="false" vary-by-developer-groups="false" vary-
    <cache-key>redis-cache-key</cache-key>
    <expiration-interval>300</expiration-interval>
```

```
      <redis>
        <connection-string>@(context.Variables["redisConnectionString"])</connecti
      </redis>
    </cache-lookup>
  </inbound>
```

Best Practices

1. Secure Your APIs

- **Authentication and Authorization:** Use Azure AD or other OAuth providers to secure access to your APIs.
- **IP Restrictions:** Implement IP filtering to restrict access to your APIs from trusted IP addresses only.
- **CORS Policy:** Configure Cross-Origin Resource Sharing (CORS) to control which domains can access your APIs .

2. Use Virtual Networks

- **Connect your APIM instance to a virtual network to ensure internal APIs are not exposed to the internet and can only be accessed through the APIM gateway .**

3. Enable Application Insights:

- Integrate Application Insights to monitor and log API usage, performance, and errors. This helps in identifying and troubleshooting issues quickly .

4. Implement Rate Limiting and Throttling

- Use rate limiting and throttling policies to control the number of API requests from clients. This prevents abuse and ensures fair usage of your APIs .

5. Version Your APIs

Implement API versioning to manage changes and ensure backward compatibility. This allows clients to continue using the older versions of APIs while new versions are developed and deployed .

6. Use API Policies for Customization

- Leverage API Management policies to implement custom behaviors such as caching, transformations, and format conversions without changing the backend service .

7. Automate Deployments

- Use infrastructure as code (IaC) tools like Azure Resource Manager (ARM) templates, Terraform, or Azure CLI scripts to automate the deployment and management of your API Management service .

8. Monitor and Analyze API Usage

- Regularly monitor API usage metrics and analyze traffic patterns to optimize performance and identify potential issues. Use built-in analytics and logging features in APIM .

Azure Apim Api Management Api Gateway Best Practices

Written by Darshana Mihiran Edirisinghe

Follow

258 Followers · 28 Following

Full Stack Developer | Information Technology Consultant | Scrum Master | Tech Enthusiast

No responses yet

What are your thoughts?

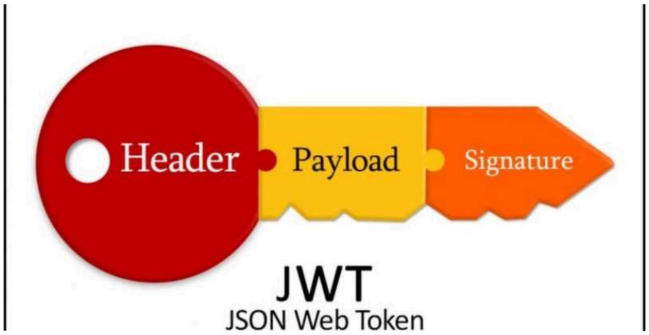
Respond

More from Darshana Mihiran Edirisinghe

Darshana Mihiran Edirisinghe

Azure Service Bus: Part 2— Sending and Receiving Messages

This is the 2nd article of Azure Service Bus. In this article, we'll explore Azure Service Bus...



Darshana Mihiran Edirisinghe

JWT authentication: Basics and best practices

In this article we are going to discuss about common use cases of JWT, How to impleme...

Jun 3, 2024

Feb 19, 2023 64 1

Darshana Mihiran Edirisinghe

Entity Framework performance improvement: [Section 1]: Differen...

Entity Framework Core is an ORM (Object Relational Mapping) framework that is...

Feb 17, 2023 95

Darshana Mihiran Edirisinghe

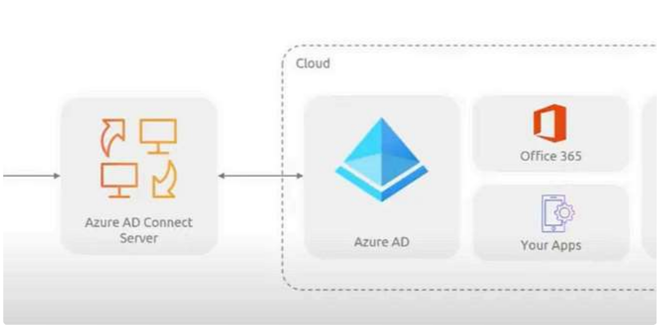
Task Parallel Library (TPL) in C#

TPL stands for "Task Parallel Library" in C#. It is a set of classes and APIs provided by...

Oct 3, 2023 28 2

See all from Darshana Mihiran Edirisinghe

Recommended from Medium

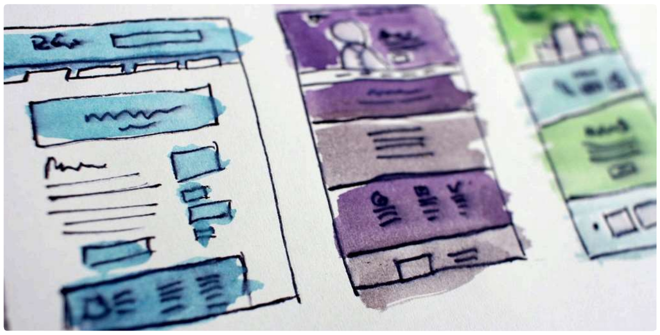


MOHD ABDUL JAMEEL

Azure AD Connect: Hybrid Identity Management

Syncing On-Premises Active Directory with Microsoft Entra ID

Oct 12, 2024 23



In Towards Dev by Eman Hassan

How to Use the No-Code Azure Developer Portal to Showcase Yo...

Handling API subscriptions and providing users with a portal to manage their accounts...

Sep 12, 2024 461 19

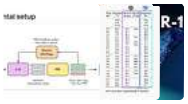
Lists

ChatGPT

21 stories · 954 saves

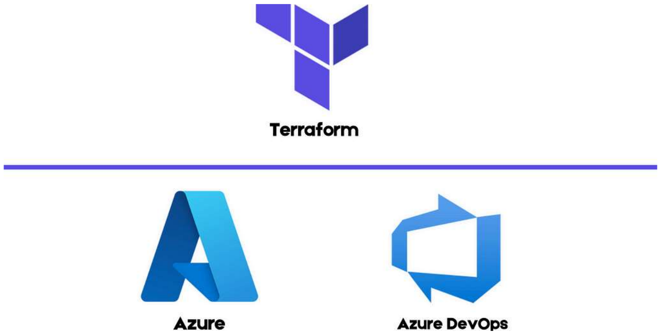
Generative AI Recommended Reading

52 stories · 1618 saves



Natural Language Processing

1899 stories · 1556 saves



Mücahid Özçelik

Feng Li

Deploying .NET Core Application on Azure App Service with...

How to use Terraform to deploy a fully-fledged cloud infrastructure on Azure App...

✦ Aug 11, 2024 👏 108

A study note of Azure APIM

This post will describe workflow from applications to LLM endpoints via APIM...

✦ Aug 19, 2024

statusCode

200

body

1. Publish quality content consistently to attract readers and increase engagement.

2. Engage with other Medium users by commenting on

riovtech

How to Implement OpenAI in Azure Logic Apps: A Comprehensive...

Hello! 👋 😁 Today, I conducted a few experiments on using the Azure OpenAI...

✦ Sep 3, 2024 👏 407 💬 5

How to Schedule Messages on WhatsApp /.Net with The Easy Guide
WhatsApp
WhatsApp

Wassenger

How to Schedule Messages on WhatsApp with C# (.NET): The...

In this tutorial, you will learn how to schedule the delivery of messages using the API.

Jan 8 👏 40 💬 2

See more recommendations