# Create an App registration in Microsoft Entra ID

## Introduction

This guide outlines the steps required to create an App Registration in the Azure portal and expose an API using the default Application ID URI. This process is essential for integrating your application with Microsoft Entra ID (Azure AD) and enabling secure authentication and authorization using OAuth 2.0.

## Prerequisites

* An active Azure subscription.
* Access to the Azure Portal.
* Permissions to create App Registrations (usually, you need to be an Microsoft Entra ID administrator).

## Step 1: Creating an App Registration in Azure

1. Log in to Azure Portal:
   * Open your browser and go to [Azure Portal](https://portal.azure.com/).
   * Use your credentials to log in.
2. Navigate to 'Microsoft Entra ID':
   * In the left-hand navigation pane, click on **Microsoft Entra ID**.
   * If you don’t see it immediately, search for **Microsoft Entra ID** in the search bar at the top of the Azure Portal.

A screenshot of a computer

Description automatically generated

1. **App Registrations:**
   * In the **Microsoft Entra ID** blade, click on **Manage** then **App registrations** from the left-hand menu.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. **Create New Registration:**
   * Click the **New registration** button at the top.

A screenshot of a computer

Description automatically generated

1. **Fill in the Registration Form:**
   * **Name:** Provide a meaningful name for your application (e.g., "NodiniteLogApi\_OAuth").
   * **Supported account types:** Select the appropriate option based on who will access the API:
     + **Accounts in this organizational directory only** (for single-tenant applications).
   * **Redirect URI:** This is optional for API-only applications. You can skip step and leave it empty.

A screenshot of a computer

Description automatically generated

1. **Click 'Register':**
   * After filling in the form, click **Register** to create your application.

## Step 2: Expose an API

1. Expose an API:
   * In the NodiniteLogApi\_OAuth App registration **Manage**, click **Expose an API** from the left-hand menu.
2. Add Application ID URI:
   * Under **Application ID URI**, click **Add**. The default value will be something like api://{Application (client) ID}. You can either accept this default or customize it.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



## Step 3: Creating a Client Secret (For App Authentication)

1. Certificates & Secrets:
   * In the NodiniteLogApi\_OAuth App registration **Manage**, click **Certificates & secrets** from the left-hand menu.
2. Create a New Client Secret
   * Under the **Client secrets** section, click **New client secret**.
3. Add Description and Expiration:
   * **Description:** Enter a description for the secret (e.g., "MyAppSecret").
   * **Expires:** Select an expiration period for the secret (options are 6 months, 12 months, or 24 months).
4. Add Secret:
   * Click **Add**. A new secret will be generated.
5. Copy the **Value** (Not the secret ID) Immediately:
   * After the secret is created, you will be shown the **secret value**. **Copy this value immediately**, as you will not be able to view it again later.
   * ***Note:*** *This secret will be used as the client secret in your Application for authentication requests.*

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Step 4: This step is to fix the invalid issuer problem

* + App Registration has a Manifest’s Json file for its configuration, In the NodiniteLogApi\_OAuth App registration **Manage**, click **Manifest** from the left-hand menu.
  + There is a field called **requestedAccessTokenVersion** set its value to 2.
  + ***Note: We need to do this extra step since when the value was set to null the default token format will be v1.0.***

A screenshot of a computer

Description automatically generated

# Install and configure secure LogApi manually

## Step 1: Create the Application Pool

A screenshot of a computer

Description automatically generated

Open the Internet Information Services (IIS) Manager

1. Click in the Application Pools
2. Right click to open the context menu and click “Add Application Pool…”

A screenshot of a computer

Description automatically generated

1. Give the application pool a name
2. Choose “No Managed Code” in the “.NET CLR Version” combo box
3. Click the “OK” button

A screenshot of a computer

Description automatically generated

1. Select the newly created Application Pool
2. Click the “Advanced Settings…” link

A screenshot of a computer

Description automatically generated

1. Find the “Identity” row and click on the “…” button

A screenshot of a computer

Description automatically generated

1. Select the “Custom account:” radio button
2. Click the “Set…” button

A screenshot of a login box

Description automatically generated

1. Add the service account and the password and click the “OK” button

A screenshot of a computer

Description automatically generated

1. Click the “OK" button

A screenshot of a computer

Description automatically generated

1. Click the “OK” button

## Step 2: Create the Web Site

A screenshot of a computer

Description automatically generated

1. Right click on the “Sites” folder
2. Click on the “Add Website…” button

A screenshot of a computer

Description automatically generated

1. Click on the “Select…” button

A screenshot of a computer application

Description automatically generated

1. Select the Application Pool you created in Step 1 on the “Application Pool:” combo box
2. Click the “OK” button

A screenshot of a computer

Description automatically generated

1. Add the site name
2. Add the location on disk where the files are stored
3. Select “https” in the “Type:” combo box
4. Add you host name in the “Host name:” text box
5. Select the SSL Certificate in the “SSL Certificate:” Combo box

A screenshot of a computer

Description automatically generated

1. Extract the content of “LogAPI.zip” to the “C:\Program Files\Nodinite\SecureLogApi” folder
2. Select the SecureLogApi
3. Click on the “Restart” button

A screenshot of a computer

Description automatically generated

You can browse to <https://[DOMAINNAME]/logapi/api/isalive> to check if the LogApi is running correctly

# Configure the Secure LogAPI

**Step 1: Configure Authentication for the Nodinite LogApi**

* Navigate to the installation folder of the LogApi.
* Open the `appsettings.json` file.
* Set the following properties:
  + `**AuthenticationType**` to "OAuth"
  + `**ValidAudience**` to your "ClientId"
  + `**services:idp:http:0**` to the OpenID Connect metadata document from Azure.

A screenshot of a computer

Description automatically generated

**A screenshot of a computer

Description automatically generated**

# Configure the PickupService to use the secured LogAPI

## Step 1: Configure Authentication for the Nodinite PickupService

* Navigate to the installation folder of the **Logging Agent - Pickup Service**.
* Open the `settings.json` file.
* Set the following properties under the **LogAPI** section:
  + `**LogApiBaseUrl**` to the base url for the Secure LogAPI https://[DOMAINNAME]/LogAPI
  + ` **LogApiIsProtected**` to **true**
  + ` **ClientId**` to your "ClientId"
  + ` **ClientSecret**` to your “ClientSecret”.
  + ` **Scope**` to the Application ID URI plus /.default for example “api://de95ae0d-9158-42e8-a444-5ea0cffd3ec6/.default”
  + ` **IdpTokenEndpoint**`to the OAuth 2.0 token endpoint (v2) in your app registration

A screenshot of a computer

Description automatically generated

## Step 2: Restart the PickupService

A screenshot of a computer

Description automatically generated

1. In the Services dialog right click on the “Nodinite Logging Agent – Pickup LogEvents Service”
2. Click on “Restart”

### Example of the PickupService settings.json

{

  "Environment": "Test",

  "Debug": false,

  "Version": "3.7",

  "CultureInfo": "sv-SE",

  "Folders": [

/\* \*\*File Example\*\*

    {

      "Enabled": true,

      "LogAPI": {

        "LogApiBaseUrl": "http://localhost/Nodinite/Dev/LogAPI/",

        "LogApiIsProtected": false,

        "ClientId": "",

        "ClientSecret": "",

        "Scope": "",

        "IdpTokenEndpoint": ""

      },

      "UseConfigurationDatabase": false,

      "ConfigurationDatabaseConnectionString": "Server=localhost;Database=NodiniteConfig\_Dev;Integrated Security=SSPI;Connection Timeout=60;TrustServerCertificate=true",

      "IsDFSShare": false,

      "Folder": "C:\\Temp\\LogAgent",

      "BackoutFolder": "C:\\Temp\\BackoutFiles",

      "RemoveEmptyFiles": true,

      "Filter": "\*.json",

      "IncludeChildFolders": false,

      "ExcludeChildFolders": [

        {

          "Name": "C:\\Temp\\LogAgent\\NotMe"

        }

      ]

    }

\*/

  ],

  "ActiveMQs": [

  /\* \*\*Active MQ Example\*\*

    {

      "Enabled": true,

      "LogAPI": {

        "LogApiBaseUrl": "http://localhost/Nodinite/Dev/LogAPI/",

        "LogApiIsProtected": false,

        "ClientId": "",

        "ClientSecret": "",

        "Scope": "",

        "IdpTokenEndpoint": ""

      },

      "UseConfigurationDatabase": false,

      "ConfigurationDatabaseConnectionString": "Server=localhost;Database=NodiniteConfig\_Dev;Integrated Security=SSPI;Connection Timeout=60;TrustServerCertificate=true",

      "ActiveConsumers": 4,

      "ConnectionString": "failover:(tcp://127.0.0.1:61616,tcp://localhost:61616)?randomize=false&timeout=1000&startupMaxReconnectAttempts=2&maxReconnectDelay=2000",

      "Username": "system",

      "Password": "manager",

      "Queue": "Nodinite.LogAgent.PickupService.Test",

      "DeadLetterQueue": "Nodinite.LogAgent.PickupService.Test.DLQ",

      "ReplaceRules": []

    }

\*/

  ],

  "AMQPs": [

/\* \*\*AMQPs Example\*\*

                      {

      "Enabled": true,

      "LogAPI": {

        "LogApiBaseUrl": "http://localhost/Nodinite/Dev/LogAPI/",

        "LogApiIsProtected": false,

        "ClientId": "",

        "ClientSecret": "",

        "Scope": "",

        "IdpTokenEndpoint": ""

      },

      "UseConfigurationDatabase": false,

      "ConfigurationDatabaseConnectionString": "Server=localhost;Database=NodiniteConfig\_Dev;Integrated Security=SSPI;Connection Timeout=60;TrustServerCertificate=true",

      "Host": "localhost",

      "Port": "5672",

      "Username": "AMQPs",

      "Password": "AMQPs",

      "Queue": "a1",

      "DeadLetterQueue": "DLQ",

      "ReplaceRules": []

    }

\*/

],

  "MSMQs": [

  /\* \*\*MSMQ MQ Example\*\*

    {

      "Enabled": true,

      "LogAPI": {

        "LogApiBaseUrl": "http://localhost/Nodinite/Dev/LogAPI/",

        "LogApiIsProtected": false,

        "ClientId": "",

        "ClientSecret": "",

        "Scope": "",

        "IdpTokenEndpoint": ""

      },

      "UseConfigurationDatabase": false,

      "ConfigurationDatabaseConnectionString": "Server=localhost;Database=NodiniteConfig\_Dev;Integrated Security=SSPI;Connection Timeout=60;TrustServerCertificate=true",

      "Server": ".",

      "Queue": "Nodinite.LogAgent.PickupService.Test",

      "DeadLetterQueue": "Nodinite.LogAgent.PickupService.Test.DLQ",

      "ActiveConsumers": 4,

      "ReplaceRules": []

    }

 \*/

  ],

"ServiceBusQueues": [

 /\* \*\*ServiceBusQueue Example\*\*

    {

      "Enabled": true,

      "LogAPI": {

        "LogApiBaseUrl": "http://localhost/Nodinite/Dev/LogAPI/",

        "LogApiIsProtected": false,

        "ClientId": "",

        "ClientSecret": "",

        "Scope": "",

        "IdpTokenEndpoint": ""

      },

      "UseConfigurationDatabase": false,

      "ConfigurationDatabaseConnectionString": "Server=localhost;Database=NodiniteConfig\_Dev;Integrated Security=SSPI;Connection Timeout=60;TrustServerCertificate=true",

      "ConnectionString": "Endpoint=sb://nodinite.com/servicebus.windows.net/;SharedAccessKeyName=[keyname];SharedAccessKey=[key]",

      "Queue": "pickup",

      "ReplaceRules": []

    }

 \*/

  ],

  "PostgreSQLs": [

/\* \*\*PostgreSQL Example\*\*

    {

      "Enabled": true,

      "LogAPI": {

        "LogApiBaseUrl": "http://localhost/Nodinite/Dev/LogAPI/",

        "LogApiIsProtected": false,

        "ClientId": "",

        "ClientSecret": "",

        "Scope": "",

        "IdpTokenEndpoint": ""

      },

      "UseConfigurationDatabase": false,

      "ConfigurationDatabaseConnectionString": "Server=localhost;Database=NodiniteConfig\_Dev;Integrated Security=SSPI;Connection Timeout=60;TrustServerCertificate=true",

      "ConnectionString": "Server=name.databases.appdomain.cloud;Port=31902;Database=ibmclouddb;User Id=ibm\_cloud\_user;Password=yoursecretpassword;SSLMode=Prefer;Trust Server Certificate=true",

      "MaxRowsLimit": 500,

      "ReplaceRules": []

    }

\*/

],

  "SQLServers": [

/\* \*\*SQLServer Example\*\*

    {

      "Enabled": true,

      "LogAPI": {

        "LogApiBaseUrl": "http://localhost/Nodinite/Dev/LogAPI/",

        "LogApiIsProtected": false,

        "ClientId": "",

        "ClientSecret": "",

        "Scope": "",

        "IdpTokenEndpoint": ""

      },

      "UseConfigurationDatabase": false,

      "ConfigurationDatabaseConnectionString": "Server=localhost;Database=NodiniteConfig\_Dev;Integrated Security=SSPI;Connection Timeout=60;TrustServerCertificate=true",

      "ConnectionString": "Server=localhost;Database=NodiniteDBWithLogEventsTable;Integrated Security=SSPI;Connection Timeout=60;TrustServerCertificate=true",

      "MaxRowsLimit": 500,

      "ReplaceRules": []

    }

\*/

],

  "EventHubs": [

/\* \*\*Eventhubs Example\*\*

                      {

      "Enabled": true,

      "LogAPI": {

        "LogApiBaseUrl": "http://localhost/Nodinite/Dev/LogAPI/",

        "LogApiIsProtected": false,

        "ClientId": "",

        "ClientSecret": "",

        "Scope": "",

        "IdpTokenEndpoint": ""

      },

      "WebProxy": "",

      "UseConfigurationDatabase": false,

      "ConfigurationDatabaseConnectionString": "Server=localhost;Database=NodiniteConfig\_Dev;Integrated Security=SSPI;Connection Timeout=60;TrustServerCertificate=true",

      "EventHubConnectionString": "Endpoint=sb://nodinite-dev.servicebus.windows.net/;SharedAccessKeyName=[keyname];SharedAccessKey=[key];EntityPath=nodinitelogevents",

      "EnableAmqpLinkRedirect": true,

      "EventHubName": null,

      "StorageContainerName": null,

      "StorageAccountName": null,

      "StorageAccountKey": null,

      "ReplaceRules": []

    }

\*/

],

  "AnypointMQs": [

/\* \*\*AnypointMQs Example\*\*

                      {

      "Enabled": true,

      "LogAPI": {

        "LogApiBaseUrl": "http://localhost/Nodinite/Dev/LogAPI/",

        "LogApiIsProtected": false,

        "ClientId": "",

        "ClientSecret": "",

        "Scope": "",

        "IdpTokenEndpoint": ""

      },

      "UseConfigurationDatabase": false,

      "ConfigurationDatabaseConnectionString": "Server=localhost;Database=NodiniteConfig\_Dev;Integrated Security=SSPI;Connection Timeout=60;TrustServerCertificate=true",

      "ActiveConsumers": 1,

      "Username": "",

      "Password":"",

    "UseConnectedApp": true,

    "IsEuPanel": false,

    "ClientId": "",

      "ClientSecret":"",

      "Environment": "",

      "EnvironmentId" : "",

      "OrganizationId": "",

      "RegionId": "eu-west-1",

      "Queue": "",

      "DeadLetterQueue": "",

      "ReplaceRules": null

    }

\*/

],

 "BlobContainers": [

/\* \*\*BlobContainers Example\*\*

    {

      "Enabled": true,

      "LogAPI": {

        "LogApiBaseUrl": "http://localhost/Nodinite/Dev/LogAPI/",

        "LogApiIsProtected": false,

        "ClientId": "",

        "ClientSecret": "",

        "Scope": "",

        "IdpTokenEndpoint": ""

      },

      "UseConfigurationDatabase": false,

      "ConfigurationDatabaseConnectionString": null,

      "BlobStorageConnectionString": "DefaultEndpointsProtocol=https;AccountName=xyz;AccountKey=xyz==;EndpointSuffix=core.windows.net",

      "ContainerName": "nodinitelogevents",

      "BackoutContainerName": "nodinitelogeventsbackout",

      "DisableCheckAccessRights": false,

      "RemoveEmptyBlobs": false,

      "ReplaceRules": [

        {

          "Name": "Fix Endpoint real customer id to {customerId}",

          "ReplaceName": false,

          "ReplaceUri": true,

          "Pattern": "/([0-9]{4,})$",

          "Group1Value": "{customerId}"

        }

      ]

    }

  \*/

  ]

}