Dileepa Mabulage
Posted on Jan 30, 2023

💖 1

Access Azure KeyVault Secrets Through Nodejs Application

#javascript #node #azure #keyvault

Azure Key Vault is a cloud-based service that allows users to securely store and manage sensitive information, such as passwords, keys, and certificates. This allows for a more secure and efficient way to manage and access sensitive information in a cloud environment.

In this article, we will discuss how to access these secrets through a Node.js application. We will cover how to set up an Azure Key Vault, how to authenticate with it, and how to retrieve and use the secrets in your application. By the end of this article, you will have a better understanding of how to use Azure Key Vault to secure and manage your application's sensitive information.

Table of Contents

1. Create Nodejs server

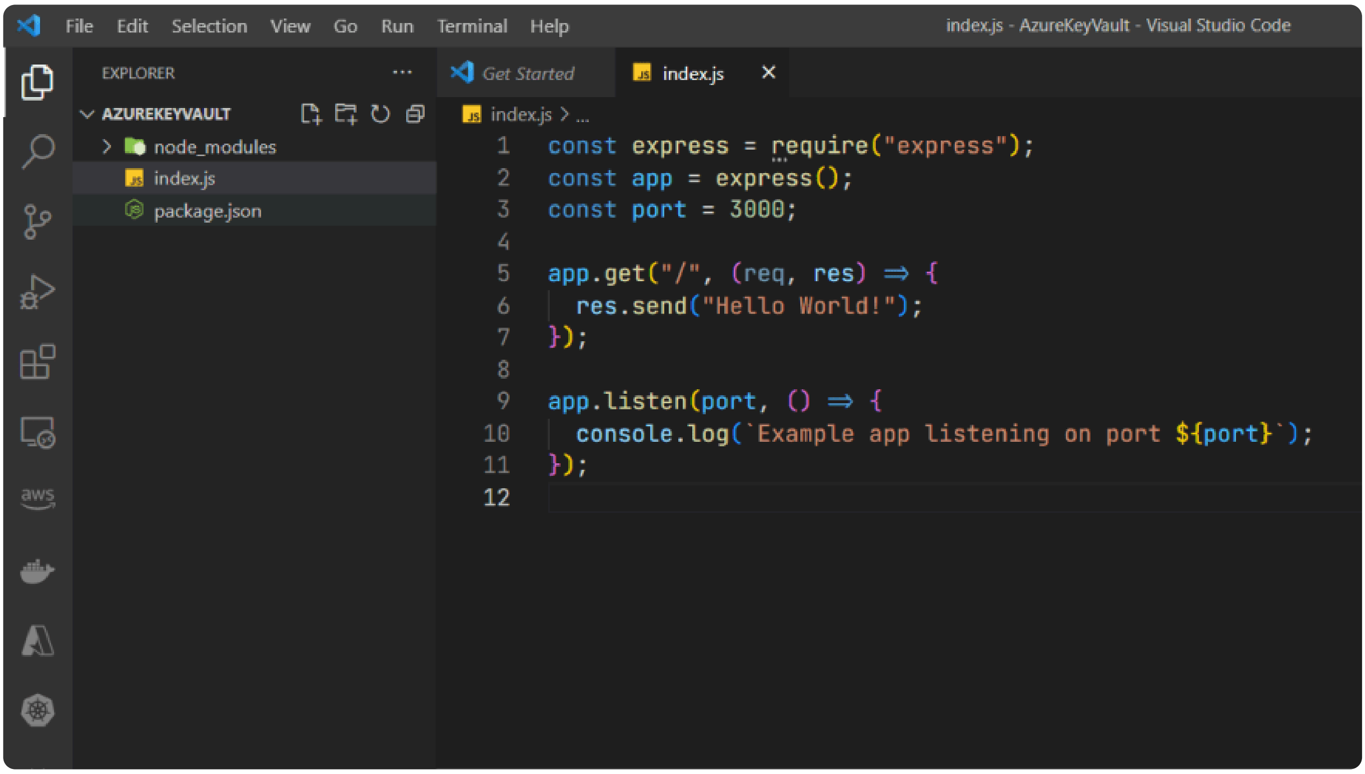
1. Create the directory and run `npm init -y` in the command prompt
2. Open that directory in VSCode using typing `code .` in the command prompt
3. Open vs code terminal and install the following

```
npm install express --save
npm install nodemon --save-dev
```

Nodemon is a tool that automatically restarts a Node.js application when changes are made to the code. This can save developers time and effort by eliminating the need to manually stop and start the application each time a change is made.

1. Create an index.js file and paste the following code

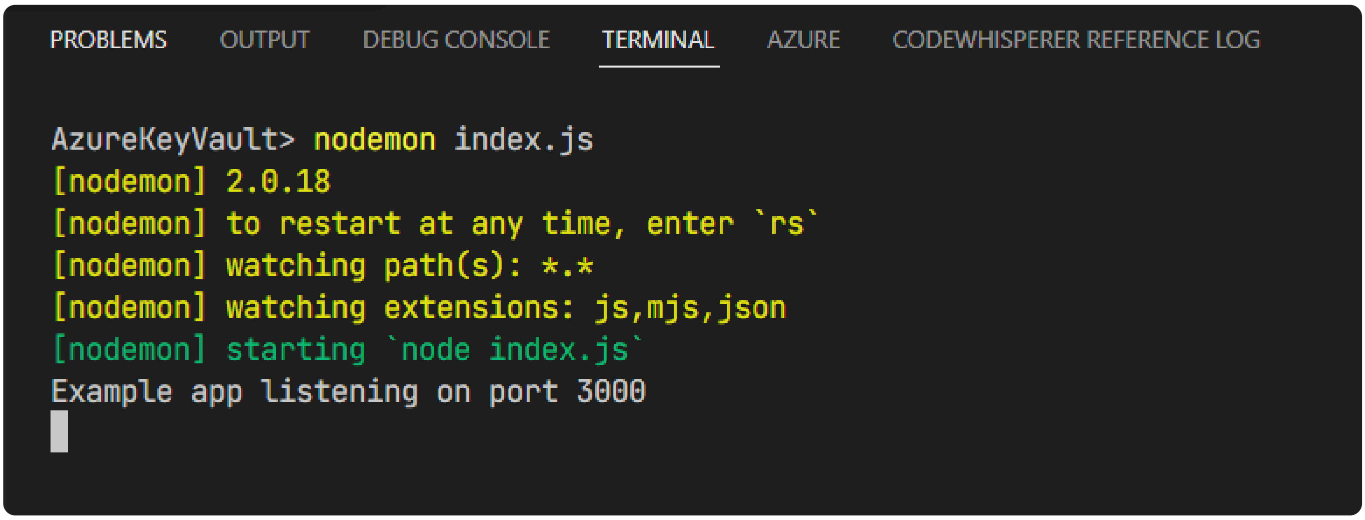
```
const express = require('express');
const app = express();
const port = 3000;

app.get('/', (req, res) => {
  res.send('Hello World!');
});

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`);
});
```



File Edit Selection View Go Run Terminal Help
index.js - AzureKeyVault - Visual Studio Code
EXPLORER
AZUREKEYVAULT
node_modules
index.js
package.json
Get Started
index.js
index.js > ...

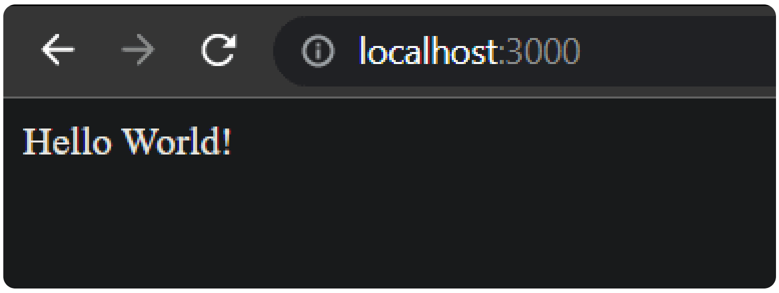
```
const express = require("express");
const app = express();
const port = 3000;

app.get("/", (req, res) => {
  res.send("Hello World!");
});

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`);
});
```

1. In your terminal (which should be in the project directory), type `nodemon index.js` and hit the Enter button.

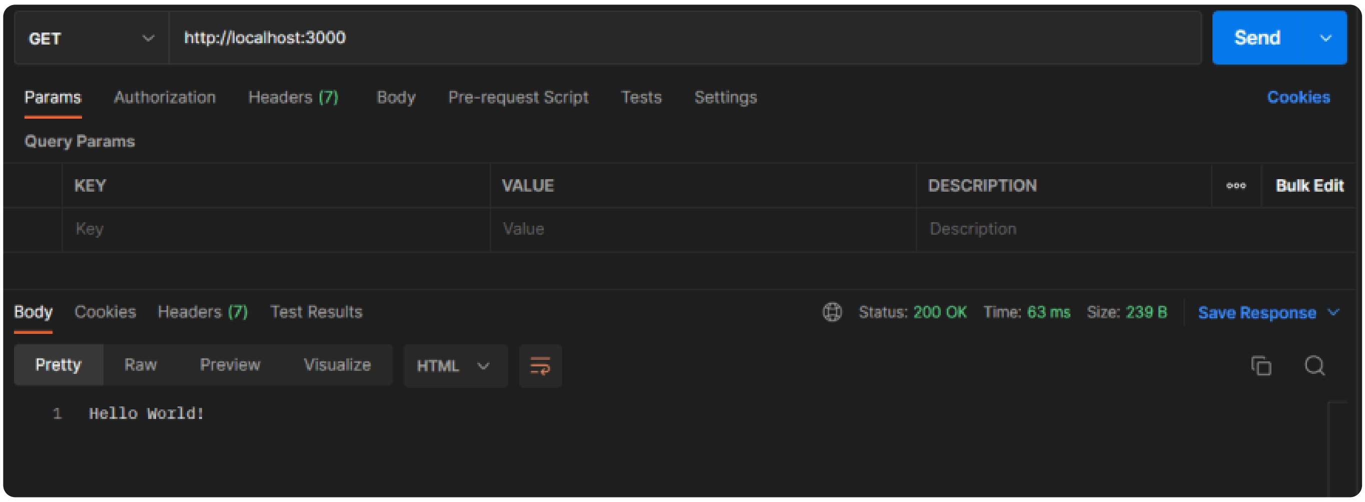


```
AzureKeyVault> nodemon index.js
[nodemon] 2.0.18
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
Example app listening on port 3000
```

1. Open a new tab in postman or any web browser and the address bar, type http://localhost:3000, and hit the Enter button



Hello World!



GET http://localhost:3000

Status: 200 OK Time: 63 ms Size: 239 B

```
Hello World!
```

Now the Node server is up and running...

2. Create Azure key vault

1. Sign in to the Azure portal at https://portal.azure.com
2. To set up a Key Vault in Azure:

3. Open the Azure portal and select "Create a resource" from the menu or Home page.

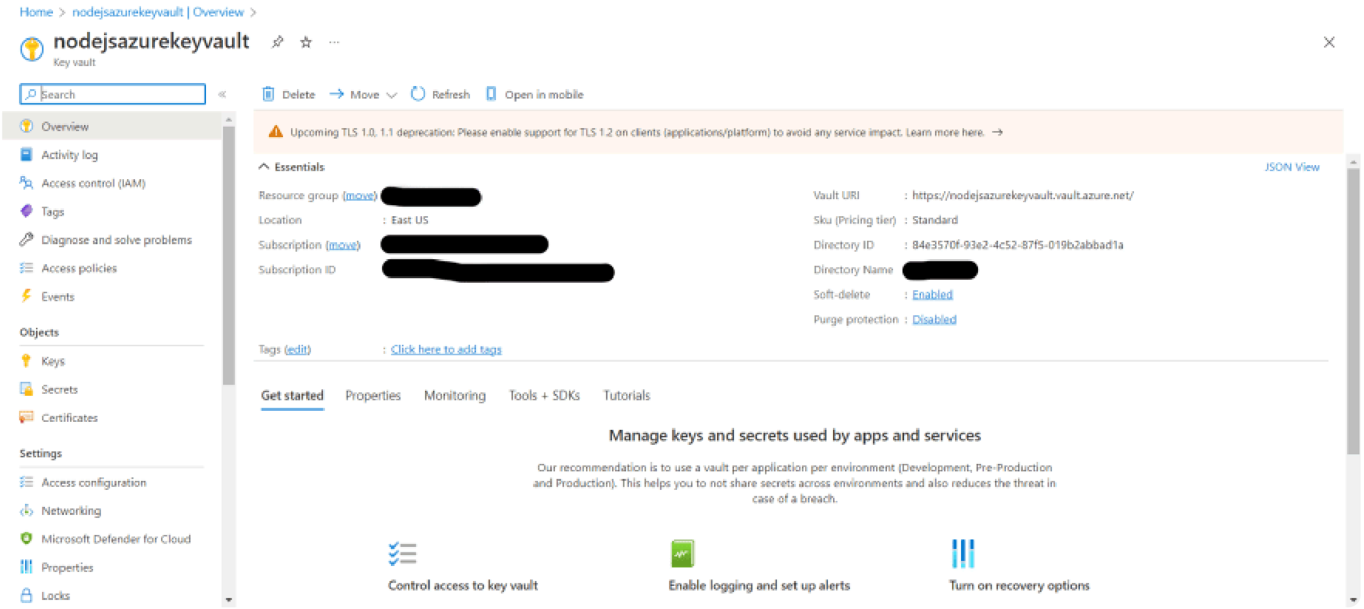
4. Search for "Key Vault" and select it from the results.

5. Click on "Create" in the Key Vault section.

6. In the "Create key vault" section, enter a unique name for the vault (e.g. "nodejsazurekeyvault") *A vault's name must be between 3-24 alphanumeric characters. The name must begin with a letter, end with a letter or digit, and not contain consecutive hyphens*, select a subscription and create a new resource group.

7. Pick a location and keep the other options unchanged.

8. Click on "Create" to finalize the setup.



Upcoming TLS 1.0, 1.1 deprecation: Please enable support for TLS 1.2 on clients (applications/platform) to avoid any service impact. Learn more here. →

Resource group (move) [redacted]
Location : East US
Subscription (move) [redacted]
Subscription ID [redacted]

Vault URI : https://nodejsazurekeyvault.vault.azure.net/
Sku (Pricing tier) : Standard
Directory ID : 84e3570f-93e2-4c52-87f5-019b2abbad1a
Directory Name [redacted]
Soft-delete : Enabled
Purge protection : Disabled

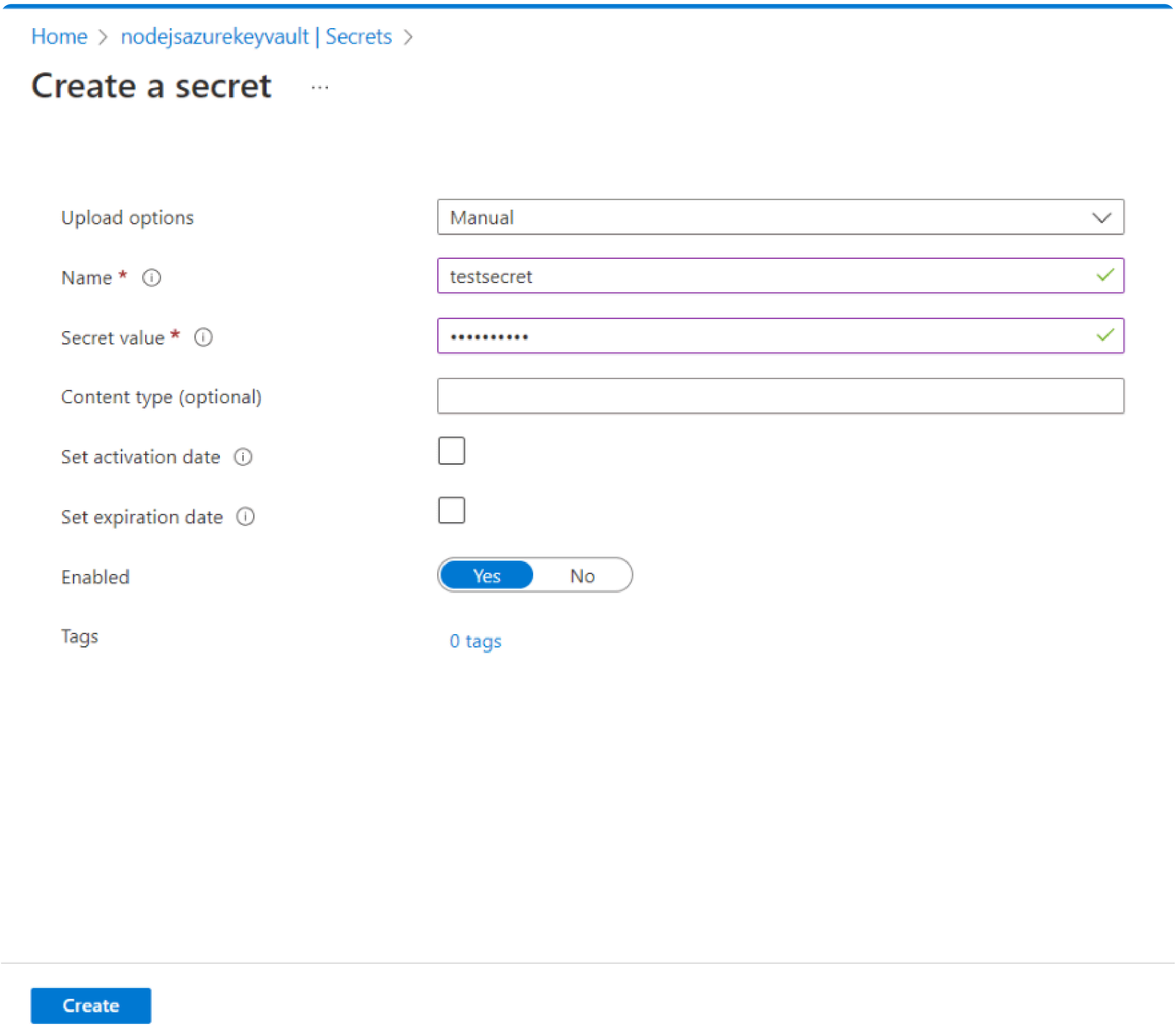
Tags (edit) : Click here to add tags

Manage keys and secrets used by apps and services

Our recommendation is to use a vault per application per environment (Development, Pre-Production and Production). This helps you to not share secrets across environments and also reduces the threat in case of a breach.

3. Add secrets to key Vault

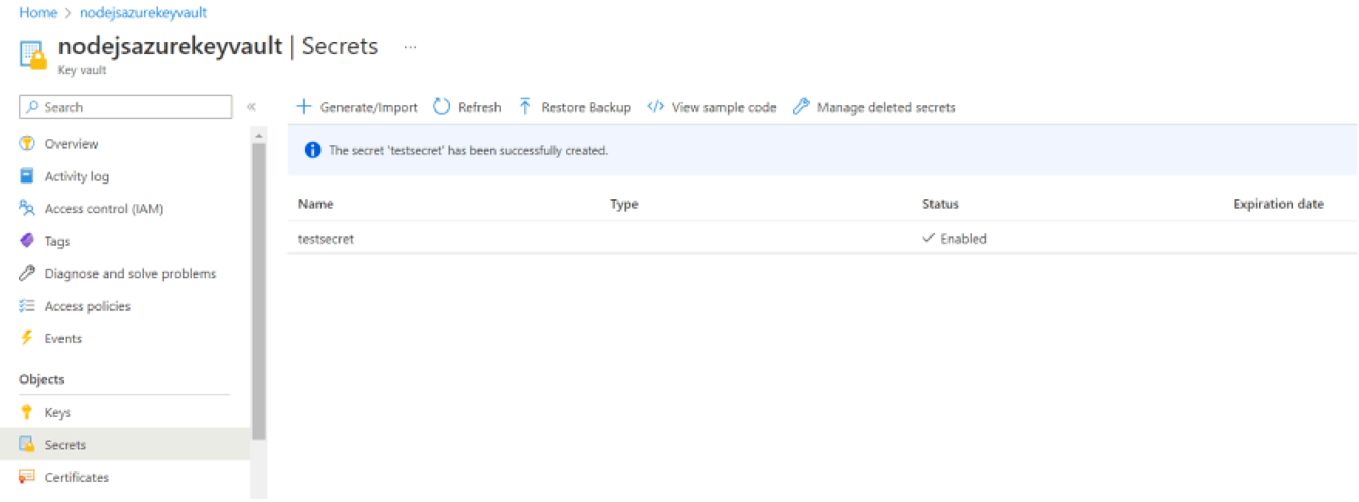
1. Click secrets in the left panel
2. Click Generate/Import at top of the page
3. Add a secret name, and value
4. Toggle enables to yes
5. Click Create



Home > nodejsazurekeyvault | Secrets >

Create a secret ···

Upload options	Manual
Name * ⓘ	testsecret ✓
Secret value * ⓘ	•••••••••• ✓
Content type (optional)	
Set activation date ⓘ	□
Set expiration date ⓘ	□
Enabled	[✓ Yes] No
Tags	0 tags



Home > nodejsazurekeyvault

nodejsazurekeyvault | Secrets ···

Key vault

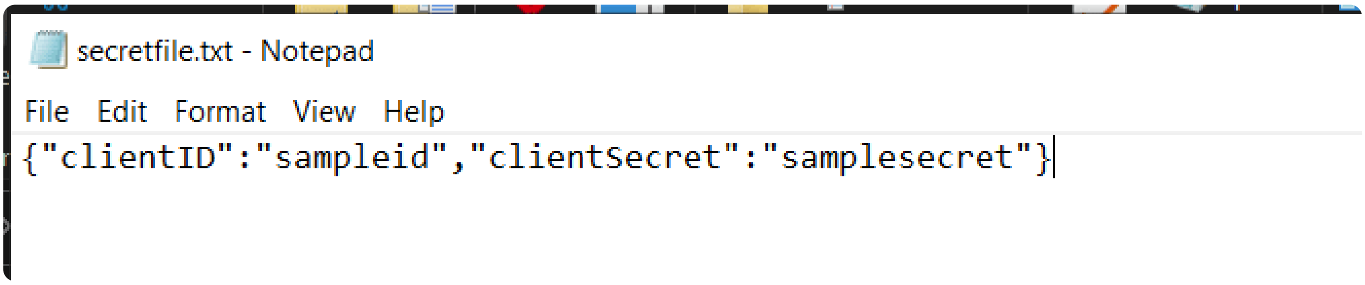
The secret 'testsecret' has been successfully created.

Name	Type	Status	Expiration date
testsecret		✓ Enabled	

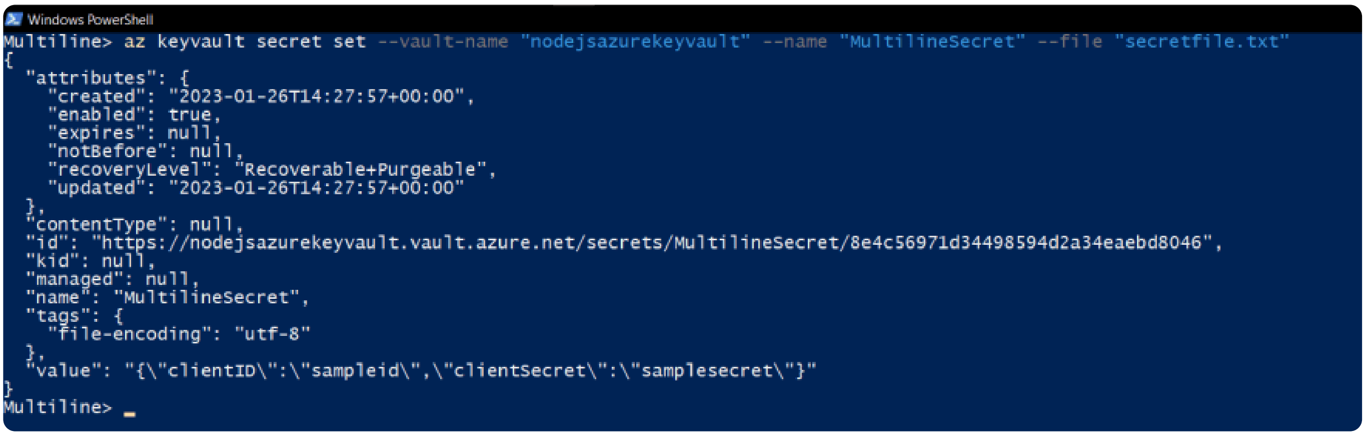
4. Add secrets from CLI

1. Install Azure CLI [Download](#)
2. Run these commands in the PowerShell window

```
az login
az keyvault secret set --vault-name "<your-unique-keyvault-name>" --name "Mul
```



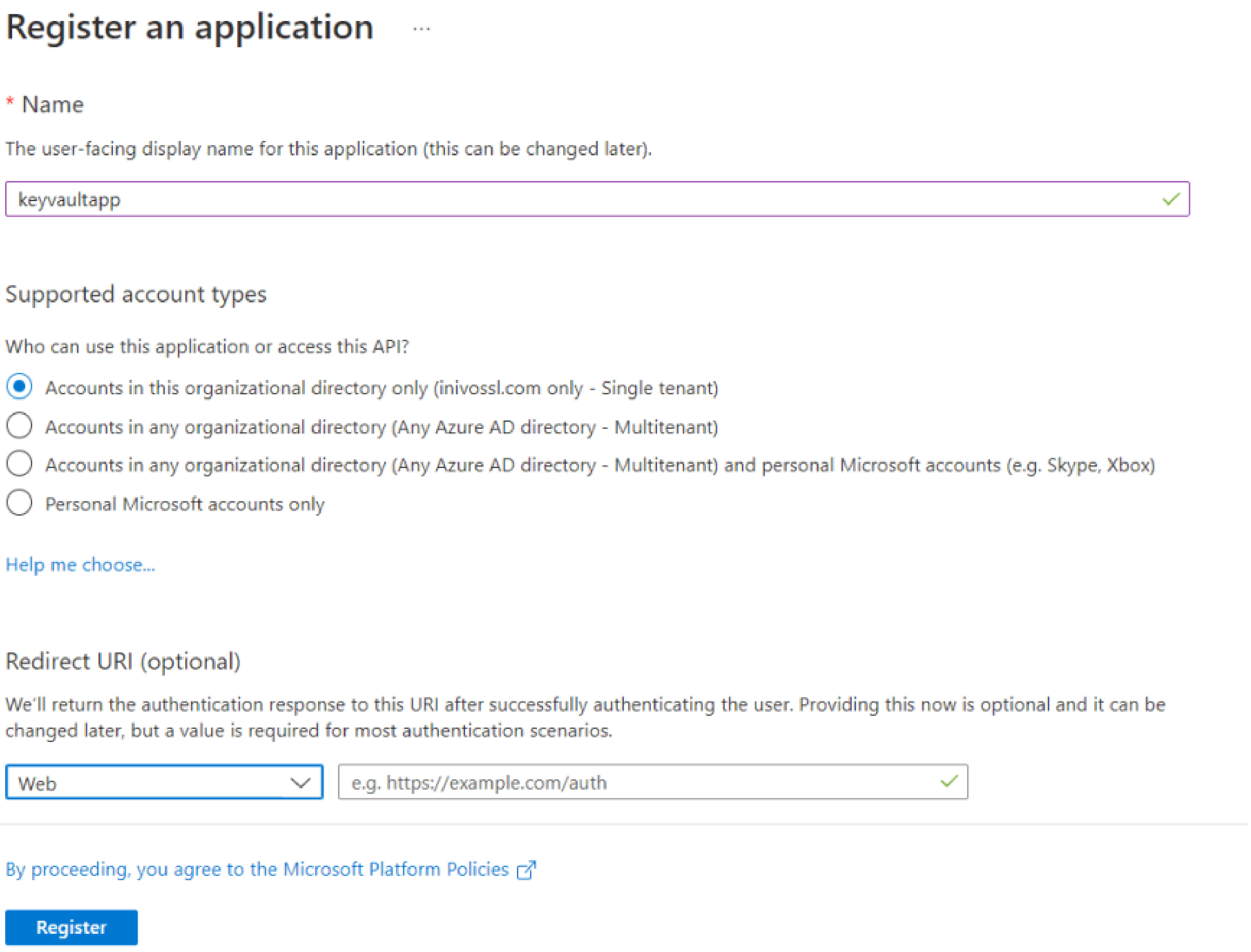
```
{"clientID":"sampleid","clientSecret":"samplesecret"}
```



```
Multiline> az keyvault secret set --vault-name "nodejsazurekeyvault" --name "MultilineSecret" --file "secretfile.txt"
{
  "attributes": {
    "created": "2023-01-26T14:27:57+00:00",
    "enabled": true,
    "expires": null,
    "notBefore": null,
    "recoveryLevel": "Recoverable+Purgeable",
    "updated": "2023-01-26T14:27:57+00:00"
  },
  "contentType": null,
  "id": "https://nodejsazurekeyvault.vault.azure.net/secrets/MultilineSecret/8e4c56971d34498594d2a34eaebd8046",
  "kid": null,
  "managed": null,
  "name": "MultilineSecret",
  "tags": {
    "file-encoding": "utf-8"
  },
  "value": "{\"clientID\":\"sampleid\",\"clientSecret\":\"samplesecret\"}"
}
Multiline> _
```

5. Register the app in Azure Active Directory

1. Navigate to Azure Active Directory
2. Click App registrations on the left panel
3. Click New Registration
4. Enter the app name and platform to Web
5. Register



Register an application ...

* Name

The user-facing display name for this application (this can be changed later).

keyvaultapp ✓

Supported account types

Who can use this application or access this API?

● Accounts in this organizational directory only (inivossl.com only - Single tenant)

○ Accounts in any organizational directory (Any Azure AD directory - Multitenant)

○ Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)

○ Personal Microsoft accounts only

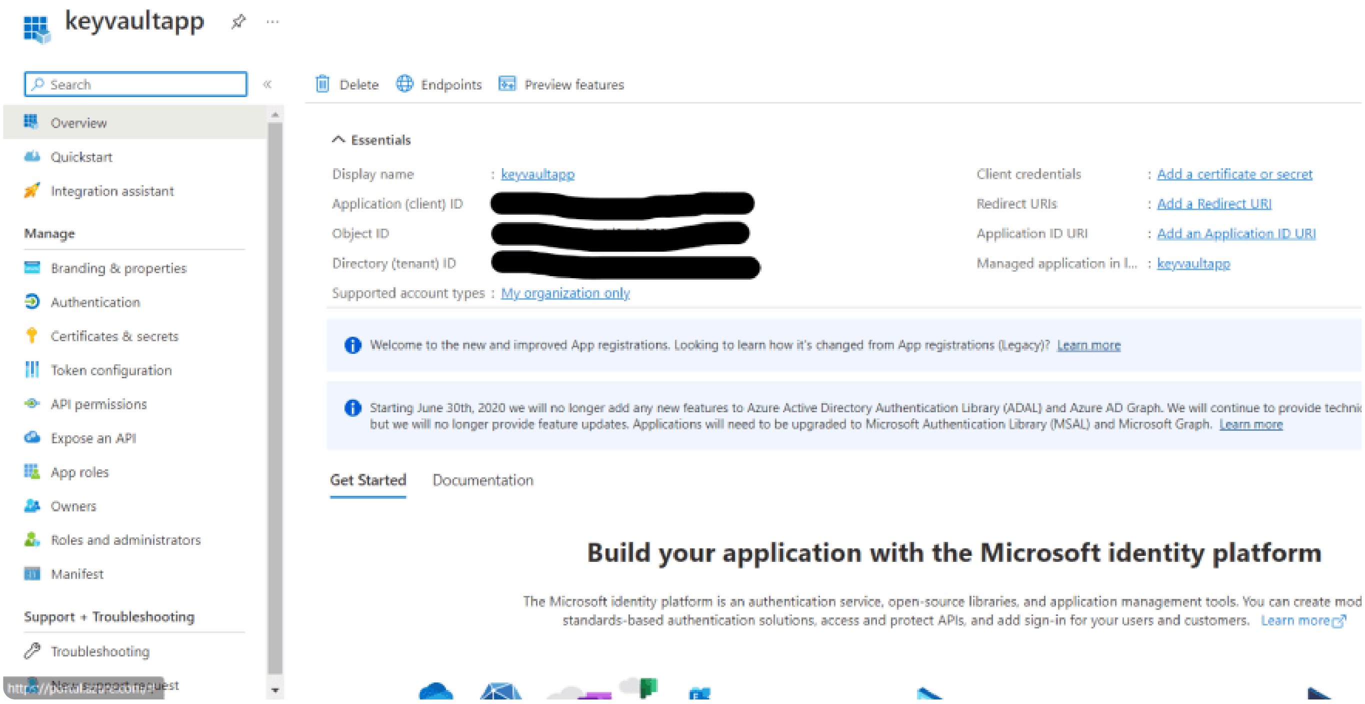
Help me choose...

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Web | e.g. https://example.com/auth ✓

By proceeding, you agree to the Microsoft Platform Policies



keyvaultapp

Essentials

Display name : keyvaultapp

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 l... : keyvaultapp

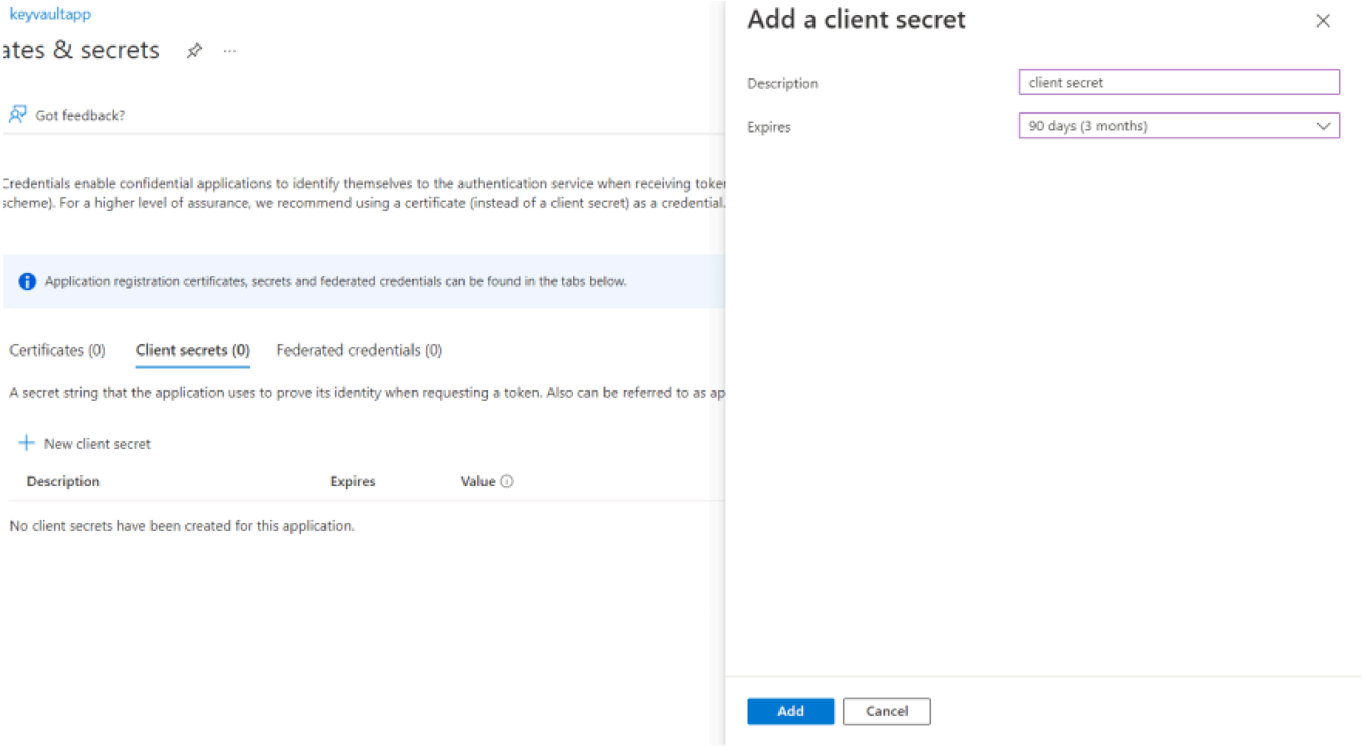
Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? Learn more

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

Build your application with the Microsoft identity platform

The Microsoft identity platform is an authentication service, open-source libraries, and application management tools. You can create mod standards-based authentication solutions, access and protect APIs, and add sign-in for your users and customers. Learn more

1. Click certificates and secrets
2. New client's secret
3. Add a description and set the expiry date
4. Add
5. Copy the value and keep it for future



keyvaultapp

ates & secrets

Got feedback?

Credentials enable confidential applications to identify themselves to the authentication service when receiving toker scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Application registration certificates, secrets and federated credentials can be found in the tabs below.

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as ap

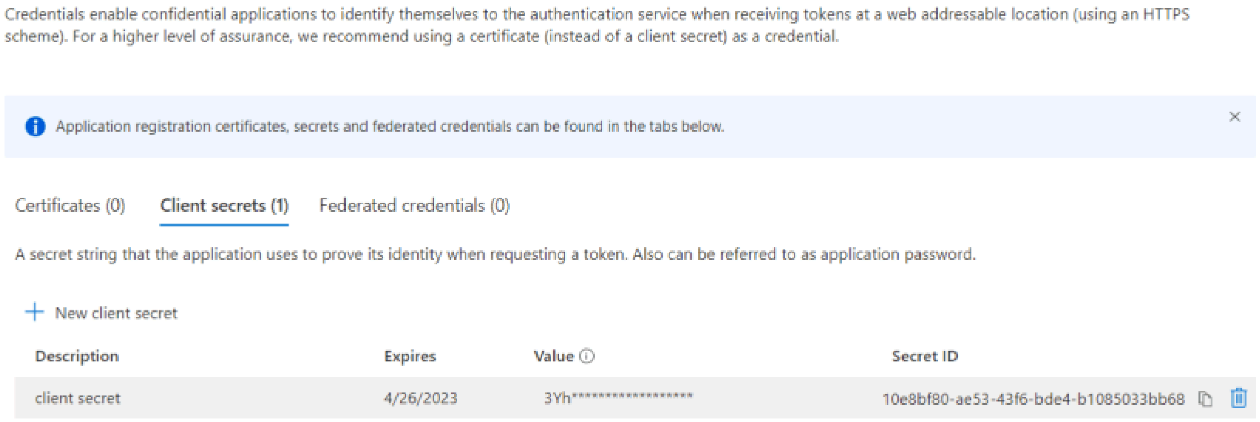
Description	Expires	Value ⓘ

No client secrets have been created for this application.

Add a client secret

Description: client secret

Expires: 90 days (3 months)



Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

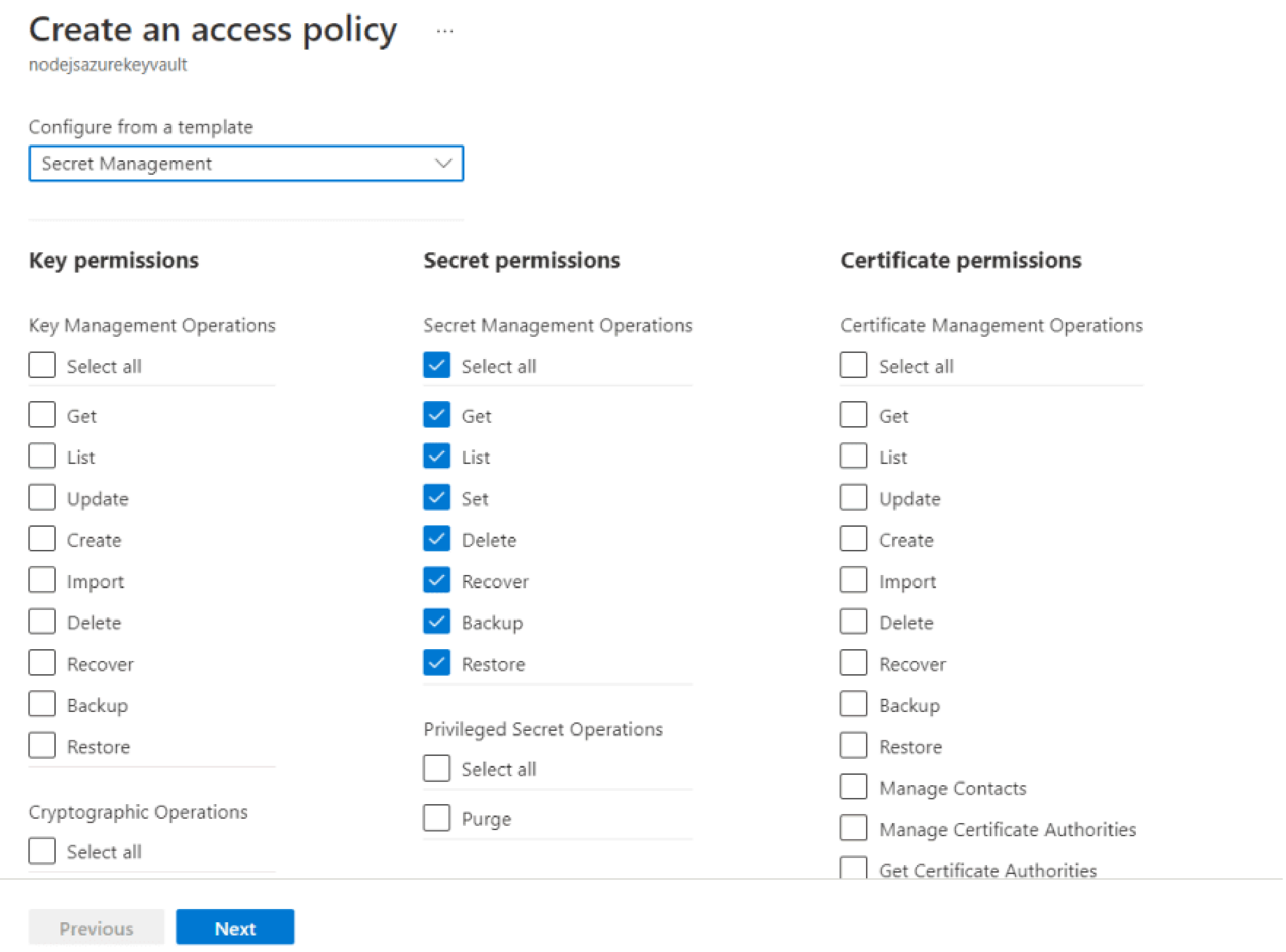
Application registration certificates, secrets and federated credentials can be found in the tabs below.

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

Description	Expires	Value ⓘ	Secret ID
client secret	4/26/2023	3Yh******************	10e8bf80-ae53-43f6-bde4-b1085033bb68

6. Add app to key Vault

1. Navigate to Key Vault
2. Click Access Policies in the left panel
3. Create
4. Select Secret Management from the template dropdown



Create an access policy ...

nodejsazurekeyvault

Configure from a template

Secret Management

Key permissions

Key Management Operations

□ Select all

□ Get
□ List
□ Update
□ Create
□ Import
□ Delete
□ Recover
□ Backup
□ Restore

Cryptographic Operations

□ Select all

Secret permissions

Secret Management Operations

☑ Select all

☑ Get
☑ List
☑ Set
☑ Delete
☑ Recover
☑ Backup
☑ Restore

Privileged Secret Operations

□ Select all

□ Purge

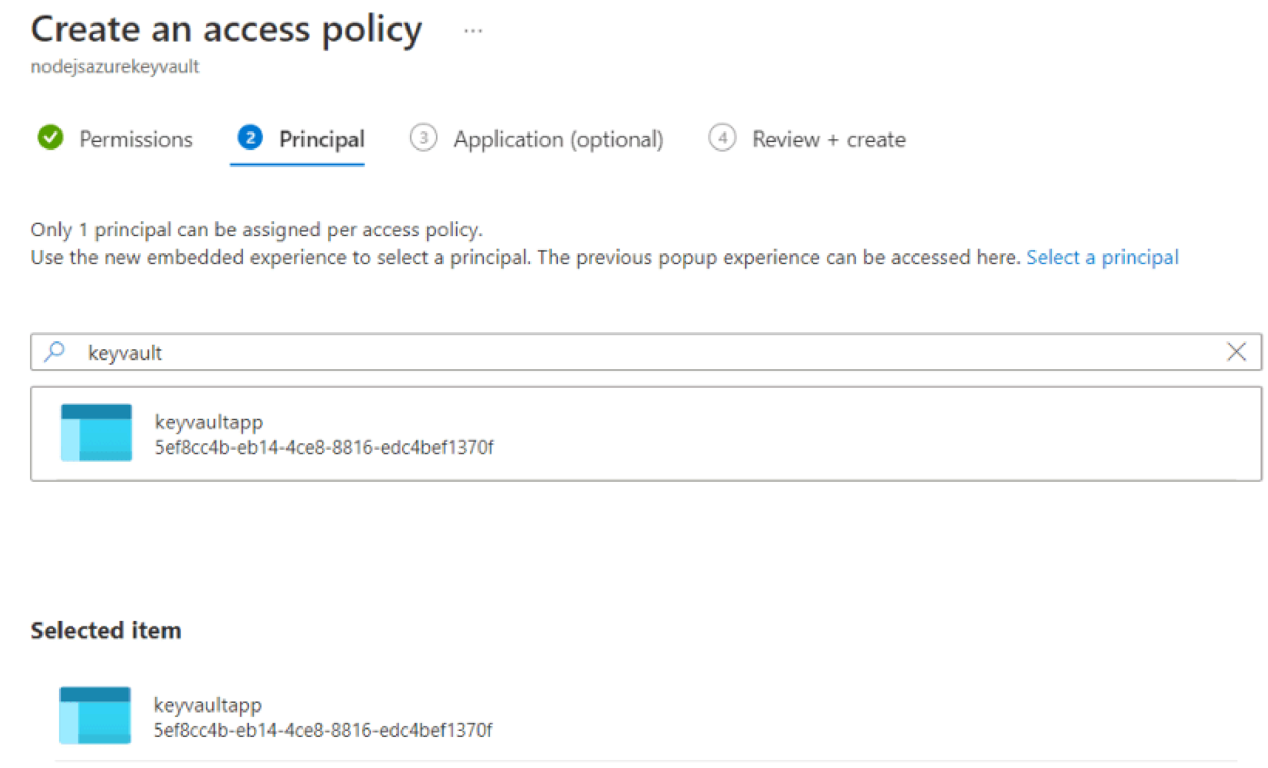
Certificate permissions

Certificate Management Operations

□ Select all

□ Get
□ List
□ Update
□ Create
□ Import
□ Delete
□ Recover
□ Backup
□ Restore
□ Manage Contacts
□ Manage Certificate Authorities
□ Get Certificate Authorities

1. Next
2. Select keyvaultapp



Create an access policy ...

nodejsazurekeyvault

Only 1 principal can be assigned per access policy.
Use the new embedded experience to select a principal. The previous popup experience can be accessed here. Select a principal

keyvault

keyvaultapp
5ef8cc4b-eb14-4ce8-8816-edc4bef1370f

Selected item

keyvaultapp
5ef8cc4b-eb14-4ce8-8816-edc4bef1370f

1. Next
2. Create

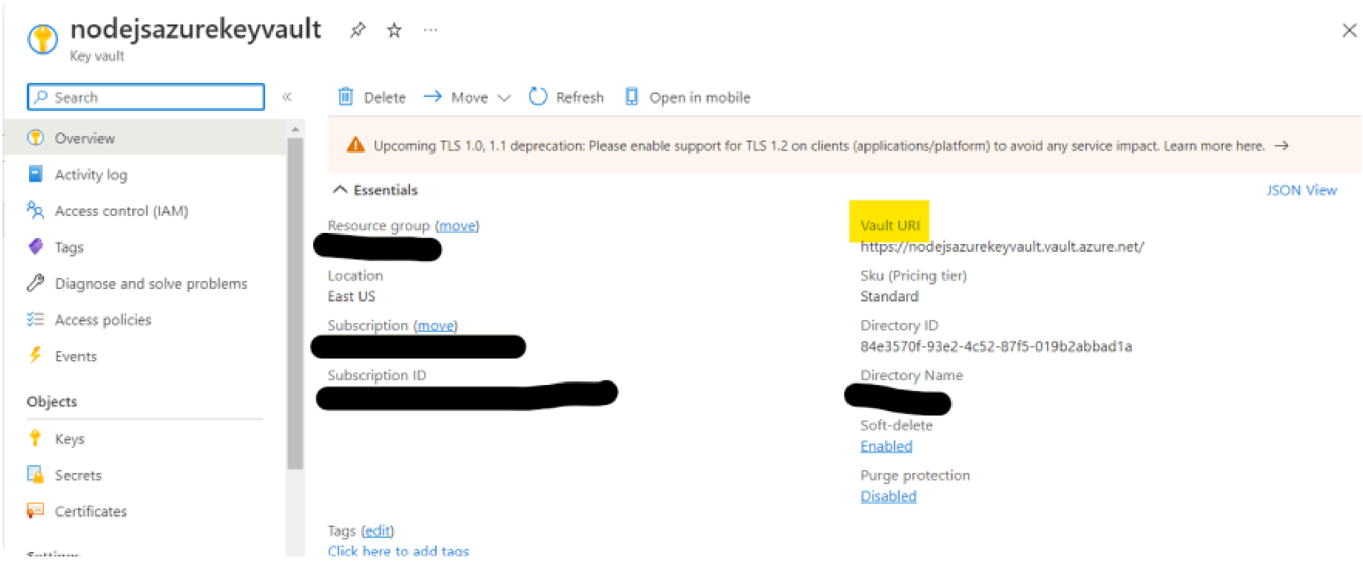
7. Reveal secrets in Nodejs application

1. Go to index.js
2. Open vs code terminal and install the following

```
npm install @azure/identity
npm install @azure/keyvault-secrets
npm install dotenv
```

1. Create a `.ENV` file and add the following code

```
KEYVAULT_URI=<"key vault URL">
AZURE_TENANT_ID=<"registered app in azure active directory">
AZURE_CLIENT_ID=<"registered app in azure active directory">
AZURE_CLIENT_SECRET=<"previously copied value">
```



Upcoming TLS 1.0, 1.1 deprecation: Please enable support for TLS 1.2 on clients (applications/platform) to avoid any service impact. Learn more here. →

Resource group (move)
[redacted]

Location
East US

Subscription (move)
[redacted]

Subscription ID
[redacted]

Vault URI
https://nodejsazurekeyvault.vault.azure.net/

Sku (Pricing tier)
Standard

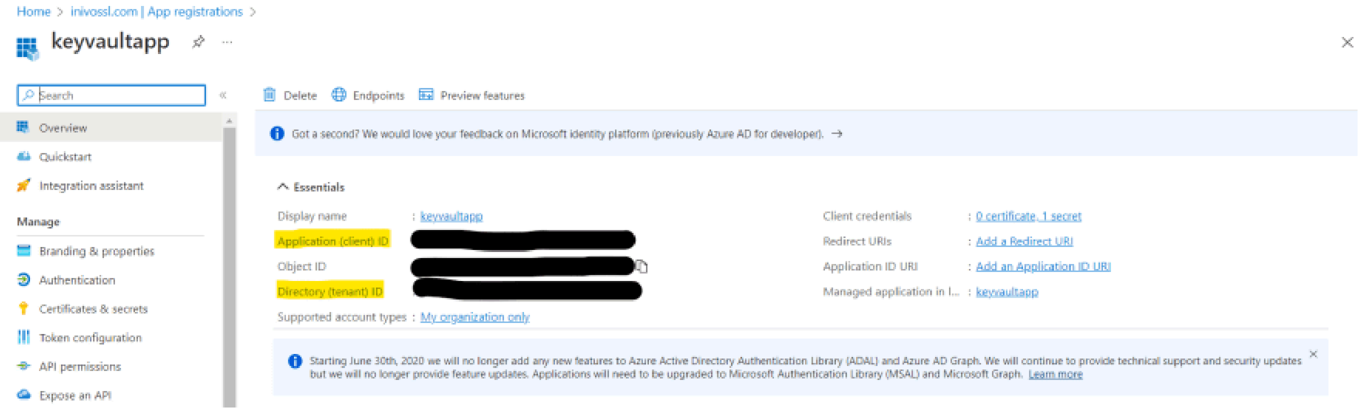
Directory ID
84e3570f-93e2-4c52-87f5-019b2abbad1a

Directory Name
[redacted]

Soft-delete
Enabled

Purge protection
Disabled

Tags (edit)
Click here to add tags

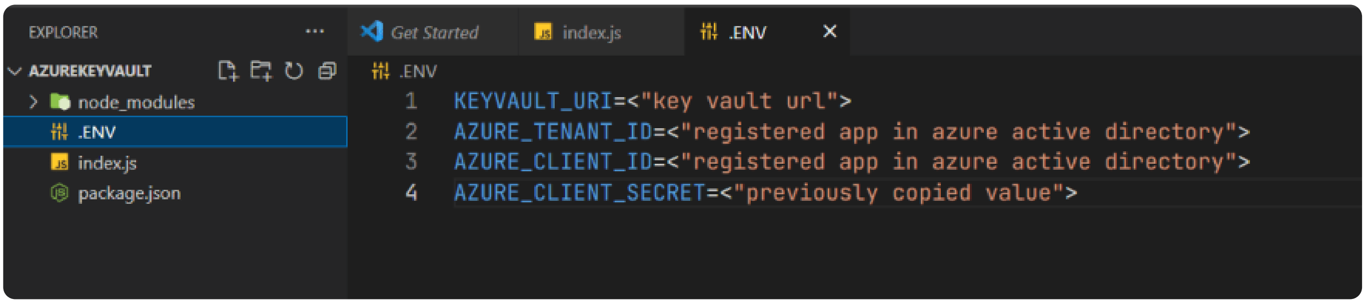


Got a second? We would love your feedback on Microsoft identity platform (previously Azure AD for developer). →

Display name : keyvaultapp
Application (client) ID [redacted]
Object ID [redacted]
Directory (tenant) ID [redacted]
Supported account types : My organization only

Client credentials : 0 certificate, 1 secret
Redirect URIs : Add a Redirect URI
Application ID URI : Add an Application ID URI
Managed application in l... : keyvaultapp

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



```
KEYVAULT_URI=<"key vault url">
AZURE_TENANT_ID=<"registered app in azure active directory">
AZURE_CLIENT_ID=<"registered app in azure active directory">
AZURE_CLIENT_SECRET=<"previously copied value">
```

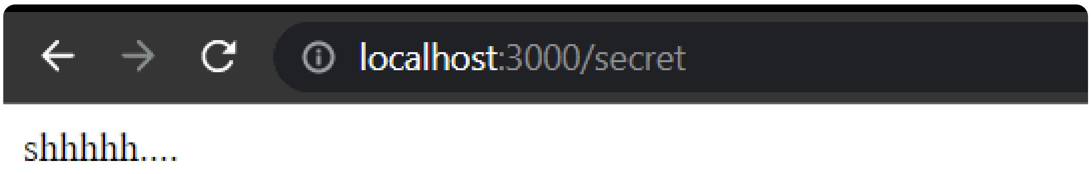
1. Add this code to index.js

```
require("dotenv").config();
const { DefaultAzureCredential } = require("@azure/identity");
const { SecretClient } = require("@azure/keyvault-secrets");
const credential = new DefaultAzureCredential();
const client = new SecretClient(process.env.KEYVAULT_URI, credential);
```

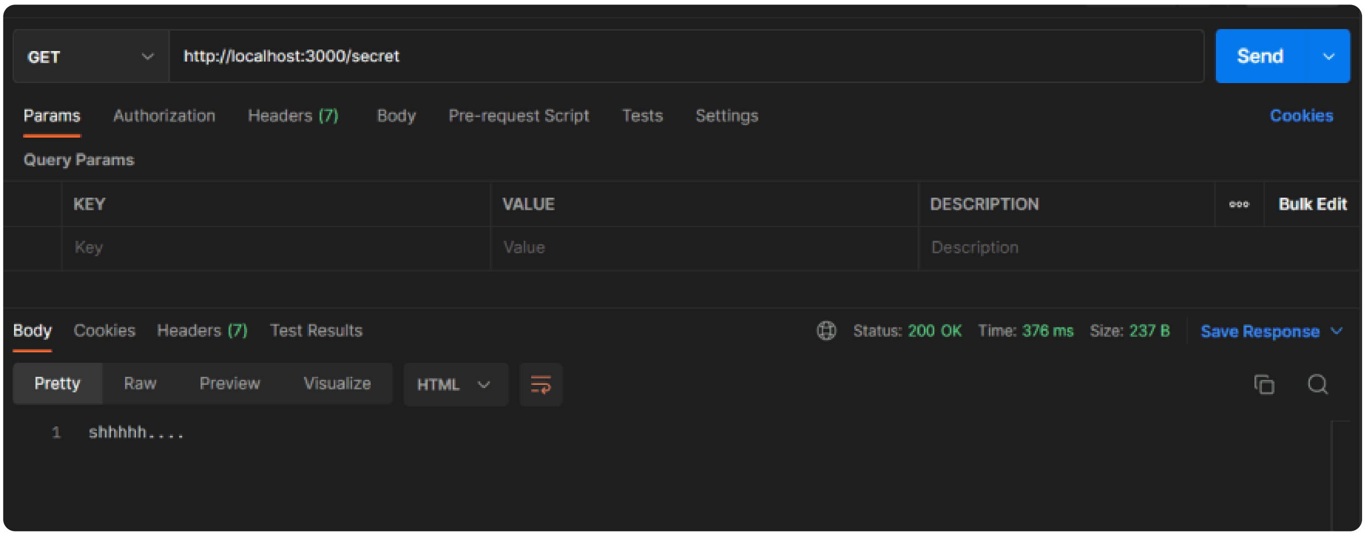
1. Create a separate route and add this code (single line secret)

```
app.get("/secret", (req, res) => {
  client
    .getSecret("testsecret")
    .then((data) => {
      res.send(data.value);
    })
    .catch((error) => {
      console.log(error);
      res.send(error);
    });
});
```

1. In your terminal (which should be in the project directory), type `nodemon index.js` and hit the Enter button.
2. Open a new tab in postman or any web browser and the address bar, type http://localhost:3000/secret, and hit the Enter button



```
shhhhh....
```



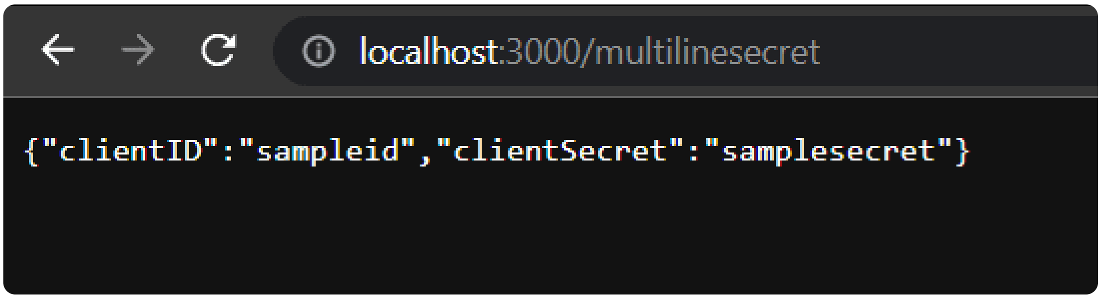
```
shhhhh....
```

1. Create a separate route and add this code (multi-line secret)

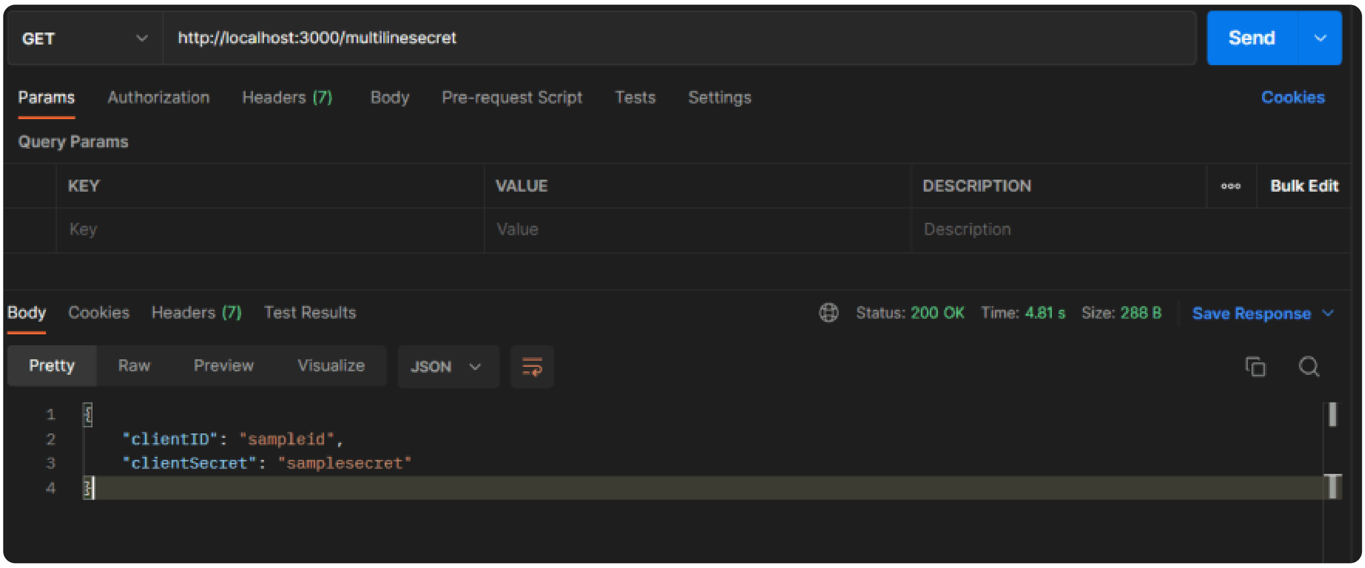
```
app.get("/multilinesecret", (req, res) => {
  client
    .getSecret("MultilineSecret")
    .then((data) => {
      const parsedSecret = JSON.parse(data.value);
      res.json(parsedSecret);
    })
    .catch((error) => {
      console.log(error);
      res.send(error);
    });
});
```

1. In your terminal (which should be in the project directory), type `nodemon index.js` and hit the Enter button.

2. Open a new tab in postman or any web browser and the address bar, type http://localhost:3000/multilinesecret, and hit the Enter button



```
{"clientID":"sampleid","clientSecret":"samplesecret"}
```



```
{
    "clientID": "sampleid",
    "clientSecret": "samplesecret"
}
```

Complete Code

```
const express = require("express");
const app = express();
const port = 3000;

require("dotenv").config();
const { DefaultAzureCredential } = require("@azure/identity");
const { SecretClient } = require("@azure/keyvault-secrets");
const credential = new DefaultAzureCredential();
const client = new SecretClient(process.env.KEYVAULT_URI, credential);

app.get("/", (req, res) => {
  res.send("Hello World!");
});

app.get("/secret", (req, res) => {
  client
    .getSecret("testsecret")
    .then((data) => {
      res.send(data.value);
    })
    .catch((error) => {
      console.log(error);
      res.send(error);
    });
});

app.get("/multilinesecret", (req, res) => {
  client
    .getSecret("MultilineSecret")
    .then((data) => {
      const parsedSecret = JSON.parse(data.value);
      res.json(parsedSecret);
    })
    .catch((error) => {
      console.log(error);
      res.send(error);
    });
});

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`);
});
```

Source Code

GitHub

Thank you.

👋 Before you go

Do your career a favor. **Join DEV.** (The website you're on right now)

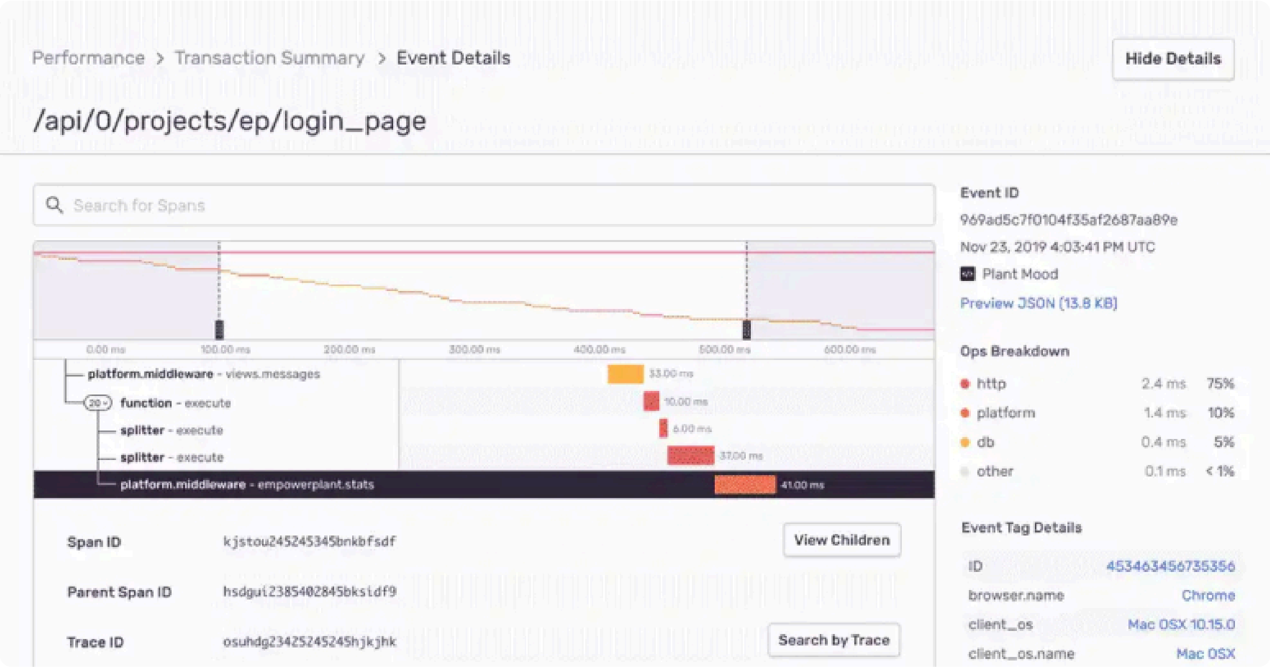
It takes ***one minute*** and is worth it for your career.

Get started

Top comments (0)

Code of Conduct • Report abuse

Sentry PROMOTED



/api/0/projects/ep/login_page

Span ID	kjstou245245345bnkbfsdf
Parent Span ID	hsdgui2385402845bksidf9
Trace ID	osuhdg23425245245hjkjhk

Event ID

969ad5c7f0104f35af2687aa89e

Nov 23, 2019 4:03:41 PM UTC

Plant Mood

Ops Breakdown

● http	2.4 ms	75%
● platform	1.4 ms	10%
● db	0.4 ms	5%
● other	0.1 ms	< 1%

Event Tag Details

ID	453463456735356
browser.name	Chrome
client_os	Mac OSX 10.15.0
client_os.name	Mac OSX

Next.js Performance Monitoring 🚫

Within minutes after installing Sentry, software teams are able to trace Next.js performance issues back to a poor performing API call as well as surface all related code errors. Engineering Managers and Developers now have a single tool to optimize the performance of their code and deliver fast customer experiences.

Try Sentry

Dileepa Mabulage

JOINED
Dec 28, 2022

More from Dileepa Mabulage

Uploading Images to a Sails.js Server: A Quick Guide
#webdev #javascript #sails #image

Creating AWS CloudWatch Logs using .NET Console Application: A Step-by-Step Guide
#javascript #cloudwatch #aws #dotnet

DEV Community

Check out this survey and help us moderate our community by becoming a tag moderator here at DEV.