

# Using Third Party Identity Providers

---



**Yaser Adel Mehraban (Yas)**

AZURE TECHNICAL TRAINER @MICROSOFT

@yashints yashints.dev



# Overview



Integrating with Auth0

Integrating with Okta

Integrating with Microsoft Azure Active Directory (AAD)



Auth0

---



**Auth0**



# Creating an Application


Create application

Name \*

My App

You can change the application name later in the application settings.


Choose an application type



**Native**

Mobile, desktop, CLI and smart device apps running natively.


e.g.: iOS, Electron, Apple TV apps



**Single Page Web Applications**

A JavaScript front-end app that uses an API.


e.g.: Angular, React, Vue



**Regular Web Applications**

Traditional web app using redirects.

e.g.: Node.js, Express, ASP.NET, Java, PHP



**Machine to Machine Applications**

CLIs, daemons or services running on your backend.

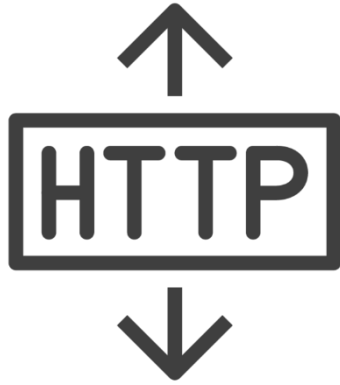
e.g.: Shell script

CREATE

CANCEL



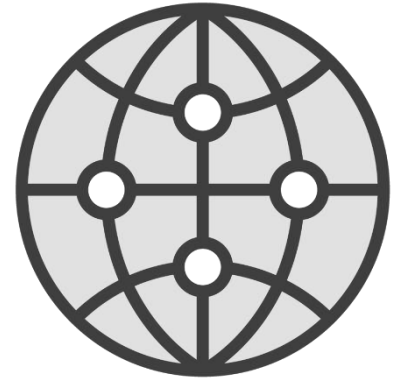
# Required Information



Callback URL




Logout URL




Web Origins



# Install the SPA Package

@auth0/auth0-spa-js 

1.13.6 • Public • Published a day ago

 Readme

 Explore 

 7 Dependencies

 136 Dependents

 51 Versions

## @auth0/auth0-spa-js

Auth0 SDK for Single Page Applications using **Authorization Code Grant Flow with PKCE**.

 PASSED

release **v1.13.6**

coverage **98%**

downloads **225k/week**

license **mit**

## Table of Contents

- [Documentation](#)
- [Installation](#)
- [Getting Started](#)
- [Contributing](#)
- [Support + Feedback](#)
- [Frequently Asked Questions](#)
- [Vulnerability Reporting](#)
- [What is Auth0](#)
- [License](#)

### Install

```
> npm i @auth0/auth0-spa-js
```

### ± Weekly Downloads

225,415

Version

**1.13.6**

License

**MIT**

Unpacked Size

**2.36 MB**

Total Files

**53**


Issues

**2**

Pull Requests

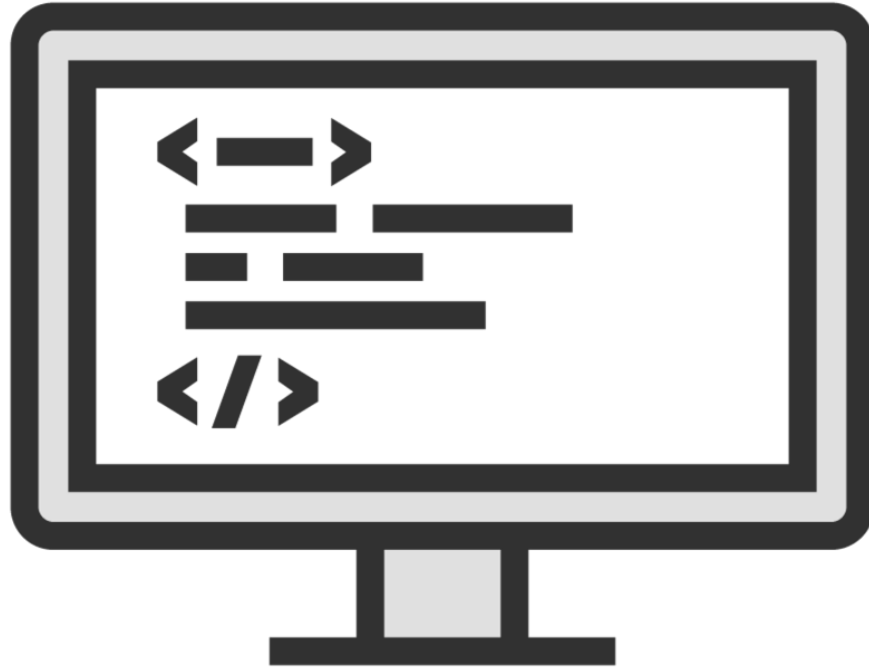
**0**

Homepage

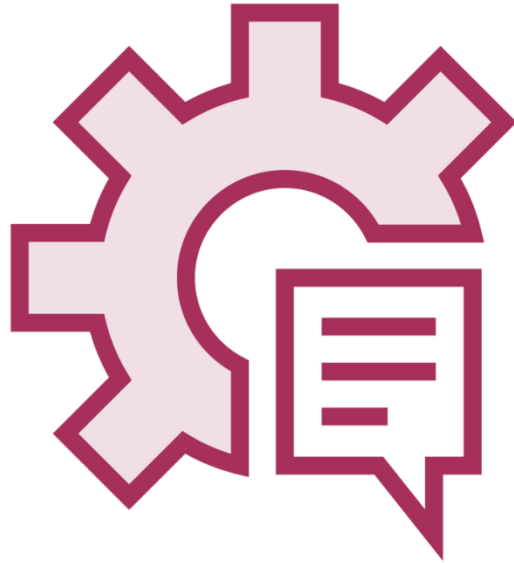
 [github.com/auth0/auth0-spa-js#readme](https://github.com/auth0/auth0-spa-js#readme)



# Create an Authenticating Wrapper



# Auth0 Configuration





# Wiring Login/Logout Buttons



# Protect the Web API Endpoints



# Demo



Creating an Auth0 Application

Adding the SPA Package

Creating the Auth Wrapper

Wiring up Login/Logout Buttons

Securing the Backend APIs



Okta

---



# Create an Okta Application

[Get Started](#)[Dashboard](#)[Users](#)[Applications](#)[API](#)[Workflow](#)[Customization](#)[Settings](#)[Upgrade](#)

## Create New Application

1

Platform

2

Settings

An application in Okta represents an integration with the software you're building.  
Choose your platform, and we'll recommend settings on the next step.



Native

iOS, Android



Single-Page App

Angular, React, etc.



Web

.NET, Java, etc.



Service


Machine-to-Machine

[Previous](#)[Cancel](#)[Next](#)

# Install and Configure the SDK

@okta/okta-vue TS

3.1.0 • Public • Published 4 days ago

 Readme

 Explore BETA

 1 Dependency

 3 Dependents

 20 Versions

## Okta Vue SDK

npm v3.1.0 build passing

Okta Vue SDK builds on top of the **Okta Auth SDK**. This SDK integrates with the **vue-router** and extends the **Vue prototype** with an **Okta Auth SDK** instance to help you quickly add authentication and authorization to your Vue single-page web application.

With the **Okta Auth SDK**, you can:

- Login and logout from Okta using the **OAuth 2.0 API**
- Retrieve user information
- Determine authentication status
- Validate the current user's session

All of these features are supported by this SDK. Additionally, using this SDK, you can:

- Add "protected" routes, which will require authentication before render
- Provide an instance of the **Okta Auth SDK** to your components on the **Vue prototype**
- Inject reactive **authState** property to your Vue components

This SDK does not provide any UI components.

### Install

```
> npm i @okta/okta-vue
```

Weekly Downloads

19,550

Version

3.1.0

License

Apache-2.0

Unpacked Size

92.4 kB

Total Files

28

Issues

8

Pull Requests

2

Homepage

 [github.com/okta/okta-vue#readme](https://github.com/okta/okta-vue#readme)

Repository

 [github.com/okta/okta-vue](https://github.com/okta/okta-vue)



## Handle the Callback from Okta

```
const CALLBACK_PATH = '/login/callback'

const router = new Router({
  mode: 'history',
  routes: [
    { path: CALLBACK_PATH, component: Auth.handleCallback() },
  ]
})
```



# Complete the Puzzle



Require Authentication



Get Info About the User





# Protect the Web API Endpoints



Demo



Creating an Okta Application

Adding the SPA SDK

Configuring the SDK

Handle the Callback from Okta

Get User Info and Use the Access Token

Setup Route Guards

Secure Our Backend APIs



# Microsoft Azure Active Directory

---



# Creating the App Registration

Home > Default Directory > test

test | Authentication

Search (Ctrl+/)

Save Discard Got feedback?

Overview

Quickstart

Integration assistant

Manage

Branding

Authentication

Certificates & secrets

Token configuration

API permissions

Expose an API

App roles | Preview

Owners

Roles and administrators | Preview

Manifest

Support + Troubleshooting

Troubleshooting

New support request

## Platform configurations

Depending on the platform or device this application is targeting, additional configuration may be required such as redirect URIs, specific authentication settings, or fields specific to the platform.

+ Add a platform

## Supported account types

Who can use this application or access this API?

- ☒ Accounts in this organizational directory only (Default Directory only - Single tenant)
- ☐ Accounts in any organizational directory (Any Azure AD directory - Multitenant)

[Help me decide...](#)

Due to temporary differences in supported functionality, we don't recommend enabling personal Microsoft accounts for an existing registration. If you need to enable personal accounts, you can do so using the manifest editor. [Learn more about these restrictions.](#)

## Advanced settings

### Allow public client flows

Enable the following mobile and desktop flows:

Yes No

- App collects plaintext password (Resource Owner Password Credential Flow) [Learn more](#)
- No keyboard (Device Code Flow) [Learn more](#)
- SSO for domain-joined Windows (Windows Integrated Auth Flow) [Learn more](#)

## Configure platforms

### Web applications



Web

Build, host, and deploy a web server application. .NET, Java, Python



Single-page application

Configure browser client applications and progressive web applications. Javascript.

### Mobile and desktop applications



iOS / macOS

Objective-C, Swift, Xamarin



Android

Java, Kotlin, Xamarin



Mobile and desktop applications

Windows, UWP, Console, IoT & Limited-entry Devices, Classic iOS + Android



# Install the MSAL.js 2.0

msal TS

1.4.4 • Public • Published 2 months ago

Readme

Explore BETA

1 Dependency

130 Dependents

55 Versions

## Microsoft Authentication Library for JavaScript (MSAL.js)

Getting Started

AAD Docs

Library Reference

Support

Samples

MSAL for JavaScript enables client-side JavaScript web applications, running in a web browser, to authenticate users using **Azure AD** work and school accounts (AAD), Microsoft personal accounts (MSA) and social identity providers like Facebook, Google, LinkedIn, Microsoft accounts, etc. through **Azure AD B2C** service. It also enables your app to get tokens to access **Microsoft Cloud** services such as **Microsoft Graph**.

npm v1.4.4 downloads 645k/month coverage 82%

### Installation

Via NPM:

```
npm install msal
```

Via CDN:

```
<script type="text/javascript" src="https://alcdn.msauth.net/lib/1.4.4/js/msal.min.js"></script>
```

Install

```
> npm i msal
```

Weekly Downloads

150,698

Version

1.4.4

License

MIT

Unpacked Size

5.08 MB

Total Files

445

Issues

64

Pull Requests

20

Homepage

[github.com/AzureAD/microsoft-authenti...](https://github.com/AzureAD/microsoft-authentication-library-for-javascript)

Repository

[github.com/AzureAD/microsoft-authenti...](https://github.com/AzureAD/microsoft-authentication-library-for-javascript)

Last publish

2 months ago



# Configure the MSAL Instance

```
const app = createApp(App);  
app.config.globalProperties.$msalInstance = {};  
app.config.globalProperties.$emitter = new Emitter();
```



# Wire up Login/Logout Buttons



# Get the Access Token

```
async getAccessToken(){
  let request = {
    scopes: ['https://storage.azure.com/user_impersonation'],
  };
  const msalInstance = new PublicClientApplication(
    this.$store.state.msalConfig,
  );
  try {
    let tokenResponse = await msalInstance.acquireTokenSilent(request);
    this.$store.commit('setAccessToken', tokenResponse.accessToken);
  } catch (error) {
    console.error( 'Silent token acquisition failed. Using interactive mode' );
    let tokenResponse = await msalInstance.acquireTokenPopup(request);
    console.log(`Access token acquired via interactive auth ${tokenResponse.accessToken}`);
    this.$store.commit('setAccessToken', tokenResponse.accessToken);
  }
}
```

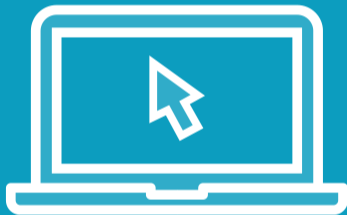




# Protect the Web API Endpoints



# Demo



Creating the App Registration

Adding MSAL 2.0 Package

Configuring the MSAL Instance

Wiring up Login/Logout Buttons

Route Guard

Secure the Backend APIs



# Summary



Integrated with Auth0

Integrated with Okta

Integrated with Microsoft AAD

Creating the Application

Installing the Client Library

Configuring the SDK



# Not Covered Items



**Refresh Tokens and Silent Refresh**



**User Consent**



**System to System Communication**



Up Next:

Testing Authenticated Vue Apps

---

