Okta integration doc recommendation list

Summary

After creating my SSO App integration with Okta, I compiled a list of improvements that can be made to the Okta integration documentation. This list was generated by my own personal experiences with the documentation as I integrated the Okta SSO product into my app.

The documentation links I used for the integration are as follows but not limited to:

1. https://www.okta.com/integrate/documentation/single-sign-on/
2. https://developer.okta.com/docs/guides/sign-into-web-app

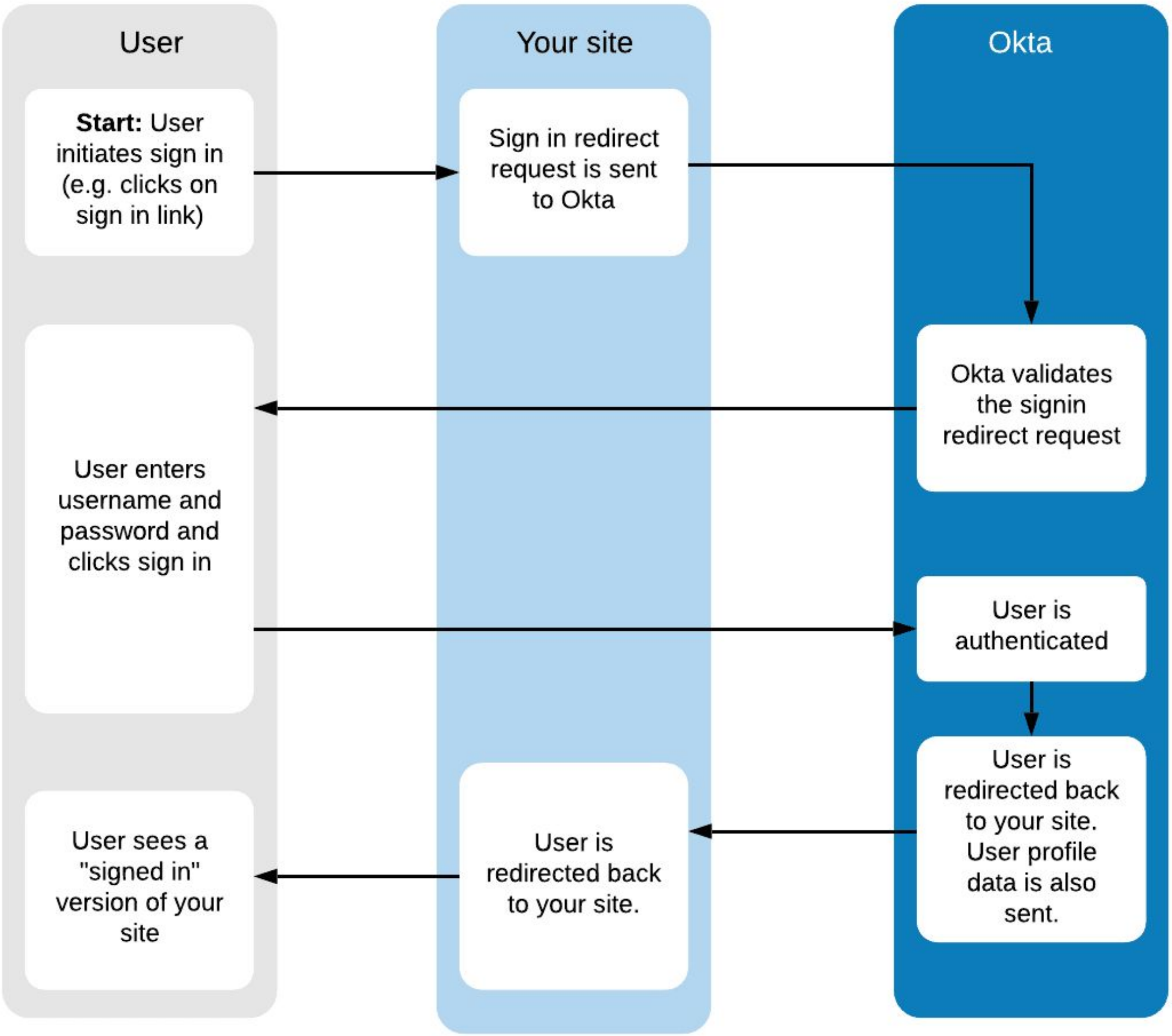
Note that I have marked each recommendation with an Importance (High, Medium, Low)

1. Add an “Overview” section (Medium)

In Sign users in your web application add a “Overview” section after “Before you begin” to give readers an idea what they will accomplish once they complete the integration guide. For example:

Overview

After you complete the steps in this guide you will have integrated a SSO in your website using Okta. The process starts with the user initiating a sign in from your website. Your website redirects the request to Okta where the user is presented the Okta sign in page. Once the user signs in and is successfully authenticated, Okta redirects the user to a URL that you provided. User profile data is also sent to the URL. See the diagram below for more details:

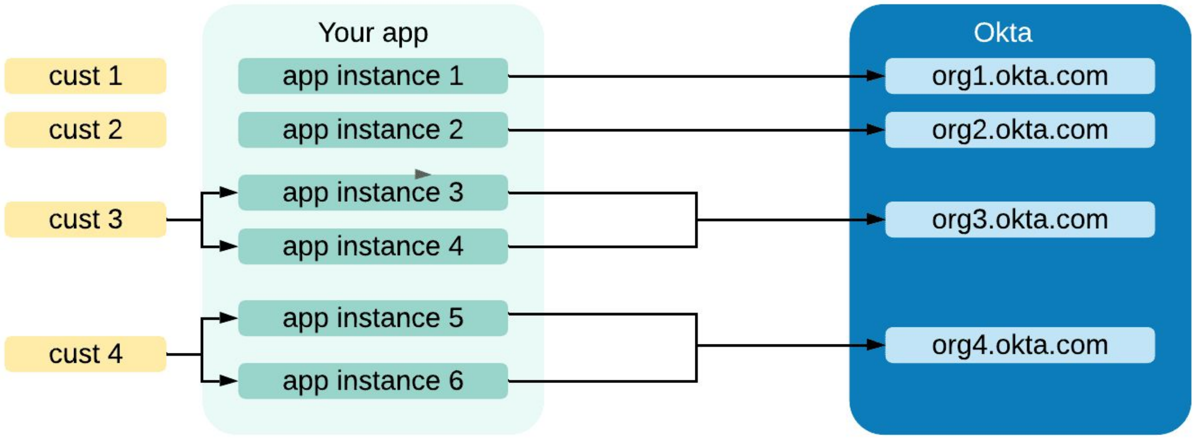


User
Your site
Okta
Start: User initiates sign in (e.g. clicks on sign in link)
Sign in redirect request is sent to Okta
Okta validates the signin redirect request
User enters username and password and clicks sign in
User is authenticated
User is redirected back to your site. User profile data is also sent.
User is redirected back to your site.
User sees a "signed in" version of your site

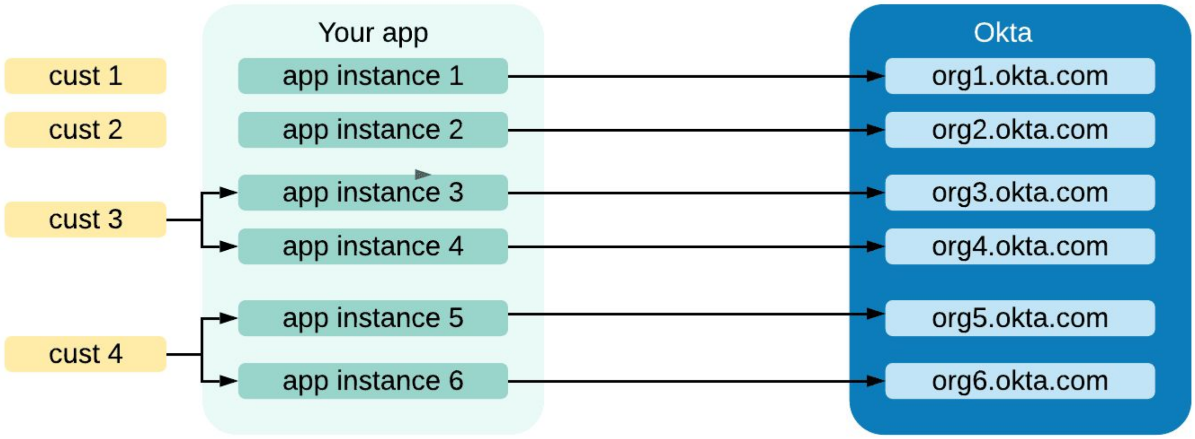
2. Improve Multi-tenant example (Medium)

In the Prepare your OIDC integration page, enhance the “multi-tenant” example by simplifying the number of apps, customers, and orgs and add a visual diagram. An sample of such content is shown below:

For example, consider a scenario where your app integration is added to 4 separate customer orgs. Two of those customers create a single instance of your app integration. However, the other two customers each create two separate instances of your app integration so they can use different configuration options. This creates a total of 6 sets of client credentials for your application to track. See the illustration below:



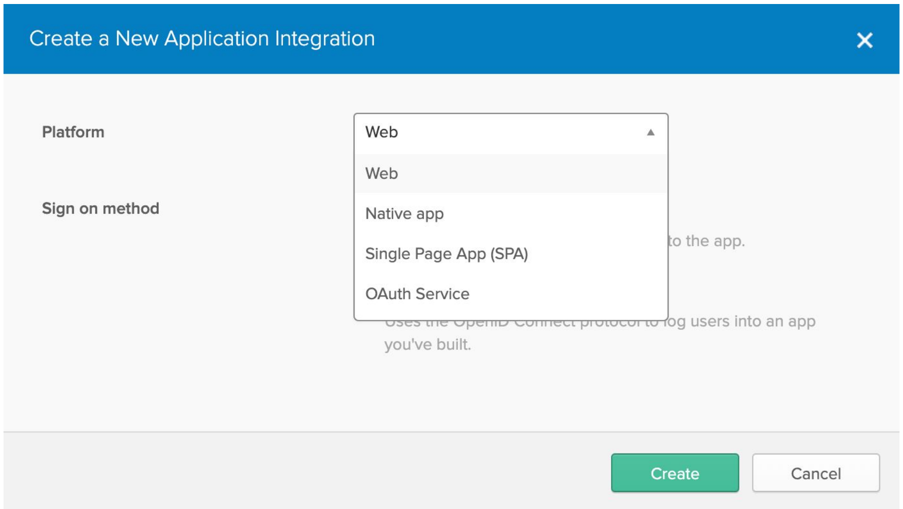
Or is this the correct diagram?



3. Add more information on Platform type (Low)

In the Create an OIDC integration section of the **Create an integration** page there should be more information about platform type. Though readers might know what Web and SPA platform types are specifically, more information on how each affects the Okta SSO integration is necessary. For example, I tried to integrate the Okta sign on widget to an Okta application that I had configured as “Web” but it seemed only to work for “SPA” apps.

For reference a list of Platforms are shown below:



Create a New Application Integration

Platform: Web

Sign on method

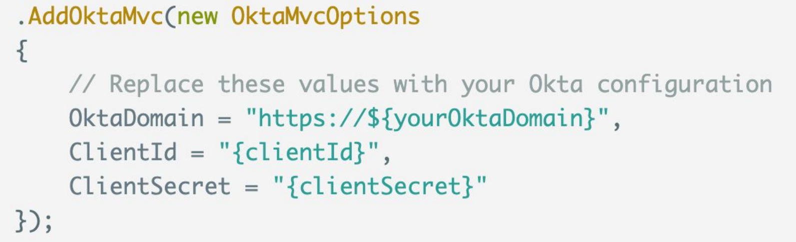
Web
Native app
Single Page App (SPA)
OAuth Service
to the app.
Uses the OpenID Connect protocol to log users into an app you've built.
Create
Cancel

4. Add a summary of all possible integrations (Medium)

New documentation that summarizes the types of integrations and nuances for each type of integration would add clarity to the available integration options. As a user of the documentation, I found it hard to identify all possible integration types. I saw guides for the Okta Signin widget and read about subtle integration differences between accessing apps directly or via the Okta dashboard. A summary will help a customer understand which integration best meets their needs.

6. Ensure consistency with placeholders (Low)

In the Configure your package section of the **Add and configure your packages** page, the domain URL placeholder contains a leading dollar sign symbol, whereas, other placeholders omit the dollar sign. In my integration I initially kept the dollar sign in the domain URL and had to track down the source of the errors. I did eventually remove the dollar sign but it caused a delay in my integration process.



```
.AddOktaMvc(new OktaMvcOptions
{
    // Replace these values with your Okta configuration
    OktaDomain = "https://${yourOktaDomain}",
    ClientId = "{clientId}",
    ClientSecret = "{clientSecret}"
});
```

7. Add more content to the callback route section (Medium)

1. Clarity on callback route and adding it to the Login Redirect URIs
Although the callback route is explained in both Understanding your callback route and Create an Okta application pages, I didn't first make the connection on how the callback route is defaulted to "authorization-code/callback" and how it needs to be added to the Login redirect URIs. Also the fact that this is an allowed list of Login redirects wasn't immediately apparent. Adding text similar to below may increase clarity:

*After the user sign in is successfully authenticated on the Okta Signin page, Okta will redirect back to the URL that you define in the Okta dashboard. If you don't define a redirect URL, the default (http://{referring domain/authorization-code/callback.) will be used. This URL needs to be in the allowed list of Redirect URLs for the application. Add the redirect URL to the **Login redirect URIs** field in the General tab for the application.*

NOTE: If you like to define a redirect URL go to the "Change the SignIn Redirect URL".

2. Clarity on the following sentence:
In the Understanding the callback route page the following sentence needs more explanation:

"The callback route is not seen by the user, and it's not the user's final destination. It's just one step in the authentication redirect flow."

A new user reading this may want to know what the other steps are. What are the potential user's final destinations? What are the other steps in the auth redirect flow once the user is redirected back to your page. In my app, I sent the user to the home page but I was left wondering if there was another step I had to perform.

8. Add prerequisites to MFA Setup (Low)

Following the steps to set up MFA for my app's users, I had to back out twice to add a group and define a factor. Having these two steps as prerequisites would have made the process smoother.

9. Add step to add person to the Okta application (Low)

When I completed my integration, I received an unhandled error on sign in which occurred because the user, though existed in the Okta org did not exist in the Okta application. A new step needs to be added to the Sign users into your web application page that mentions adding a person to the Okta application. For new developers not familiar with Okta business objects (e.g. domain, app, person, groups, etc.), this step may not be obvious.

10. Add content to assign a self service user to an app (Medium)

New content needs to be added to explain how to automatically give new self registered users access to the Okta applications. Such content is missing in the self-service registration instructions. After finding help on some dev forums, I was able to grant new users access by performing the following steps on the Okta Admin console:

1. Define a new group such as **NewSelfServedUsers** and assign this group to the Okta application.
2. In the Self-Service Registration tab, add the **NewSelfServedUsers** to the **Assign to Group** field.

After I performed these steps, new users were assigned the **NewSelfServedUsers** group and allowed access to my application

11. Add guidance on how to Authorize users once they are authenticated (Medium)

Once I successfully integrated SSO authentication into my app, I wanted to learn and understand how to authorize the users for different features in my app. For example, users of an “Admin” group can access the Administrative screen while others cannot. Specifically, I wanted to retrieve these groups in ASP.Net and perform authorization logic. Unfortunately, I could not find how to perform this authorization given my integration type.

12. Remove renderEL in the docs (High)

I attempted an integration with the Okta Sign in widget and used the Okta Sign-In Widget Guide. In the guide it recommends using OktaSignIn.renderEL method. However, the Okta Sign-in Widget Migration guide discourages its use and recommends other options.