PingFederate™ Integration Guide

Version 9.1.0

(October 2024 Prerelease)

Nok Nok™ PingFederate™ Integration Guide, October 2024

Table of Contents

Introduction

This document guides you through the installation and configuration of the Nok Nok PingFederate Adapter. Using this Adapter you can implement passwordless authentication on your PingFederate Server, providing a seamless and secure authentication experience for your users.

Your app authenticates a user by sending a JWT to the PingFederate Server. The PingFederate Server then sends the JWT to the Nok Nok PingFederate Adapter for validation. The Server can send this JWT to the Adapter in any one of the following places in its request:

- header
- cookie
- form POST[1]

Upon successful validation of the JWT, the Nok Nok Adapter extracts the sub claim value and stores it in the `username` attribute of the JWT for the PingFederate Server. See the Sequence Diagrams for more detail.

The package includes:

- The Nok Nok Adapter and its required JAR libraries
- The Nok Nok Web App SDK and the `nnlsignin` web application

Related Documents

This Guide references several resources that provide additional information about the Nok Nok App SDK and the Nok Nok Authentication Server. You should locate these documents and reference them as you read this Guide.

- **Nok Nok App SDK for JavaScript Developer Guide**: Details how to include strong authentication in a web app using the Nok Nok App SDK.
- **Nok Nok App UI Customization Guide:** Explains how to customize the UI that the Nok Nok App SDK displays. This document contains instructions for editing a cross-platform JSON configuration file to localize and style the UI.
- **Nok Nok S3 Authentication Suite Configuration Guide**: This guide provides step-by-step instructions to perform common post-installation and operational tasks on the Nok Nok Authentication Server. It explains concepts such as RP IDs, facet IDs, session data, and Adaptive Authentication. It is organized around the set up needed for UAF and FIDO2 as well as major features like Adaptive Authentication, FIDO out-of-band (OOB) Authentication and External Authentication.
- **Client API Documentation**: The standard automatically-generated API documentation. Located in the `docs/jssdk` directory within the Nok Nok JavaScript App SDK deployment package: `nn_web_app_sdk_<version>.<build_number>.zip`.

[1] If your end user sign-in page and the Nok Nok Adapter have entirely different domains, then you *must* use form POST for the JWT.

Installation

Before beginning these steps, make sure that the Nok Nok Server is deployed and operational. Then follow these instructions to install the Nok Nok Adapter on your PingFederate deployment:

1. Copy all JAR libraries from the installation package to the PingFederate server's deploy directory:
   `<pf-install>/pingfederate/server/default/deploy`
2. Replace any older versions of the third-party JARs included in the package to prevent duplication.
3. Prepare a `jwt_config.json` file for the Nok Nok Server tenant. To ensure compatibility with the Nok Nok API Server, make sure the keys and configuration parameters for JWT in the `jwt_config.json` file match those of the API server. If using a symmetric key on the server, you can export the JWT configuration from the API server session plugin. Refer to the *Nok Nok Authentication Suite Configuration Guide,* section *Exporting and Importing Configuration and Objects* for instructions on how to export the API Server configuration.
4. Copy the prepared `jwt_config.json` file to `<pf-install>/pingfederate/server/default/conf/configurations/<TENANT IDENTIFIER>/SessionPlugin/jwt_config.json`, creating the required intermediate directories as needed. See the Nok Nok Adapter's configuration parameters below.
5. Restart the PingFederate Server. The Nok Nok Adapter appears on the PingFederate Admin Console.
6. Find the `nnlsignin` application inside the Nok Nok Web App SDK package. Deploy the `nnlsignin` web application on a web server. `nnlsignin` can be hosted on an origin that is different from the origin that hosts the Nok Nok Adapter. You will configure the sign-in page after you configure the Adapter.

Multi-Tenant Deployments

If your Nok Nok deployment supports multiple tenants, follow these additional installation instructions:

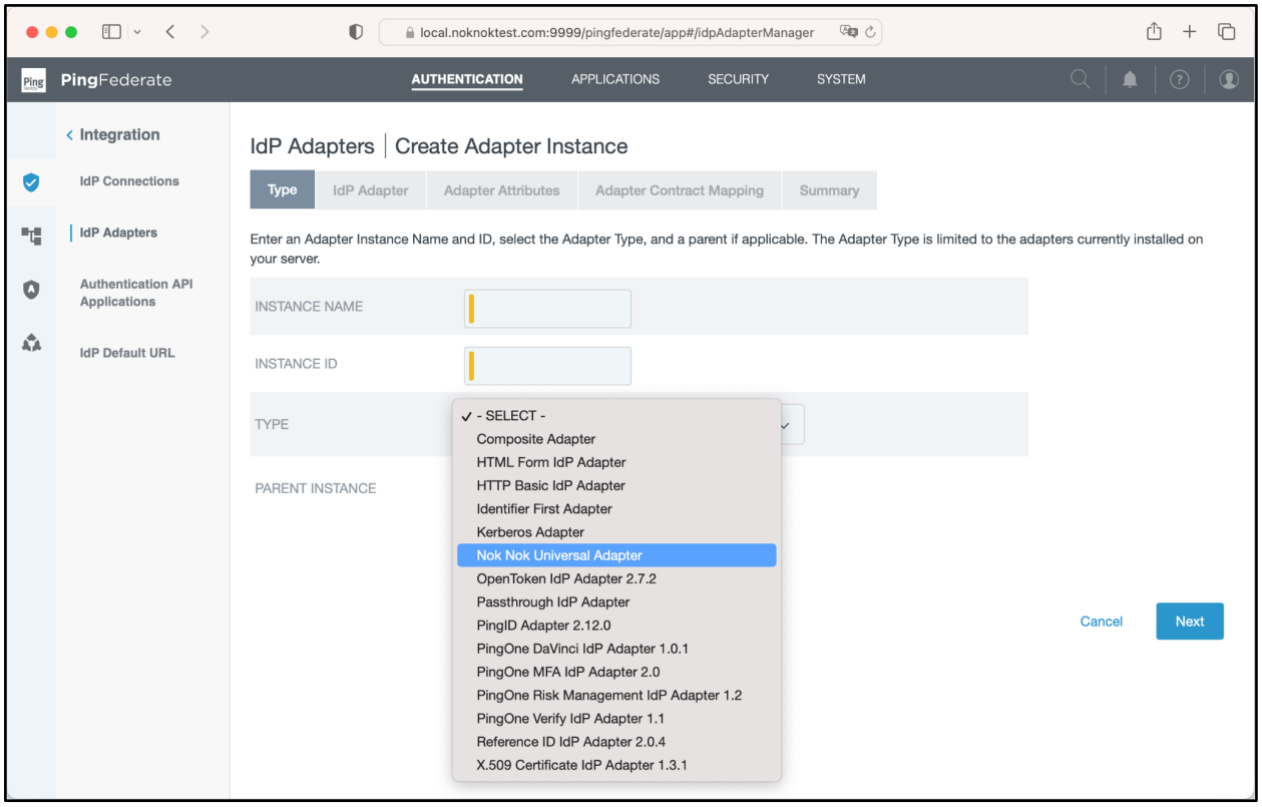
1. Use the PingFederate Admin Console to create and configure a separate realm in your PingFederate server for each Nok Nok Server tenant. The Nok Nok Adapter uses the TENANT IDENTIFIER that you configure for a realm in your PingFederate server to find the directory containing the configurations for that realm. See Step 4. above for details of this directory structure.
2. Also while configuring each realm, make sure that the SIGN IN ENDPOINT includes the tenant_id parameter. For example, the following value for SIGN IN ENDPOINT is needed for the `nnlsignin` app:
   `<SIGN IN ENDPOINT>?tenant_id=<TENANT IDENTIFIER>`
3. Repeat Step 3. and Step 4. in the Installation instructions above for each tenant.

Configuration

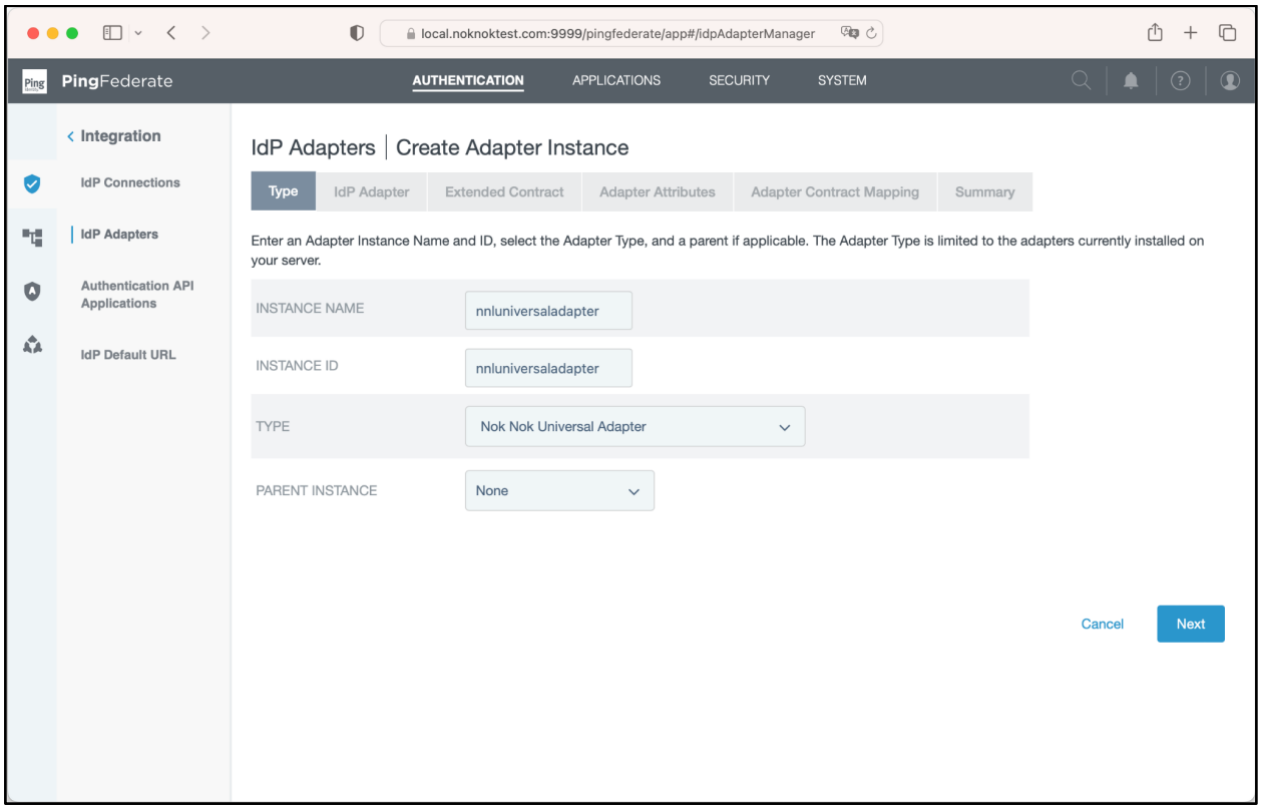
To configure the Nok Nok Adapter for your PingFederate deployment, follow the instructions below using the PingFederate Admin Console:

Step 1) Create IdP Adapter Instance

To create an instance of the Nok Nok Adapter, navigate to **Authentication** > **Integration** > **IdP Adapters**. Click **Create New Instance**. If the Nok Nok Adapter has been successfully installed, **Nok Nok Universal Adapter** is displayed in the list of available adapters.

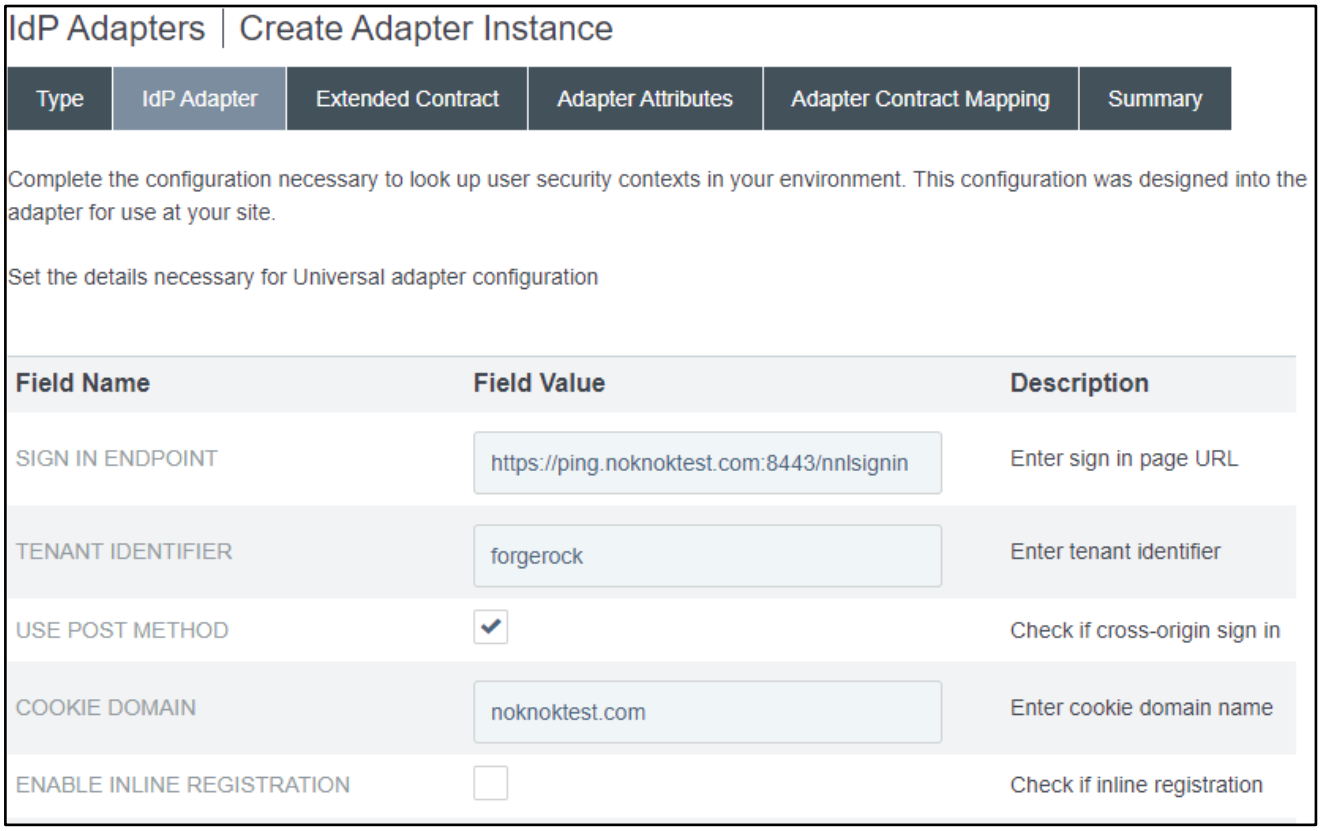


Fill in the details and click **Next**:

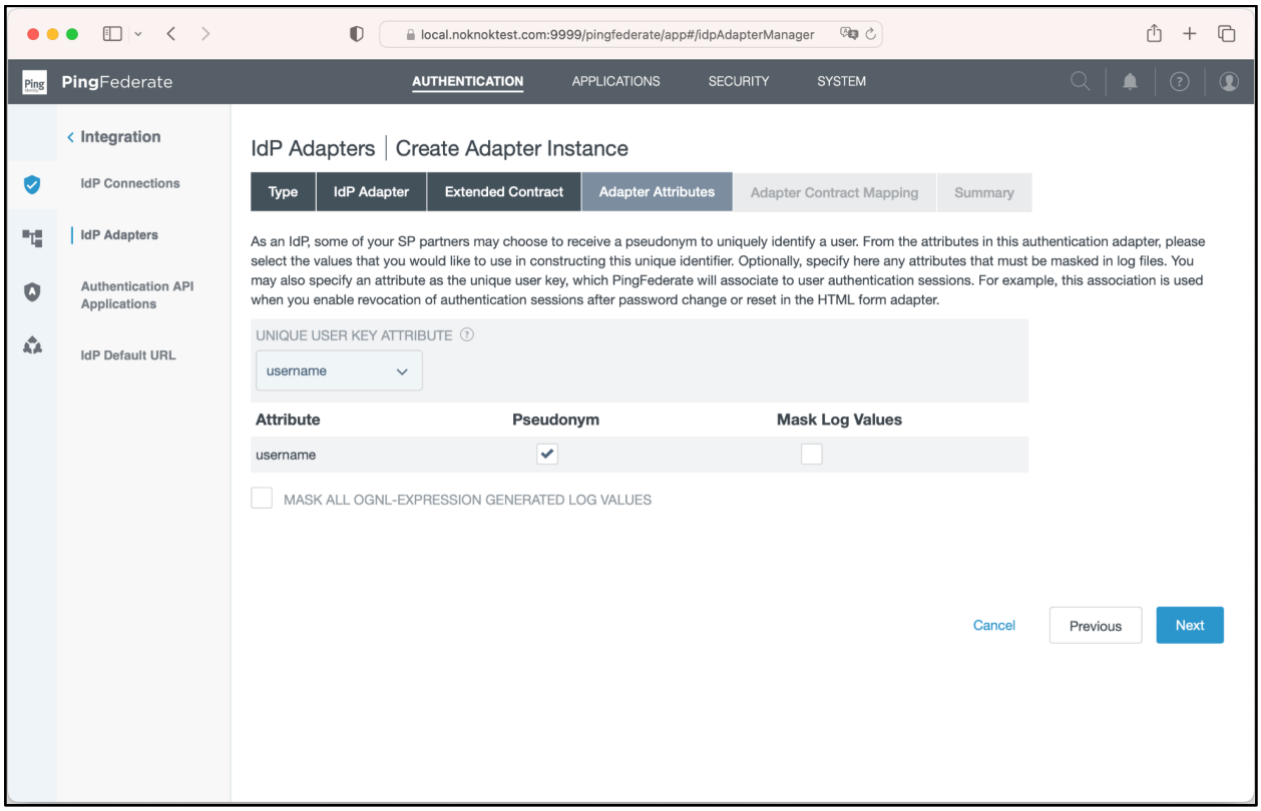


In the PingFederate Admin Console, the Nok Nok Adapter provides the following configuration parameters:

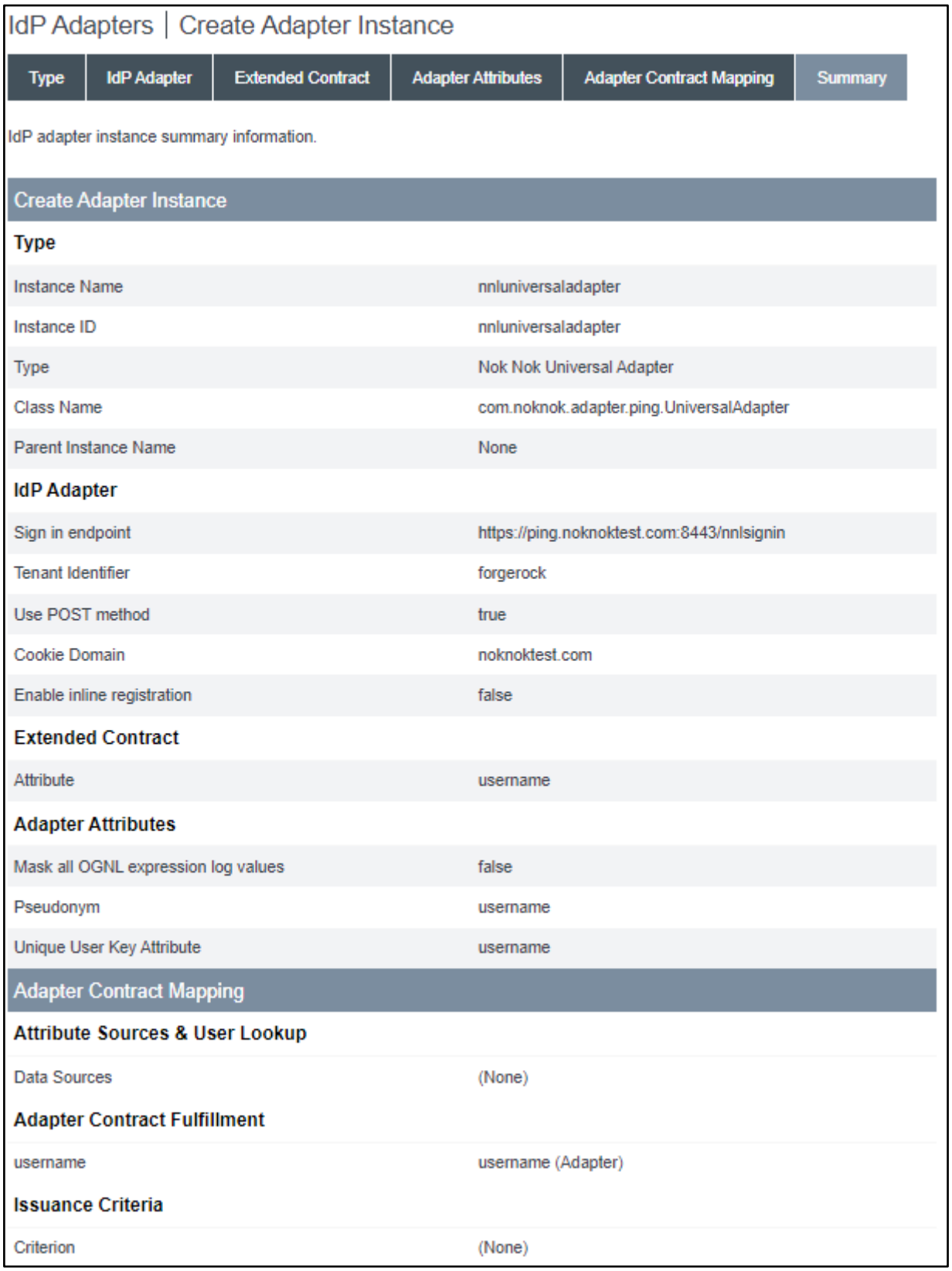
- **SIGN IN ENDPOINT** - This is the URL of the sign-in page where the Adapter redirects if the JWT is missing from the request, or if the JWT is present in the request but invalid.
- **TENANT IDENTIFIER** - Specifies the tenant ID used for creating tenant-specific intermediate directories.
- **USE POST METHOD** - Check this box if one or both of the following applies:
    - your sign-in page delivers the JWT to the Adapter as a request parameter in a form POST.
    - your sign-in page and the PingFederate Server have different origins.
- **COOKIE DOMAIN** - If the PingFederate Server and the nnlsignin app have the same parent domain, specify the domain name here. This setting is ignored if "USE POST METHOD" is enabled.
- **ENABLE INLINE REGISTRATION -** When Inline Registration is enabled, end users are prompted to register if they have no credentials already registered. The following are the prerequisites to enabling Inline Registration:
    - The Nok Nok Adapter is used for second-factor authentication. The adapter should be set as part of an adapter chaining flow in the policy to enable multi-step authentication.
    - Your PingFederate policy is configured to permit Inline Registration.



Navigate to the **Adapter Attributes** tab and select username.



To finish setting up the Nok Nok Adapter, click **Next** to access the summary screen. Take a moment to review the summary and confirm that all information is accurate. After confirming the information, click **Save** to create the adapter instance.



IdP Adapters | Create Adapter Instance

Type | IdP Adapter | Extended Contract | Adapter Attributes | Adapter Contract Mapping | Summary

IdP adapter instance summary information.

Create Adapter Instance

Type

Instance Name	nnluniversaladapter
Instance ID	nnluniversaladapter
Type	Nok Nok Universal Adapter
Class Name	com.noknok.adapter.ping.UniversalAdapter
Parent Instance Name	None

IdP Adapter

Sign in endpoint	https://ping.noknoktest.com:8443/nnlsignin
Tenant Identifier	forgerock
Use POST method	true
Cookie Domain	noknoktest.com
Enable inline registration	false

Extended Contract

Attribute	username

Adapter Attributes

Mask all OGNL expression log values	false
Pseudonym	username
Unique User Key Attribute	username

Adapter Contract Mapping

Attribute Sources & User Lookup

Data Sources	(None)

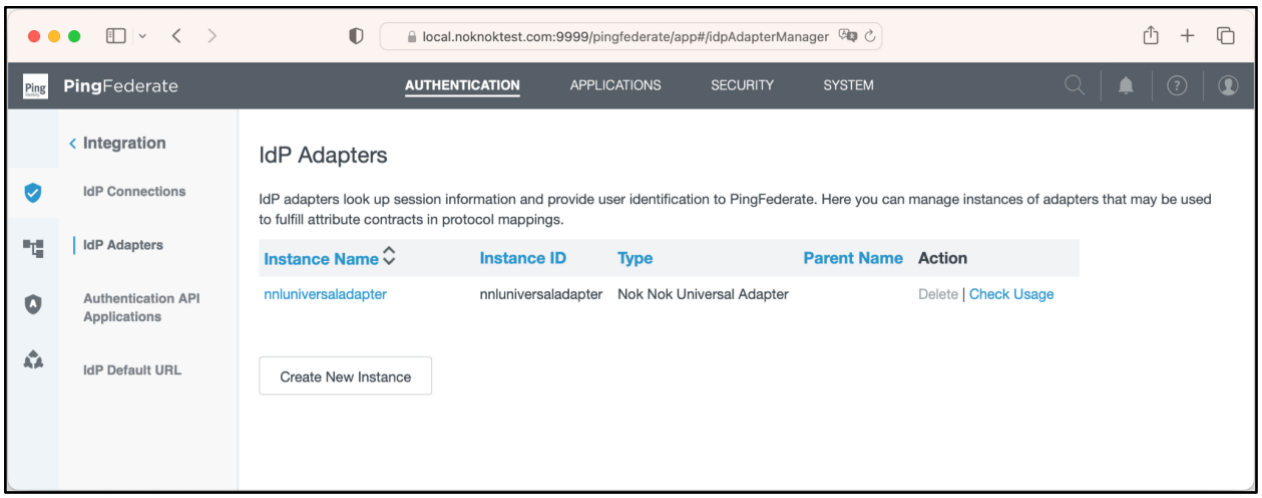
Adapter Contract Fulfillment

username	username (Adapter)

Issuance Criteria

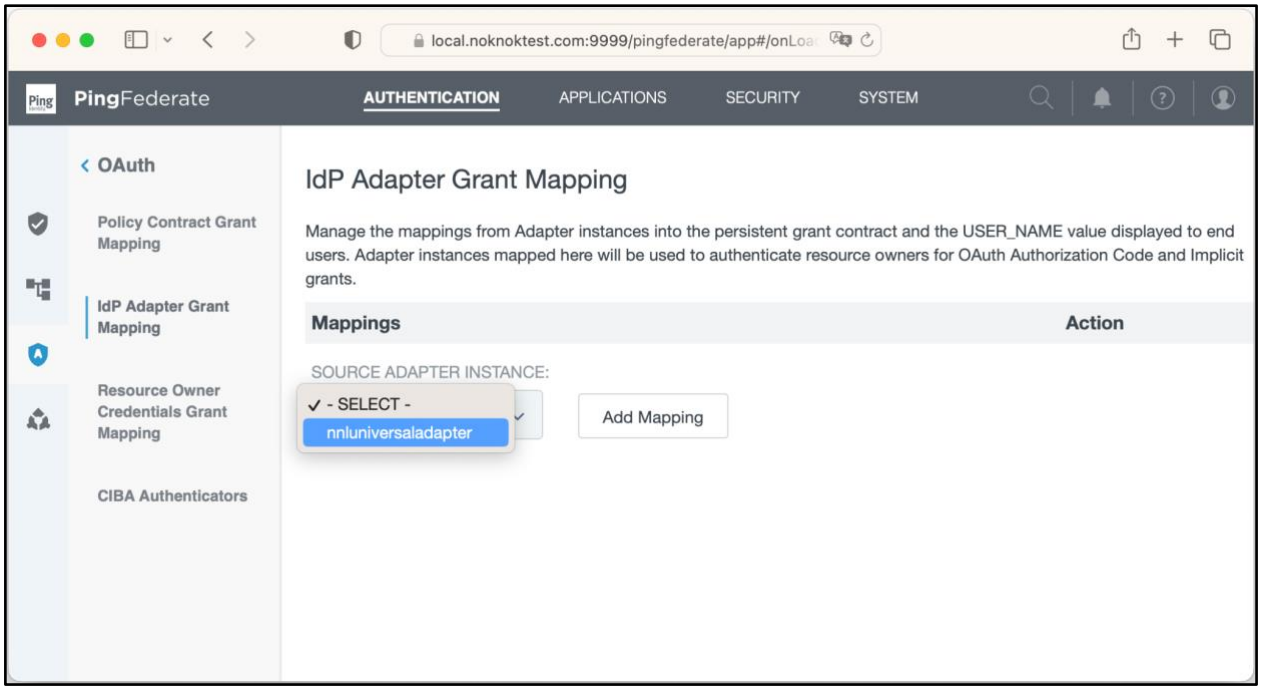
Criterion	(None)

The Nok Nok Adapter instance is now displayed on the Identity Provider (IdP) Adapters screen.

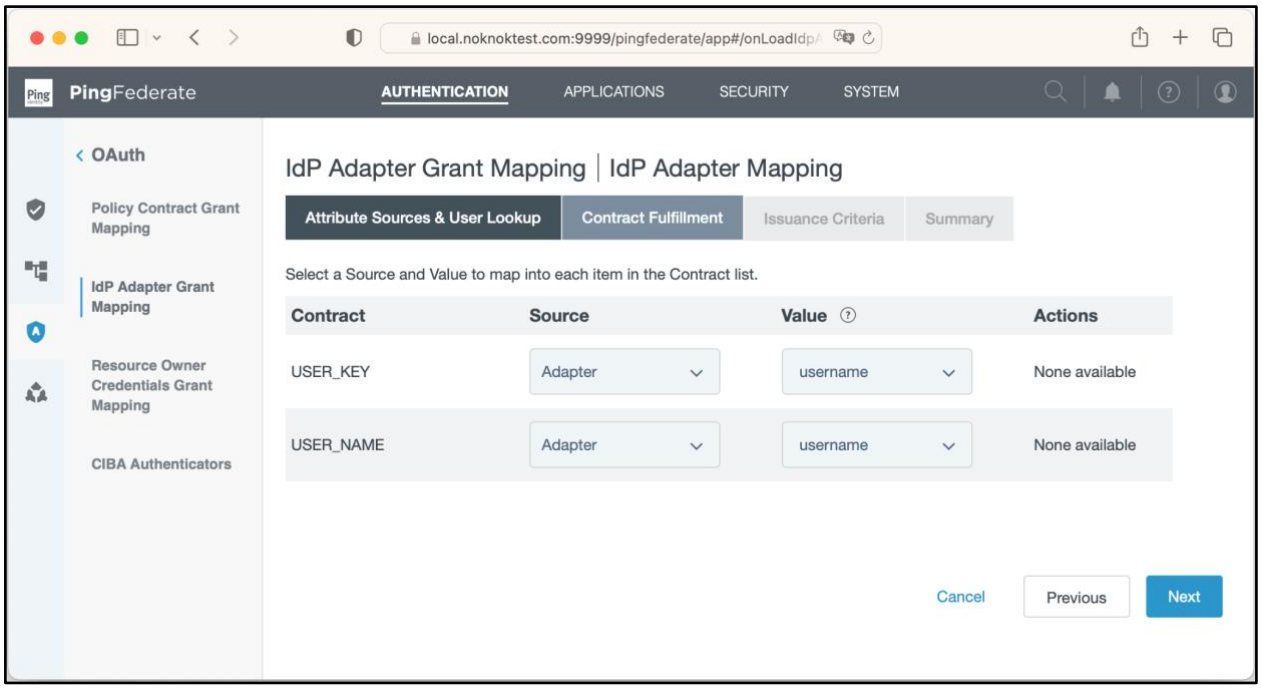


Step 2) Map Adapter Instance to Grant Contract

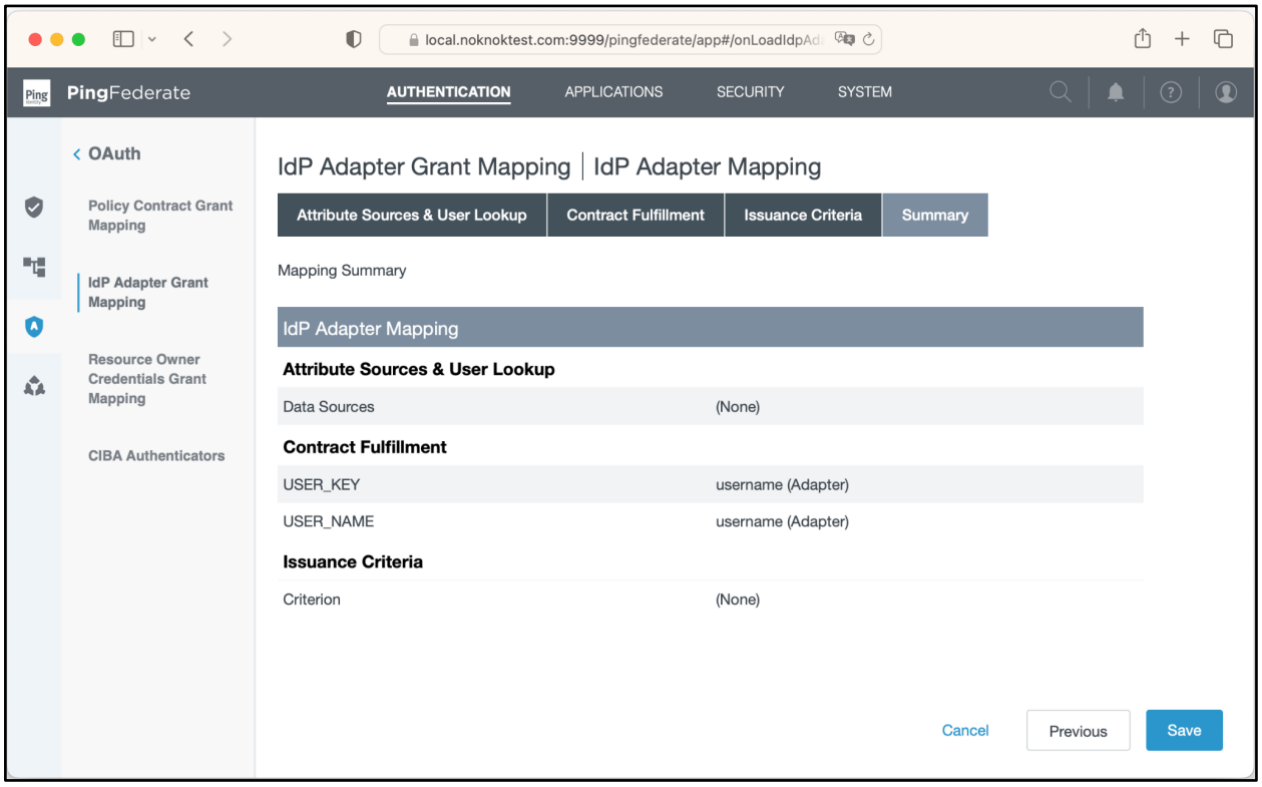
To map the Nok Nok Adapter instance to a persistent grant contract, navigate to **Authentication** > **OAuth** > **IdP Adapter Grant Mapping**. From the list, select the Nok Nok Adapter instance and click **Add Mapping**.



Leave the **Attribute Sources** and **User Lookup** blank and click on the **Contract Fulfilment** tab. Add details as below:

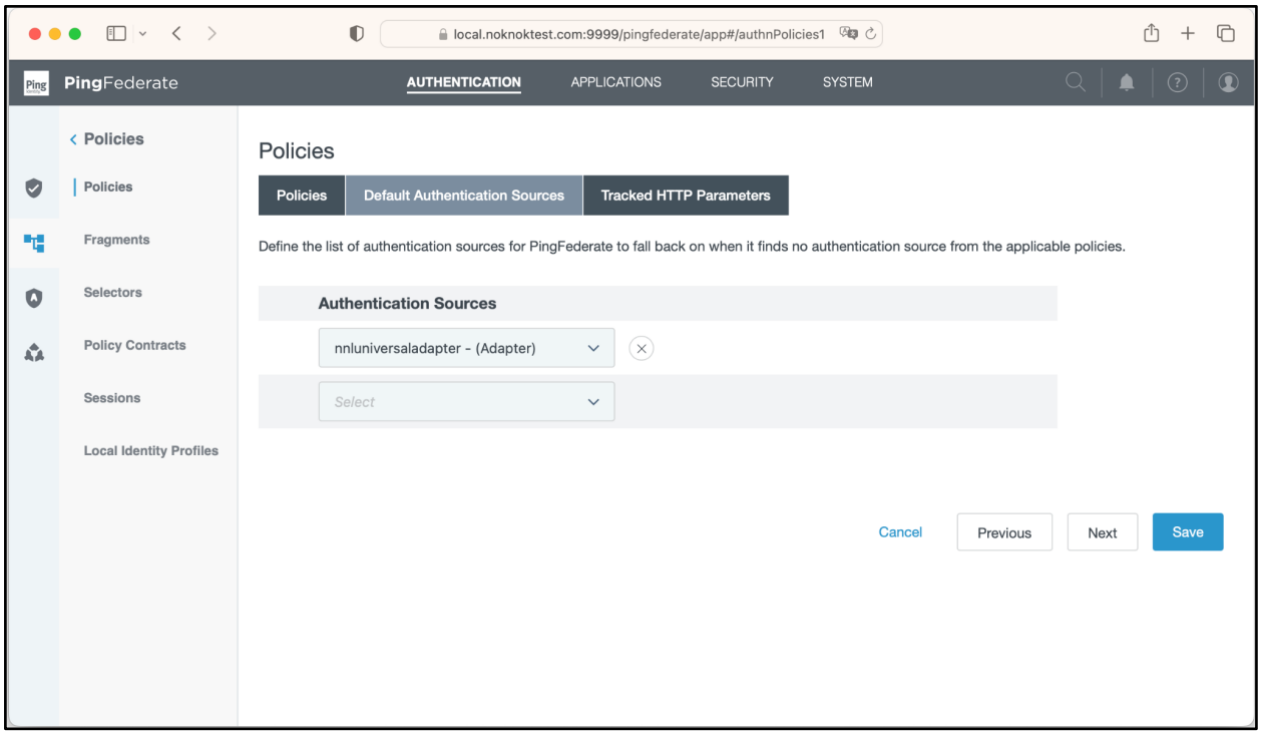


No changes are required on the **Issuance Criteria** tab. Click **Next** to see a summary as below. After reviewing the summary, click **Save**.



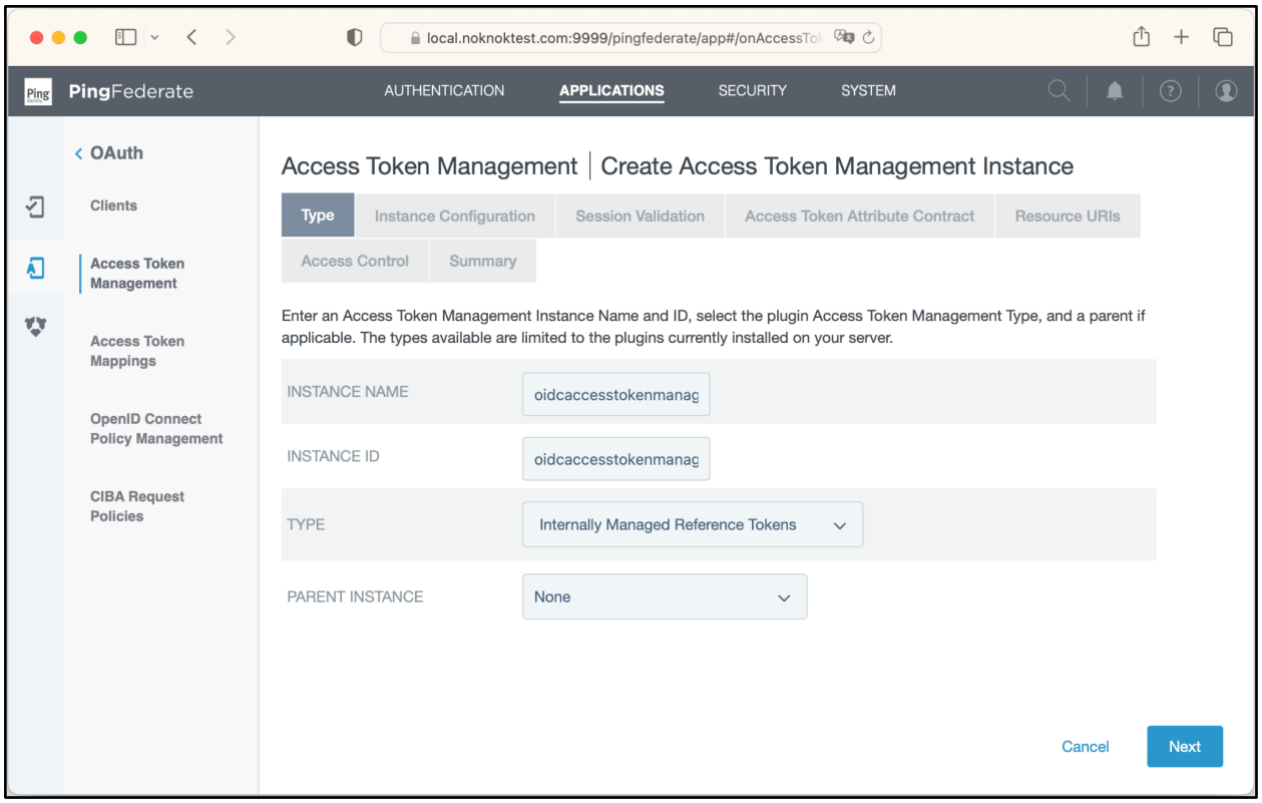
Step 3) Set Up a Default Authentication Source

To add the Nok Nok Adapter as a default authentication source, navigate to **Authentication** > **Policies** > **Default Authentication Sources**. Then select the Nok Nok Adapter instance and click **Save**.

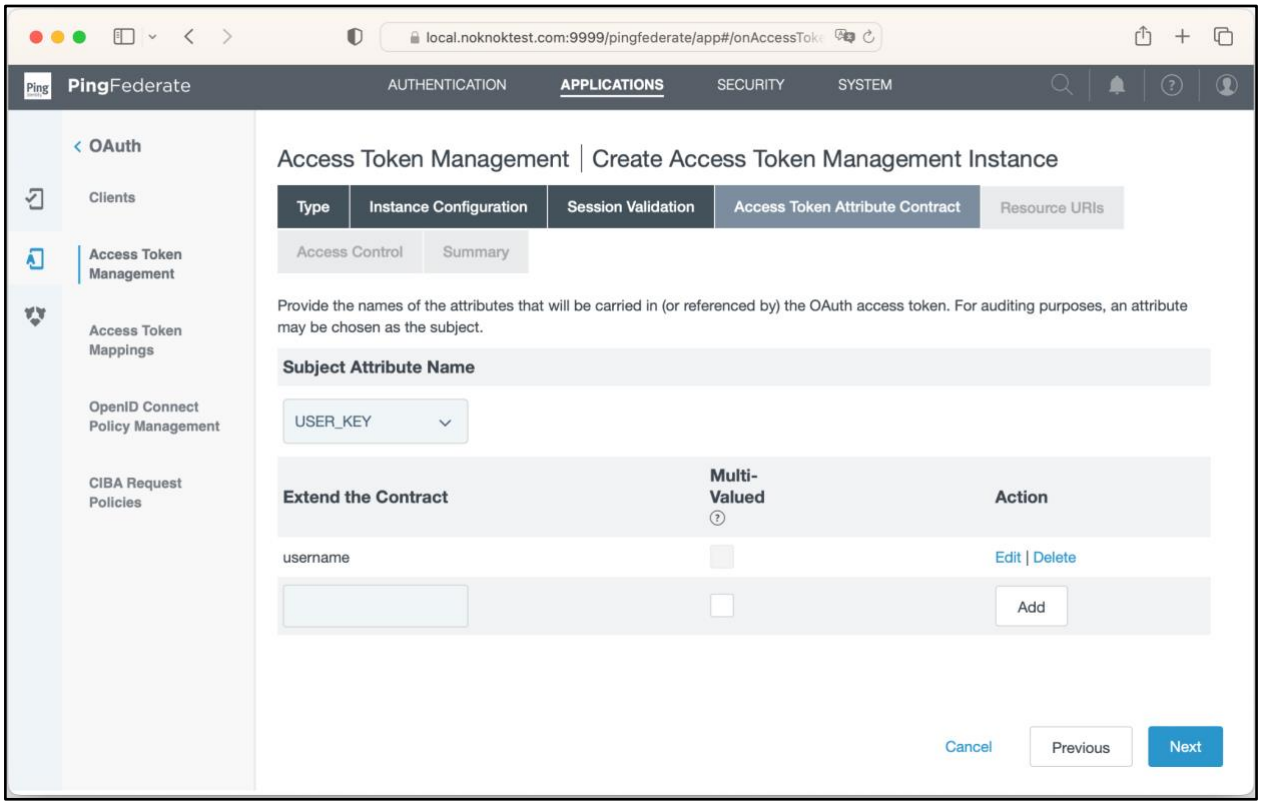


Step 4) Create Access Token Management Instance

To create the access token management instance, navigate to **Applications** > **OAuth** > **Access Token Management** and click **Create New Instance**.



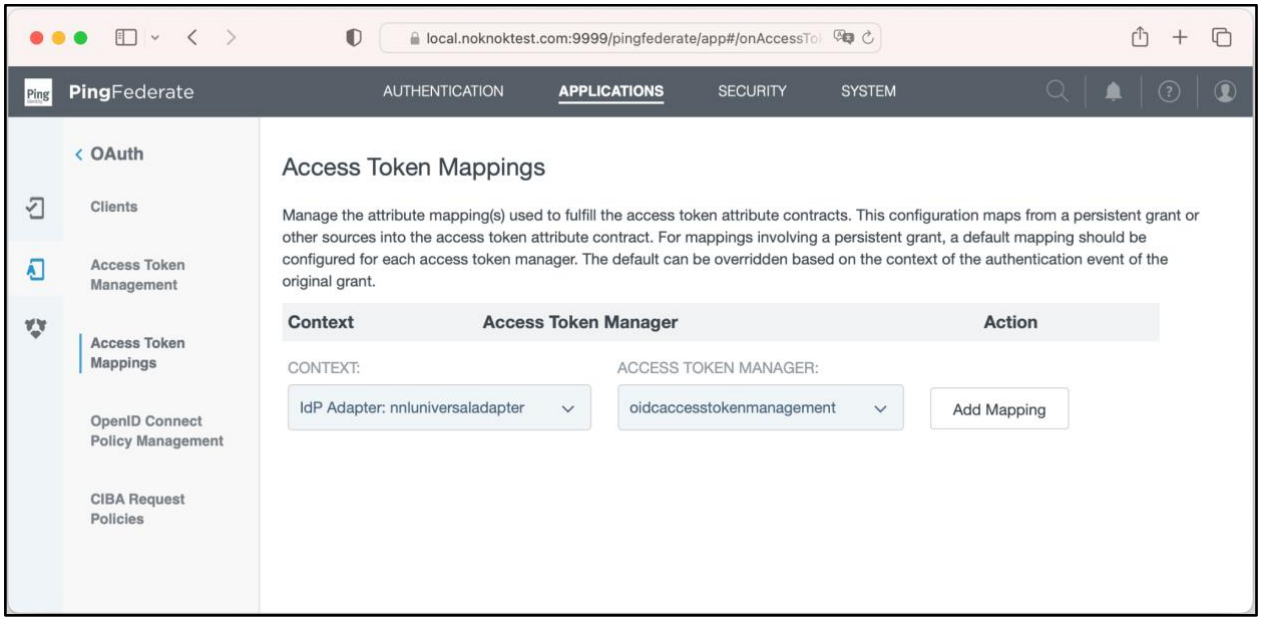
Click **Next** until you reach the **Access Token Attribute Contract** tab. Add the **username** attribute to **Extend the Contract**.



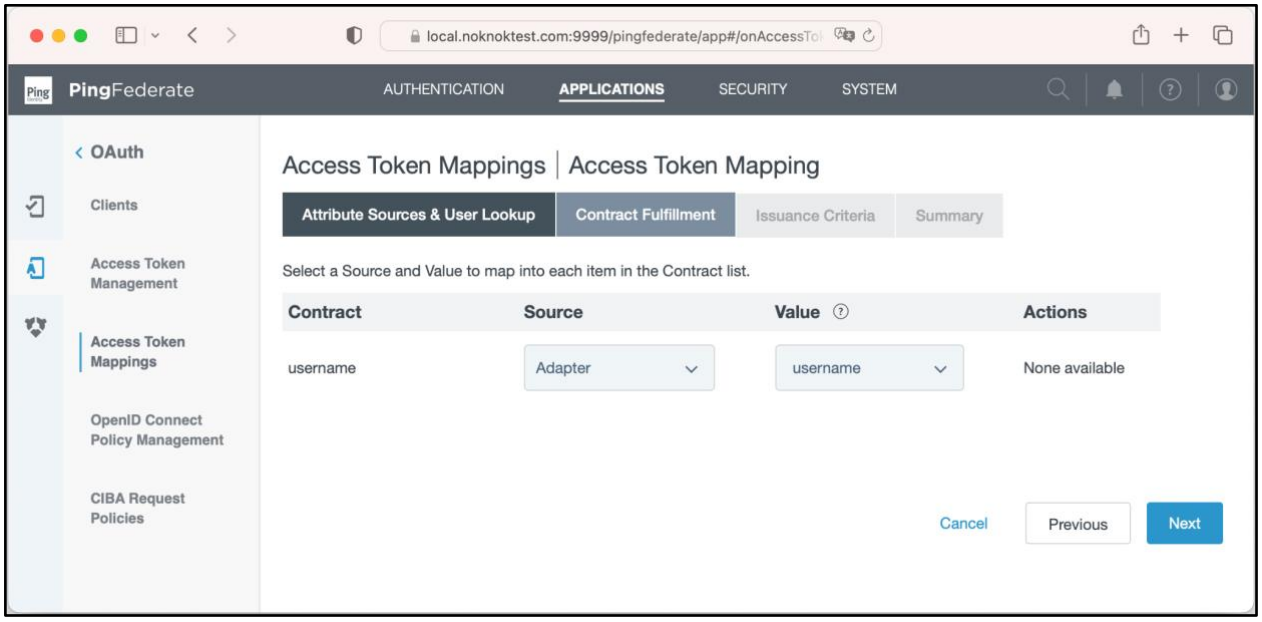
Click **Next** until you reach the **Summary** tab. Click **Save**.

Step 5) Add Access Token Mapping

To add the access token mapping, navigate to **Applications** > **OAuth** > **Access Token Mappings**. Select the **Idp Adapter** CONTEXT and ACCESS TOKEN MANAGER, and click **Add Mapping**:



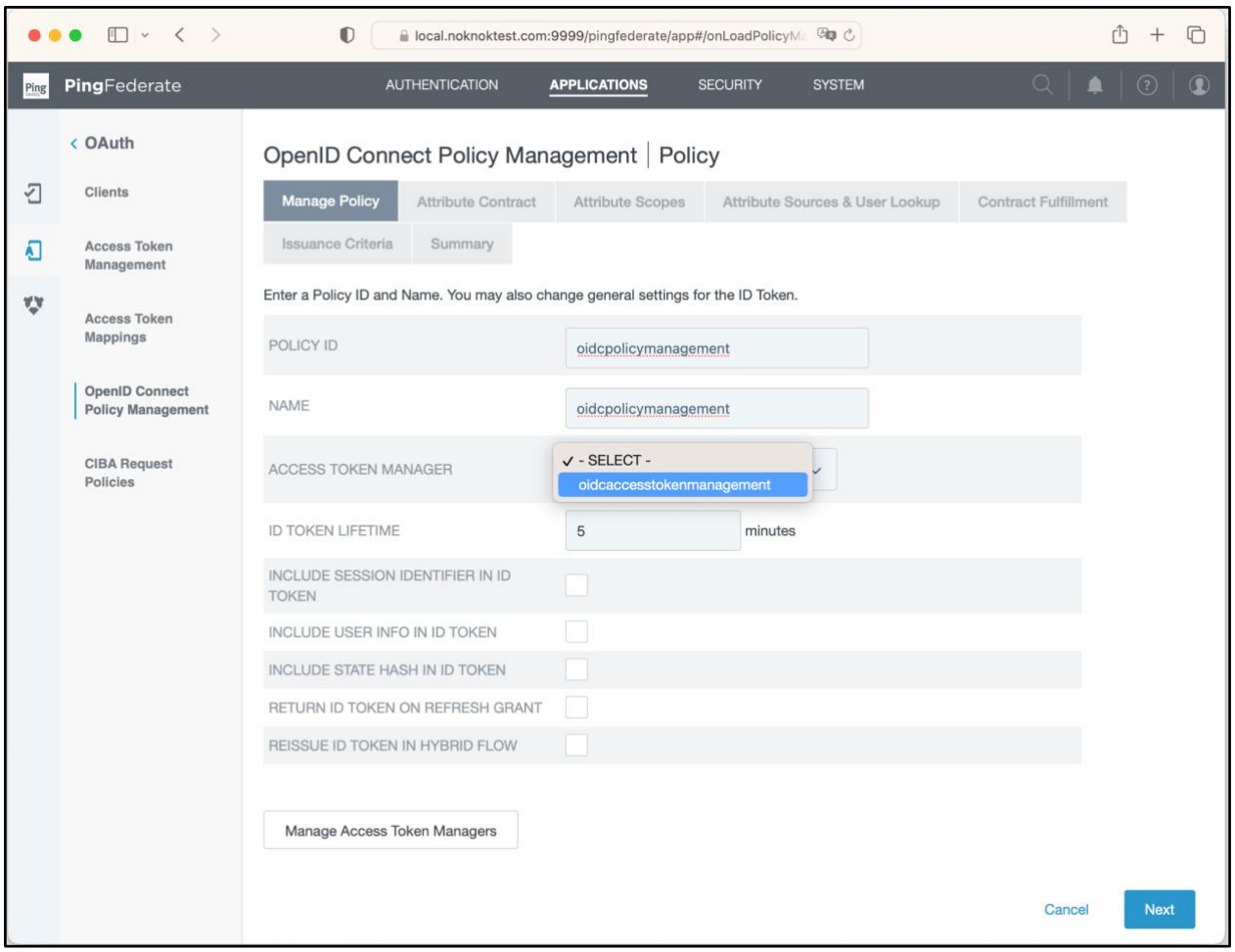
Click **Next** to reach the **Contract Fulfillment** tab. Select Adapter as your **Source**, and select username as your **Value**.



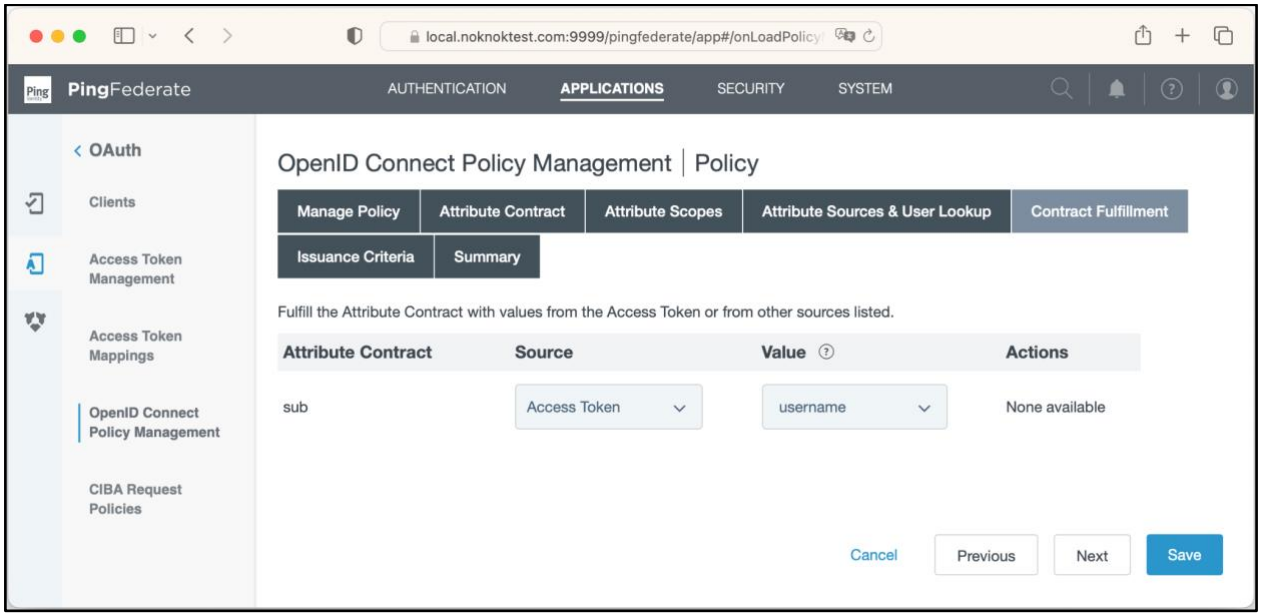
Click **Next** until you reach the **Summary** tab. Click **Save**.

Step 6) Add OpenID Connect Policy

To add the OIDC policy, navigate to **Applications** > **OAuth** > **OpenID Connect Policy Management** and click **Add Policy**.



Click **Next** until you reach the **Contract Fulfillment** tab. Map **sub** claim in **Access Token** to **username** value. Click **Next** until you reach the **Summary** tab. Click **Save**.



Step 7) Add OAuth Clients

Your Nok Nok Adapter is now ready to use. You can add your OAuth clients by referring to the instructions at [Configuring OAuth Clients - PingFederate].

Sign-in Page

If you are integrating FIDO authentication into a web app, or if you are using a webview to integrate it into your mobile app, then you need to have a sign-in web page for your end users. You can either use the `nnlsignin` web application included in the Nok Nok Web App SDK package, or you can use your own sign-in page. In either case, the sign-in page and the Nok Nok Adapter can be hosted on the same or on different origins.

The `nnlsignin` web application included in Nok Nok Web App SDK package implements the required functionality. Test your integration with this `nnlsignin` web application, even if you intend to implement your own sign-in web page later. This section first describes how to configure the `nnlsignin` web app, and then tells how to implement your own PingFederate sign-in page.

`nnlsignin` Web Application

To use the `nnlsignin` app as your sign-in page:

1. Extract `nnlsignin.war` from the Nok Nok Web App SDK package into the `webapps` directory of your chosen application server.
2. Update the information in `nnlsignin/config/config.js`:

```
// Configuration object.
var signinConfig = {};

// The version of the Nok Nok JS App SDK the App was built with.
signinConfig.appsdk_version = {{nnl-version}};

// A suffix can be added to this version to point out changes
// to the configuration within the same release.
signinConfig.version = {{nnl-version}};

signinConfig.apiserver = {server-host-where-nnlappsdk-located};

// If you have a single Nok Nok Server tenant, then the tenant_id can be hard
//  coded. If your deployment supports multi-tenancy, get this from the
//  tenant_id query parameter.
signinConfig.tenant_id = {{tenant_id}};

// Default configuration fields.
signinConfig.nnlappsdk_url = "${apiserver}/nnlappsdk-${appsdk_version}";
signinConfig.storage_endpoint = "${apiserver}/nnlgateway/storage";
signinConfig.reg_endpoint = "${apiserver}/nnlgateway/nnl/${tenant_id}/reg";
signinConfig.auth_endpoint = "${apiserver}/nnlgateway/nnl/${tenant_id}/auth";

// For the OOB method to function, the web_oob_url is mandatory, since code sets
the QrType to be UNIVERSAL_ANY_RP.
signinConfig.web_oob_url = '${apiserver}/nnlsignin/oobrecv.html';

// If sign-in page and PingFederate Server are hosted on different origins,
// set this to the base URI of the PingFederate Server to enable a form post.
// For example: "/path/to/fedserver" or "${host}/path/to/fedserver".
signinConfig.federation_resume_uri = null;
```

Replace {{nnl-version}} with the supplied Web App SDK version number. Replace {{tenant-id}} with the supplied tenant name.

Note: signinConfig.federation_resume_uri is a base URI, full or relative, because the nnlsignin app concatenates dynamic, session specific parameters with this base URI before submitting the form POST. If your sign-in page is hosted on the same origin as the Nok Nok Adapter, leave this null.

3. Update the following context parameters in the nnlsignin/WEB-INF/web.xml file:

Parameter name	Description
quick_mode	Specifies quick authentication mode. Refer to the QuickType enumeration in the JavaScript Client API Docs for a list of possible values. The default value is None.
auth_start_mode	Specifies the mode that the authentication should start in. Refer to the AutoStart enumeration in the JavaScript Client API Docs for a list of possible values. The default value is SIGN_IN.
cookie_domain	Domain to set in the authorization cookie before redirecting to Nok Nok Adapter.

Note: When you configure inline registration in the Nok Nok Adapter, set auth_start_mode to AUTO_FIDO if passkey or security key is used for authentication, or set auth_start_mode to AUTO_ANY if any authentication method is used.

4. **Optional**: You can choose to use your own strings and images, and modify the style of the user interface for the nnlsignin app by editing a JSON UI Configuration file. See the *Nok Nok App UI Customization Guide* for more information.

Implement your own PingFederate Sign-in Page

Once the Nok Nok Adapter is working with the `nnlsignin` web application, you can choose to implement your own PingFederate sign-in page. Please refer to the `submitLogin()` function in file `Controller.js` of the `nnlsignin` app to see how to implement FIDO authentication in your page. The `submitLogin()` function also shows how to do a form POST after successful authentication. That JavaScript code makes use of the Nok Nok Javascript App SDK to initiate FIDO authentication.

The Nok Nok Adapter redirects to your PingFederate sign-in page with the following URL query parameters:

URL Query Parameter	Description
`goto <resume_url>`	If you are hosting the sign-in page on the same origin as the Nok Nok Adapter, then this parameter is required.
`resume_path <relative_url>`	If you are hosting the sign-in page on an origin that is different from the Nok Nok Adapter, then this parameter is required.
`username`	Add this parameter only if the user is pre-authenticated, as is the case with a step-up authentication.
`tenant_id`	Add this parameter only if your Nok Nok deployment has more than one tenant.

Sample URL query if you are hosting the sign-in page on the same origin:

```
https://<same-origin-host-name>/nnlsignin/?goto=https%3A%2F%2F<same-origin-host-name>%3A9031%2Fas%2FPB2TAsJfGf%2Fresume%2Fas%2Fauthorization.ping&tenant_id=default
```

Sample URL query if you are hosting the sign-in page on a different origin:

```
https://<different-origin-sign-in-page-host-name>/nnlsignin/index.jsp?resume_path=%2Fas%2F1QmfS2twMi%2Fresume%2Fas%2Fauthorization.ping&tenant_id=eval
```

If the authentication is successful, the Nok Nok JavaScript App SDK returns a JWT in `sessionData.sessionKey`.

Your Javascript code sends the JWT to the Nok Nok Adapter in one of the following places:

- Authorization header
- Authorization cookie
- form POST

If you are using the `goto` URL query parameter, send the JWT as the Authorization header or cookie. This header or cookie is sent with the request to the PingFederate resume URL. If you are using the `resume_path` URL query parameter, send the JWT as the `token` parameter.

For more information on how to add FIDO functionality to your sign-in page, refer to the *Nok Nok App SDK for JavaScript Developer Guide*.

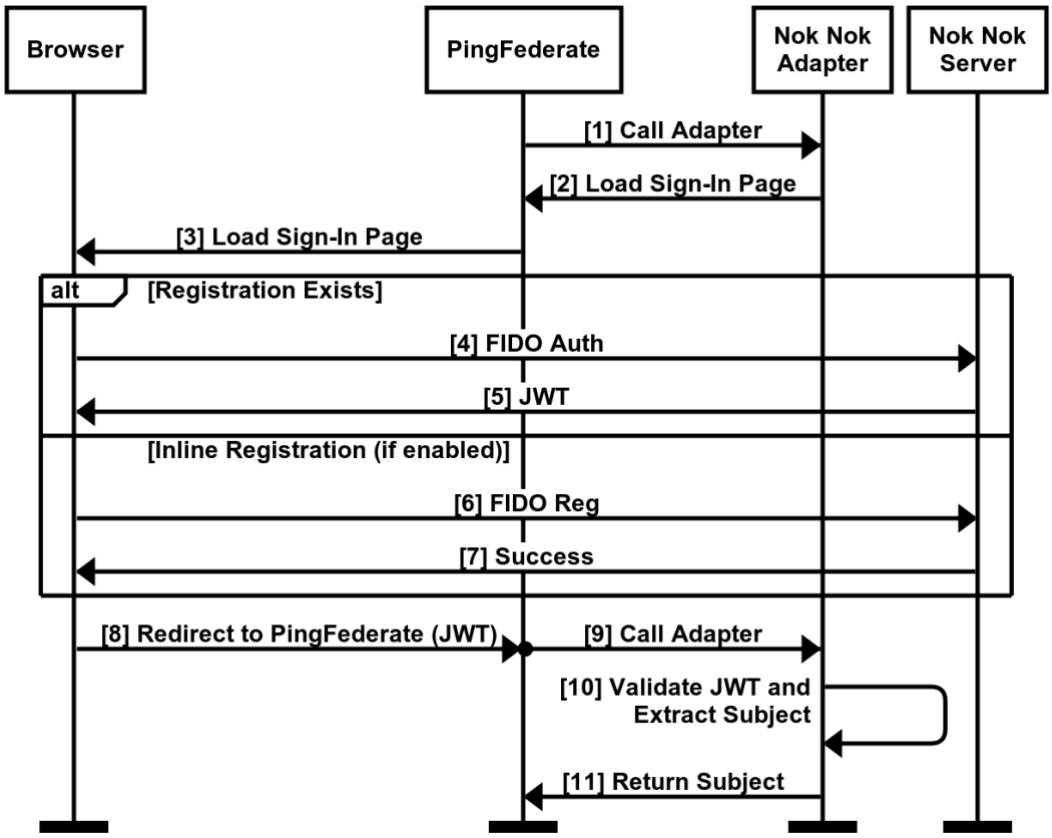
Initial FIDO Registration through OIDC

Before a user can sign in with FIDO, the user must first authenticate with an external identity provider and then register a FIDO authenticator. The Nok Nok API Server incorporates built-in OIDC client functionality, facilitating access to the FIDO registration page after a user has successfully authenticated with an external identity provider. For comprehensive guidance on configuring External Identity Providers (IdPs), please refer to the *Nok Nok S3 Authentication Suite Configuration Guide,* section *External Identity Provider Configuration* in Appendix B.

Appendix A: Sequence Diagrams

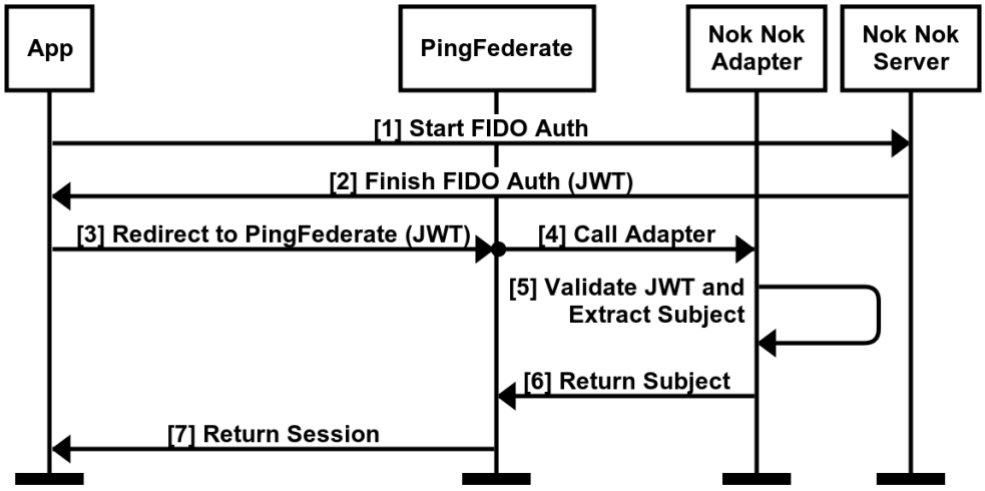
Web App Sign In Integration

This sequence diagram illustrates the flow for FIDO authentication in a web app.



1. PingFederate calls the Nok Nok Adapter.
2. The Nok Nok Adapter redirects to the sign-in page.
3. PingFederate returns a redirect to the sign-in page
4. If a FIDO registration exists, FIDO Authentication is performed and response sent to Nok Nok Server.
5. The Nok Nok Server returns a JWT.
6. If a FIDO registration does not exist and inline registration is enabled, FIDO registration is performed and the response sent to Nok Nok Server.
7. The Nok Nok Server returns success.
8. Sign in page redirects to PingFederate with the JWT.
9. PingFederate calls the Nok Nok Adapter.
10. The Nok Nok Adapter validates the JWT and extracts the username and any custom attributes from the JWT.
11. This username is returned to the PingFederate.

Native App Sign In Integration



This sequence diagram illustrates the authentication flow for FIDO authentication in a native app:

1. Using the Nok Nok App SDK, your app sends a "start FIDO authentication" to the Nok Nok Server.
2. The FIDO authentication finishes and the Nok Nok Server returns a JWT.
3. The app redirects to the PingFederate.
4. PingFederate calls the Adapter.
5. The Adapter validates the JWT and extracts the username and any custom attributes from the JWT.
6. This username is returned to PingFederate.
7. PingFederate sends the app the session.

Appendix B: Upgrading

Upgrading from Previous Version to October 2024 Version

- In a multi-tenant deployment, the SIGN IN ENDPOINT configuration parameter must now specify the TENANT IDENTIFIER for the target tenant in the Nok Nok Server. In the previous release, the Nok Nok Adapter automatically added the TENANT IDENTIFIER to the SIGN IN ENDPOINT as a URL parameter, but in the current release the customer needs to add the TENANT IDENTIFIER to the SIGN IN ENDPOINT as the URL parameter `tenant_id`. For example, using the PingFederate Admin Console, the customer now configures the SIGN IN ENDPOINT for the `nnlsignin` app as:
  `<SIGN IN ENDPOINT>?tenant_id=<TENANT IDENTIFIER>`

- If your Nok Nok deployment only has one tenant named `default`, you can skip this step because your directory structure has not changed from the previous version.
  The Nok Nok Adapter now uses the TENANT IDENTIFIER configuration parameter to find the JWT configuration `jwt_config.json` in your directory structure. This enables the Nok Nok Adapter to have different JWT configurations for different tenants. See Installation section Step 4. for details.