

NativeSPAddIdP

Talk to a new IdP

The amount of configuration the SP needs to talk to a new IdP depends on how much special treatment that IdP will require compared to your defaults, and various aspects of the overall architecture, such as how [discovery](#) is being handled. The basic case is very simple: add the IdP's metadata to a metadata sourcereferenced by a `<MetadataProvider>` from `shibboleth2.xml`, and you're done.

If you're using a single IdP, then you need to ensure its `entityID` is set in the `<SSO>` element to route requests to it, but in many cases you're adding "another" IdP, so by definition you have to work through the discovery problem and that's a very deployment-specific topic. Truly federated services need to accept a true discovery solution such as the [Embedded Discovery Service](#). Non-federated applications often rely on host or path-driven rules for identifying the IdP, in which case you may be able to address this by adding an `entityID` setting to the `<RequestMap>` or an Apache [command](#) as applicable.

If you do need to treat an IdP specially in one of the following ways, read below:

Use a different SP entityID

Add a `<RelyingParty>` element to the `<Application>` configuration with a new `Name` matching the `entityID` of the IdP. The SP will name itself by a specified `entityID` attribute when it talks to the IdP with that `Name`.

This won't work if the legacy WAYF discovery protocol is used, but it will work with a modern DS.

Different cryptography requirements

Add a `<RelyingParty>` element to the `<Application>` configuration with a new `Name` matching the `entityID` of the IdP. Use a property like `keyName="specialKey"` to refer to a specific `<CredentialResolver>` containing a non-default key. You can also change the default `encryption` and `signing` settings, or the use of TLS to authenticate to other providers, but this is rarely required.

Special Attribute Rules

To apply special attribute filtering rules for a this IdP, add the rules to `attribute-policy.xml`. No `<RelyingParty>` settings are necessary.