NativeSPSessionCreationParameters

The Shibboleth SP does not have an application API per se, but the SessionInitiator mechanism supports a simple redirect protocol capable of triggering, and influencing, the creation of authentication requests.

Initiator Protocol

This protocol supports a small set of query string parameters that correspond to identically named attributes that can be supplied either directly on a `<SessionInitiator>` element or as content settings on a per-resource basis.

When a query string parameter is used, it overrides any other source of the same setting/property.

Not all SessionInitiator handlers support all the possible parameters. In fact, most are specific to the `SAML2` handler. Unsupported parameters are ignored.

- `entityID` (URI)
  - The IdP to request authentication from.
- `target` (absolute URL)
  - The URL to return the user to after authenticating. If unspecified, the `homeURL` attribute for the application is used.
- `acsIndex` (string)
  - The `index` value of the `<md:AssertionConsumerService>` element to instruct the IdP to use in returning an assertion to the SP.
- `forceAuthn` (boolean) (defaults to false) (`SAML2` only)
  - Establish a value for the `ForceAuthn` attribute of the `<samlp:AuthnRequest>`. This asks for forced reauthentication by the IdP (bypassing SSO).
- `isPassive` (boolean) (defaults to false) (`SAML2` and `SAMLDS` only)
  - Establish a value for the `IsPassive` attribute of the `<samlp:AuthnRequest>` or the `IsPassive` parameter of the DS redirect.
- `authnContextClassRef` (URI) (`SAML2` only)
  - Requests that a particular authentication context class be used by the IdP. As of V2.5, this can be a whitespace-delimited list of classes to request.
- `authnContextComparison` ("exact", "minimum", "maximum", "better") (defaults to "exact") (`SAML2` only)
  - Indicates the required relationship between a requested context class and the resulting form of authentication.
- `NameIDFormat` (URI) (`SAML2` only) (Version 2.3 and Above)
  - If set, causes the authentication request to carry a `saml:NameIDPolicy` with a `Format` containing the provided value. If the receiving IdP can not fulfill this requirement it should return an error response.
- `SPNameQualifier` (URI) (`SAML2` only) (Version 2.3 and Above)
  - If set, causes the authentication request to carry a `saml:NameIDPolicy` with an `SPNameQualifier` containing the provided value. If the receiving IdP can not fulfill this requirement it should return an error response.
- `discoveryPolicy` (string) (`SAMLDS` only) (Version 2.5 and Above)
  - Used as input to some discovery protocols that take parameters modifying discovery behavior. In the case of the `type="SAMLDS"`SessionInitiator, this is passed as a `policy` parameter value.
- `template` (base64-encoded SAML `<AuthnRequest>` message) (`SAML2` only) (Version 2.6 and Above)
  - If supplied, the eventual SAML request is constructed based on the message supplied, apart from per-request information or settings supplied directly in the configuration or as parameters. Allows a message to be constructed externally with extensions or dynamic content, and then re-issued by the SP.

Examples

The redirection examples shown are illustrated by way of the HTTP Location header that would be returned to a client by an application. Refer to your programming environment's documentation for information on how to generate redirects and produce such a header. Note that you should always be sure to URL-encode any parameter values that you append.

The examples also assume that a SessionInitiator exists at the location "/Login", which is the usual default.

The most common scenario is to simply ask for a login while providing a resource to return the client to afterwards. Typically, this is the resource from which the redirect is generated.

Request a Session and Return to https://sp.example.org/resource.asp

```
Location: https://sp.example.org/Shibboleth.sso/Login?
  target=https%3A%2F%2Fsp.example.org%2Fresource.asp
```

Another common case is to specify the IdP to use. This is a simple way to implement user selection of an IdP from among a small set, for example clicking on a choice of logos. Not coincidentally, the `SAMLDS` handler is implemented by routing the result of the discovery process back to itself with the `entityID` parameter set.

Request a Session Using the IdP Named https://idp.example.org/idp/shibboleth

```
Location: https://sp.example.org/Shibboleth.sso/Login?
  target=https%3A%2F%2Fsp.example.org%2Fresource.asp&
  entityID=https%3A%2F%2Fidp.example.org%2Fidp%2Fshibboleth
```