Cookie Connector

Contents

Cookie Connector

The Cookie connector lets you set and retrieve session cookies in your PingOne DaVinci flow.

Setup

Resources

For information and setup help, see the following documentation:

- DaVinci documentation:
    - *Adding a connector*
    - *Using DaVinci flow templates*

Configuring the Cookie connector

Add the connector in DaVinci as shown in *Adding a connection*, then configure it as follows.

Connector configuration

HMAC Signing Key

The Base64-encoded, 256-bit key that the connector uses to sign the session cookies. This prevents unauthorized agents from gaining access by guessing a session ID.
To use this, enable the **Sign Cookies with HMAC Key** option in the **Set a Session Cookie** capabilities.

Using the connector in a flow

Session cookie creation

The connector has two capabilities that allow you to set session cookies:

- Set a Session Cookie
- Set a Session Cookie (Non-User)

These two capabilities are identical, except the **Non-User** variant doesn't require a user to already have been identified in the flow.

No special flow configuration is needed. Add the capability you want and populate its properties according to the help text.

You can include custom claims in the session cookie. In the **Session Cookie Custom Claims** area, click **+ Field**, select a claim from the list, and enter a value. To delete a custom claim from the list, click **Edit**.

Session cookie retrieval

The connector has two capabilities that allow you to set session cookies:

- Check a Session Cookie
- Check a Session Cookie (Non-User)

No special flow configuration is needed. Add the capability you want and populate its properties according to the help text.

Capabilities

Set Session Cookie

Sets an opaque session cookie so that the user is not asked to authenticate again during the flow.

Details

Properties

cookieName `textField`

useSecureCookie `toggleSwitch`

useHttpOnlyCookie `toggleSwitch`

cookieExpiresInSeconds `textField`

cookieDomain `textField`

cookieSameSite `dropDown`

signCookie `toggleSwitch`

setCookieClientSide `toggleSwitch`

claimsNameValuePairsSessionCookie `selectNameValueListColumn`

useSessionTokenFlag `toggleSwitch`

sessionToken `textField`

Input Schema

default `object`

userInfo `object`

User with which the cookie is to be associated.

ip `string` `minLength: 1` `maxLength: 50`

Ip address of the user in current session.

userAgent `string` `minLength: 1` `maxLength: 250`

Information about browser, OS, etc. of user in current session.

skOpenId `object`

Object containing client id of user.

origin `string` `minLength: 0` `maxLength: 500`

Origin

originCookies `string` `minLength: 0` `maxLength: 5000`

OriginCookies

Check Session Cookie

Determine if a session cookie exists in the flow. If it exists, continue with the flow, or ask the user to provide additional authentication information.

Details

Properties

cookieName `textField`

enforceClientIP `toggleSwitch`

enforceFlowIdMatch `toggleSwitch`

Resolve to User `toggleSwitch`

Attempt to resolve cookie to a user, which will then be available in {{global.userInfo}} parameter

Input Schema

default `object`

userAgent `string minLength: 1 maxLength: 250`

Information about browser, OS, etc. of user in current session.

ip `string minLength: 1 maxLength: 50`

Ip address of the user in current session.

cookies `object`

List of cookies associated with the user.

skOpenId `object`

Object containing client id of user.

origin `string minLength: 0 maxLength: 500`

Origin

originCookies `string minLength: 0 maxLength: 5000`

OriginCookies

Output Schema

output `object`

claims `object`

properties `object`

userId `string`

connectionId `string`

companyId `string`

ip `string`

userAgent `number`

flowId `string`

client_id `string`

signCookie `boolean`

createdDate `number`

cookieExpiresInSeconds `number`

loa `number`

Set Session Cookie (Non User)

Set an opaque session cookie with a set of custom claims.

Details

Properties

cookieName `textField`

useSecureCookie `toggleSwitch`

useHttpOnlyCookie `toggleSwitch`

cookieExpiresInSeconds `textField`

cookieDomain `textField`

cookieSameSite `dropDown`

signCookie `toggleSwitch`

setCookieClientSide `toggleSwitch`

claimsNameValuePairsSessionCookie `selectNameValueListColumn`

useSessionTokenFlag `toggleSwitch`

sessionToken `textField`

Input Schema

default `object`

> **ip** `string minLength: 1 maxLength: 50`
>
> > Ip address of the user in current session.
>
> **userAgent** `string minLength: 1 maxLength: 250`
>
> > Information about browser, OS, etc. of user in current session.
>
> **skOpenId** `object`
>
> > Object containing client id of user.
>
> **origin** `string minLength: 0 maxLength: 500`
>
> > Origin
>
> **originCookies** `string minLength: 0 maxLength: 5000`
>
> > OriginCookies

Check Session Cookie (Non User)

Check if a specific session cookie exists and retrieve the custom claims for decision-making purposes.

Details

Properties

> **cookieName** `textField`
>
> **enforceClientIP** `toggleSwitch`
>
> **enforceFlowIdMatch** `toggleSwitch`

Input Schema

default `object`

> **userAgent** `string minLength: 1 maxLength: 250`
>
> > Information about browser, OS, etc. of user in current session.
>
> **ip** `string minLength: 1 maxLength: 50`
>
> > Ip address of the user in current session.
>
> **cookies** `object`
>
> > List of cookies associated with the user.
>
> **skOpenId** `object`
>
> > Object containing client id of user.
>
> **origin** `string minLength: 0 maxLength: 500`

Origin

originCookies `string minLength: 0 maxLength: 5000`

OriginCookies

Output Schema

output `object`

claims `object`

properties `object`

ip `string`

userAgent `number`

flowId `string`

client_id `string`

signCookie `boolean`

createdDate `number`

cookieExpiresInSeconds `number`

loa `number`