# “Mini-ML”: How to SAML-Enable Your Custom Portal for Domo

Customers wanting to implement Domo Embed without implementing a 3rd party SSO tool (such as Okta or Ping) can still use SAML to integrate with Domo. This guide covers the key integration features that a custom portal must provide to embed Domo cards or to offer full Domo in an iFrame.

**1.  The custom portal authenticates the user and maintains profile information**

The portal should act as the SAML Identity Provider (IdP) and authenticate the user. Because Domo requires email address as the unique identifier, the portal must have an email address associated with each user.

**Note:** *If your portal hosts users from different companies, then for security reasons, you should avoid using personal names in email addresses and instead use opaque identifiers. Certain Domo features include social components that show usernames or email addresses of other users in the instance.*

The portal may also send Domo other profile information such as group memberships, role, location, department, title, employee id, etc. See **Step 4** for a full list of attributes the portal can send to Domo.

**2. Accept Domo’s SAML AuthN request**

When the portal page with an embedded Domo resource (or iFrame) loads, the user’s browser will attempt to access Domo. Because it doesn’t have a valid session cookie yet, it will redirect back to the portal site (or wherever the Identity Provider is) for authentication with a SAML AuthnRequest, which looks like this:

SAMLRequest=<saml2p:AuthnRequest xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol" AssertionConsumerServiceURL="https://INSTANCE- NAME.[domo.com/auth/saml](http://domo.com/auth/saml)" ID="\_2cf656a1-dcc5-45ae-87e9-fcf2f192de2d" IssueInstant="2017-09-22T20:18:34.623Z" ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Version="2.0"><saml2:Issuer xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">[http://PORTALADDRESS+ENTITY-ID](http://portaladdress+entity-id/)</saml2:Issuer></saml2p:AuthnRequest>

RelayState=std

The AuthN Request is Base 64 encoded. You can also have Domo sign this request (see the “Set Up Domo SSO” section for more information). Also, Domo always sends a RelayState value. Your portal IdP should capture this state and return it with the SAML Response.

**Notes:**

* *This is an abbreviated AuthN Request; for an extended example, see* Appendix A***.***

**3. Sign and Post a SAML assertion**

The portal responds to the AuthnRequest by creating a SAML assertion, signing it, and redirecting the user back to Domo. The SAML response **must be signed** and Base 64 encoded or Domo will ignore the request.

The SAML response looks like this:

SAMLResponse=<?xml version="1.0" encoding="UTF-8"?>

<saml2:Assertion

    xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion" ID="id320480196470811401327338190" IssueInstant="2017-09-22T20:42:35.679Z" Version="2.0">

    <saml2:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">[http://www](http://www/).[PORTAL-ADDR.com/Issuer](http://portal-addr.com/Issuer)</saml2:Issuer>

    <saml2:Subject>

        <saml2:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent">userName</saml2:NameID>

        <saml2:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">

            <saml2:SubjectConfirmationData NotOnOrAfter="2017-09-22T20:47:35.679Z" Recipient="<https://INSTANCE-NAME.domo.com/auth/saml>"/>

        </saml2:SubjectConfirmation>

    </saml2:Subject>

    <saml2:Conditions NotBefore="2017-09-22T20:37:35.679Z" NotOnOrAfter="2017-09-22T20:47:35.679Z">

        <saml2:AudienceRestriction>

            <saml2:Audience>https://INSTANCE-NAME.[domo.com/auth/saml](http://domo.com/auth/saml)</saml2:Audience>

        </saml2:AudienceRestriction>

    </saml2:Conditions>

    <saml2:AuthnStatement AuthnInstant="2017-09-22T20:42:35.679Z">

        <saml2:AuthnContext>

            <saml2:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport</saml2:AuthnContextClassRef>

        </saml2:AuthnContext>

    </saml2:AuthnStatement>

    <saml2:AttributeStatement>

        <saml2:Attribute Name="email" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">

            <saml2:AttributeValue

                xmlns:xs="<http://www.w3.org/2001/XMLSchema>"

                xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>" xsi:type="xs:string">user.email

            </saml2:AttributeValue>

        </saml2:Attribute>

        <saml2:Attribute Name="group" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">

            <saml2:AttributeValue

                xmlns:xs="<http://www.w3.org/2001/XMLSchema>"

                xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>" xsi:type="xs:string">GroupName Match Regex ".\*" (case-sensitive)

            </saml2:AttributeValue>

        </saml2:Attribute>

    </saml2:AttributeStatement>

</saml2:Assertion>

RelayState=/page/123456789

**4. (Optional) Set up Group & Attribute assertions**

The SAML assertion can contain groups and other attributes from the IdP (in the case, the custom portal). Here’s a list of attributes that Domo accepts in a SAML assertion:

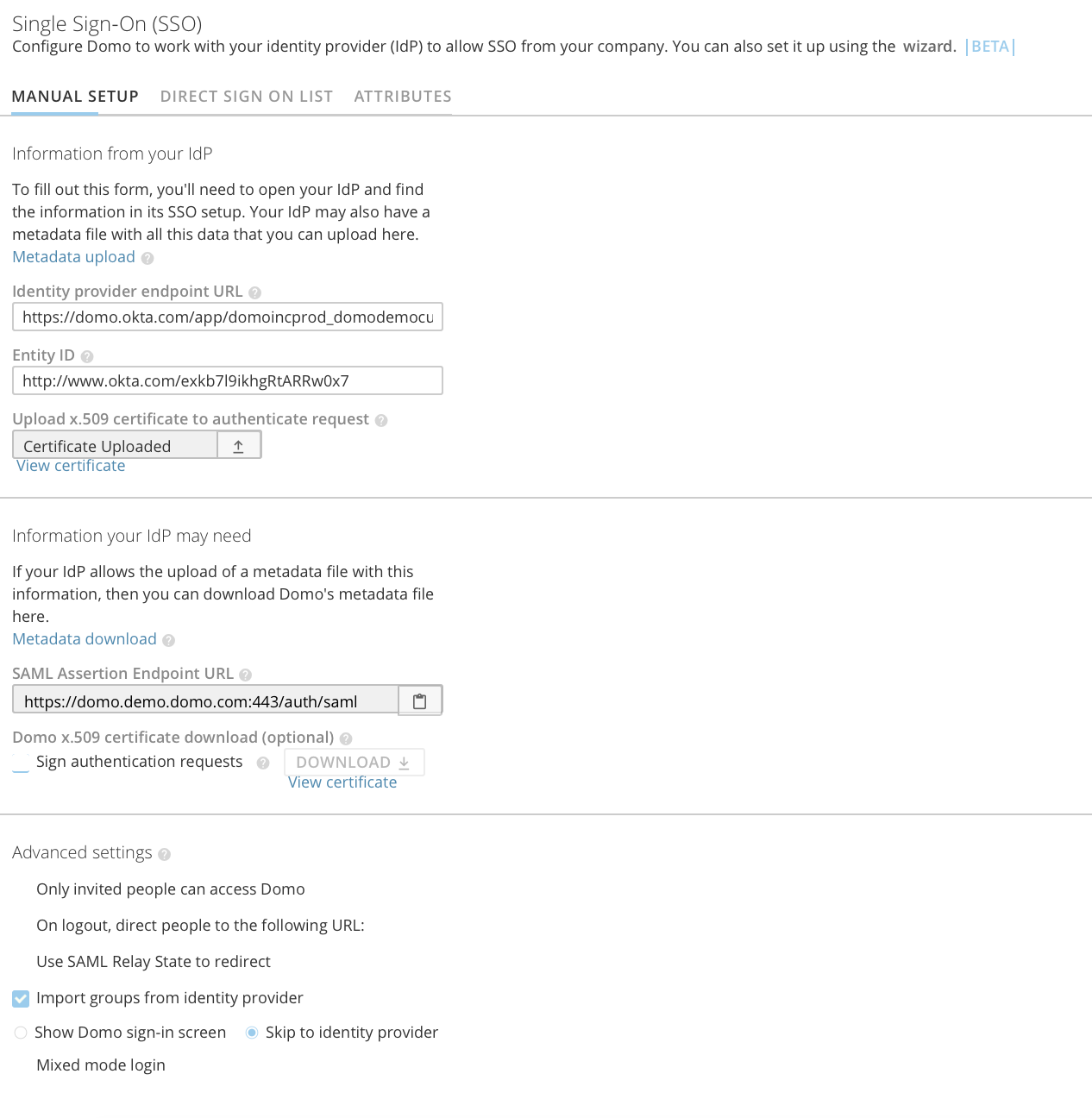
* email
* secondary email
* group (must be listed as an array of strings)
* role
* employee ID
* hire date
* name
  + (the portal can send “name.personal” and/or “name.family” and Domo will concatenate these valuese into the “name” field)
* title
* department
* location
* user phone
* desk phone
* locale
* timezone

**Note:** *See* Appendix A *for an example of how to represent these attributes in a SAML assertion.*

# Set up Domo SSO

Steps to enable Domo with your custom portal:

1. Log in to Domo as Admin
2. Go to the Admin section and select Security | Single Sign-On (SSO)
3. Configure this page:



**Notes:**

* *Enable “Use SAML Relay State to Redirect” – this provides deep linking to the embedded card after authentication*
* *The “Skip to identity provider” feature is automatic with embed links, so this option applies only to users logging into Domo directly.*
* *Note: You can use the “Direct Sign-On List” for portal admins and Domo content creators to log in directly to Domo. That way, regular Domo users (who don’t access Domo embed through the portal) can log in to full Domo without using the custom portal IdP.*

# Deep Linking to Cards and Pages using RelayState

Domo uses the RelayState value to specify and retain the destination URL during the SSO flow. Your portal can also set this value to any resource in the Domo instance when sending a SAML assertion. The RelayState is set as an object on the POST operation in plain text following the SAMLResponse.

**SAMLResponse=**PD94bWwgdmVyc2lvbj0iM … nA6QXV0aG5SZXF1ZXN0Pg%3D%3D&**RelayState=std**

Domo accepts both relative and absolute URLs, but will ignore any URL not within the Domo instance.

# Embedding Domo in a Custom Mobile App

<<under construction>>

1. User logs in mobile client
2. Mobile client requests session on IDP server
3. IDP server sets a cookie / token on the web view browser
4. DOMO sends SAML request to IDP sever
5. IDP validates the cookie / token from the browser
6. IDP sends SAML response
7. Mobile client will render embed card.

# Logging out of Domo

To ensure the browser drops Domo cookies, your portal can load the sign out URL from an iFrame:

https://<instanceName>.domo.com/auth/signoutandredirecttosigninpage

# Appendix A: Example SAML Response

This example is from an internal service at Domo named “Bourne.”

* Domo uses SAML Browser POST Profile
* The text after “SAMLResponse=” must be Base64 encoded
* The “RelayState=” text and its value are NOT Base64 encoded
* Replace the highlighted values, crypto, and control elements (like “InResponseTo=”) with details from your portal

SAMLResponse=<?xml version="1.0" encoding="UTF-8"?>

<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ID="R115792c19c27451fb8c06bf854e6fb0e" InResponseTo="\_e06aa075-b789-4e1e-b01d-b0cad9599a23" Version="2.0" IssueInstant="2017-10-24T15:35:29Z" Destination="https://bourne.dev.domo.com/auth/saml">

  <saml:Issuer/>

  <samlp:Status>

    <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>

  </samlp:Status>

  <saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ID="A1ec7453e2e6640d6b09ff77dd237a288" Version="2.0" IssueInstant="2017-10-24T15:35:29Z">

    <saml:Issuer/>

    <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">

      <ds:SignedInfo>

        <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>

        <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>

        <ds:Reference URI="#A1ec7453e2e6640d6b09ff77dd237a288">

          <ds:Transforms>

            <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>

            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>

          </ds:Transforms>

          <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>

          <ds:DigestValue>O+iqF2GO……..eDzqOnZxI=</ds:DigestValue>

        </ds:Reference>

      </ds:SignedInfo>

      <ds:SignatureValue>ztTrSjCLDyadayadayada…….yada+PIZug3IHQOXc=</ds:SignatureValue>

      <ds:KeyInfo>

        <ds:X509Data>

          <ds:X509Certificate>1UEBhMCdXMx…….0srsaP5Co3PmhJLZByrzZA2A==</ds:X509Certificate>

        </ds:X509Data>

      </ds:KeyInfo>

    </ds:Signature>

    <saml:Subject>

      <saml:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent">embed.user@domo.com</saml:NameID>

      <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">

        <saml:SubjectConfirmationData InResponseTo="\_e06aa075-b789-4e1e-b01d-b0cad9599a23" Recipient="https://bourne.dev.domo.com/auth/saml" NotOnOrAfter="2017-10-25T15:35:29Z"/>

      </saml:SubjectConfirmation>

    </saml:Subject>

    <saml:AuthnStatement AuthnInstant="2017-10-24T15:35:29Z" SessionIndex="A1ec7453e2e6640d6b09ff77dd237a288">

      <saml:AuthnContext>

        <saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport</saml:AuthnContextClassRef>

      </saml:AuthnContext>

    </saml:AuthnStatement>

    <saml:AttributeStatement>

      <saml:Attribute Name="name" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">

        <saml:AttributeValue xsi:type="xs:string">Embed User</saml:AttributeValue>

      </saml:Attribute>

      <saml:Attribute Name="email" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">

        <saml:AttributeValue xsi:type="xs:string">embed.user@domo.com</saml:AttributeValue>

      </saml:Attribute>

      <saml:Attribute Name="groups" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">

        <saml:AttributeValue xsi:type="xs:string">Group1</saml:AttributeValue>

        <saml:AttributeValue xsi:type="xs:string"> Group2</saml:AttributeValue>

      </saml:Attribute>

    </saml:AttributeStatement>

  </saml:Assertion>

</samlp:Response>

RelayState=/page/1234567

See <https://wiki.shibboleth.net/confluence/display/OpenSAML/Home> for Open Source developer resources for SAML 2.0.