SAML 2.0 Single-Sign-On
TorchLMS Enterprise v2.0
1.0 Introduction

1.1 Purpose

This document provides an overview for implementing SAML-based authentication with Torch LMS Enterprise. Security Assertion Markup Language (SAML) creates endpoints that give an organization's users a single URL to sign into, and then seamlessly access authorized applications without additional logins. SAML also provides an additional level of security.

1.2 Scope

This document outlines the integration of a SAML Service Provider (SP) with an Identity Provider (IdP). The Information exchanged between the SP and IdP includes:

- **Torch LMS metadata URL**: https://sso-[prod or uat].torchlms.com/saml/metadata
- **Entity ID**: Set as a Partner Id for identifying the source. This should be a unique url that represents your company. It is therefore recommended that it contain your domain name (e.g. https://[Your Company Domain]/idp).
- **SAML Version**: Required 2.0 or higher
- **Assertion Consumer URL**: https://sso-[prod or uat].torchlms.com/saml/SSO
- **Consumer Binding**: Transmission method is HTTP-POST
- **Name ID**: Unique identifier for sending user information (e.g. ID, Username, Employee ID, iNumber, etc.). By default this will represent the users TorchLMS username. If you would like to use the TorchLMS ID field for SSO you will have to tell our TorchLMS integration team to set that up for you.
- **Attributes**: Specified as the User Template in the Admin tool of Torch LMS
1.3 Overview

SAML is an XML-based open-standard format for exchanging authentication and authorization data between an IdP and an SP. Currently Torch LMS supports IdP initiated SAML SSO through HTTP-POST. It also supports Just-In-Time provisioning.

2.0 SAML (SP) Integration Overview

2.1 SSO Overview

Users can access Torch LMS through a custom integration using SAML 2.0. The following diagram illustrates the process workflow for the SAML SSO integration between Torch LMS and its customers:
Step 1:
- The user will first log in to the IdP (customer web portal)

Step 2:
- The user will then be able to enter Torch LMS by clicking a link or button on the customer web portal. At this point the user is not logged on to Torch LMS.

Step 3:
- The IdP retrieves a unique attribute (this will be the either the ID or the user’s username in Torch) from the user store and any additional attributes they want to be maintained in Torch LMS.

Step 4:
- The IdP F-SSO (Federation SSO) service will return an html form to the browser with a SAML response containing the authenticated assertion and the attributes as specified in your User Template. More on User Templates and Just-In-Time provisioning of users is found in Section 3. A sample SAML Response is included in the last section of this document.

- The service will then automatically post the HTML form to Torch LMS (Assertion Consumer Service) (note: it is digitally signed by customer with a private key).

Steps 5:
- Torch LMS will validate the signature and assertion (note: in order to do this we need your public key), update all the user attributes sent and redirect the user to the Torch LMS home page.
2.2 SSO Identity Providers (IdPs) Supported by TorchLMS

Torch LMS supports all major SAML IdPs. Some example IdPs include:

- Active Directory Federation Services (ADFS)
- Auth0
- Azure AD (Microsoft Azure Active Directory)
- Bitium
- Okta
- OneLogin
- Ping Identity
- Salesforce.com
- Any other provider that supports SSO systems adhering to SAML 2.0+ standards

2.3 Deployment Requirements from IdP to TorchLMS

- Ideally, users will be registered within the IdP before integration, but if Torch LMS receives a SAML response for a user that does not exist, the user will be created automatically with the specified attributes sent. Each user must have a unique id (e.g. ID, Username, Employee ID, iNumber, etc.) that becomes the ID of the user inside Torch LMS and is used to reference that user. Your Entity ID (described in section 2.1) is placed in the Issuer tag and tells us who is requested the SSO.

```xml
<saml2:Issuer>[Your Entity ID]</saml2:Issuer>
<saml2:Subject>
  <saml2:SubjectConfirmation ... >
    <saml2:SubjectConfirmationData Recipient="https://sso-[prod or uat].torchlms.com/saml/SSO" />
  </saml2:SubjectConfirmation>
</saml2:Subject>
```

- The IdP must support the name identifier format stated below:
  a. urn:oasis:names:tc:SAML:2.0:nameid-format:unspecified

- All SAML attributes should be represented using the name specified in the User Template discussed in Section 2.1 Step 4.
2.4 Information required by TorchLMS from the SAML IdP

- **Entity ID (required):** This should be a unique url that represents your company. It is therefore recommended that it contain your domain name (e.g. https://[Your Company Domain]/idp).

- **Public X.509 Certificate (required):** Torch will use this certificate to establish trust with the IdP. Torch will validate incoming SAML assertions from the IdP with this certificate.

- **Metadata (optional):** Generated by your IdP solution (if you are unable to do so, you can send the below information instead).

- **Single Logout Service (recommended):** This is where Torch LMS will send the user after a logout request to (https://sso-[prod or uat].torchlms.com/saml/SingleLogout)

- **Single Sign On (recommended):** The SSO endpoint where Torch LMS will send an authentication request.

2.5 Information provided by the SP (Torch LMS)

- **Entity ID (required):** A unique identifier for Torch LMS (https://webauth.torchlms.com/sp).

- **Assertion Consumer URL (required):** https://sso-[prod or uat].torchlms.com/saml/SSO

- **Torch LMS Metadata (optional):** This will contain all the information you need to setup your IdP to send SAML responses to Torch LMS. It can be retrieved at the following url: https://sso-[prod or uat].torchlms.com/saml/metadata.

- **Single Logout Service:** This option is required to implement single logout for Torch LMS (https://sso-[prod or uat].torchlms.com/saml/SingleLogout)
3.0 SAML Just-In-Time User Provisioning

3.1 Attributes sent to Torch LMS when creating or updating a user

- The name of each attribute can be customized with the SSO Parameter fields when setting up your User Template. This includes custom fields that your company has added and wants to keep up-to-date through JIT (Just-In-Time) provisioning. Section 3.2 shows examples of how you would specify your Attributes in your SAML Response.

- If you want to be able to create users you must, at least, specify the users sso_firstname, sso_lastname, sso_supervisor (if they do not have one this value should be set to the username of the person being created), and sso_org_1.
- The sso_username is what the user will use if they are allowed to do so manually. If it is not sent it will be set to the NameID sent in the SAML response.

- The username or the ID # can be updated at any time in the attribute fields of the SAML response, but the NameID field must be set to the old value in order to associate those changes with the user. If a NameID is sent to our system that does not currently exist in our system a new user will be created.

- A language attribute can be added if multiple languages are being assigned to users (sso_language). If only one language is being assigned to all users, this field does not need to be sent. Otherwise, use this field to indicate the user’s language (e.g. “en_US”, “es_MX”, “fr_CA”, etc.)

- The sso_phonenumber attribute can be in any format and is not required.

- The sso_timezone indicates what this users timezone should be. Accepted values are
  - “America/Los_Angeles”
  - “America/Denver”
  - “America/Chicago”
  - “America/New_York”.

- The sso_startdate is specified in the following format: “MM/dd/yyyy” and will default to the current date if not sent.

- The sso_supervisor specifies the username of this user’s supervisor. If the user does not have a supervisor, set this to the same value as the user’s username. If the supervisor has not been created, TorchLMS will create a placeholder user with name “Unknown Supervisor”. As soon as the supervisor signs in, all the correct information will get updated on the supervisor.

- The sso_roles attribute specifies any number of roles you would like to give this user. The “User” is added to all users automatically. The “Supervisor” role is also added automatically if the user is assigned as anyone’s supervisor via the sso_supervisor attribute. These roles do not need to be specified. Other Roles that you can include: “Instructor”, “Admin”, “Super Admin”, and custom roles that you create in the admin tool.
When specifying the organization group that a user belongs to you must send the full hierarchical path. The “Everyone” group is always added as the root to the other organization groups. Using the example given in Section 3.2 the user John Doe is a part of Department A (Everyone > West > Department A), therefore sso_org_1 and sso_org_2 are both specified as shown below. If you do not send this information, the user will be placed in the “Everyone” organization group.

For any Admins that administer specific organization groups you will want to specify additional attributes for each organization group that they administer. Using the example in Section 3.2, John Doe acts as the Administrator for Department A and the East Organization Groups.

Mission Statement is an example of a custom User Template field. Any custom User Template fields sent in a SAML Response will be set on the authenticating user.
3.2 Examples of different types of attributes that can be sent

```xml
<saml2:AttributeStatement>
  <saml2:Attribute Name="sso_username">
    <saml2:AttributeValue ... xsi:type="xs:string">johndoe</saml2:AttributeValue>
  </saml2:Attribute>
  <saml2:Attribute Name="sso_id">
    <saml2:AttributeValue ... xsi:type="xs:string">jdoe123</saml2:AttributeValue>
  </saml2:Attribute>
  <saml2:Attribute Name="sso_org_1">
    <saml2:AttributeValue ... xsi:type="xs:string">West</saml2:AttributeValue>
  </saml2:Attribute>
  <saml2:Attribute Name="sso_org_2">
    <saml2:AttributeValue ... xsi:type="xs:string">Department A</saml2:AttributeValue>
  </saml2:Attribute>
  <saml2:Attribute Name="sso_phonenumber">
    <saml2:AttributeValue ... xsi:type="xs:string">123-456-7890</saml2:AttributeValue>
  </saml2:Attribute>
  <saml2:Attribute Name="sso_firstname">
    <saml2:AttributeValue ... xsi:type="xs:string">John</saml2:AttributeValue>
  </saml2:Attribute>
  <saml2:Attribute Name="sso_lastname">
    <saml2:AttributeValue ... xsi:type="xs:string">Doe</saml2:AttributeValue>
  </saml2:Attribute>
  <saml2:Attribute Name="sso_roles">
    <saml2:AttributeValue ... xsi:type="xs:string">Instructor</saml2:AttributeValue>
    <saml2:AttributeValue ... xsi:type="xs:string">Admin</saml2:AttributeValue>
  </saml2:Attribute>
  <saml2:Attribute Name="sso_admin_org_1_1">
    <saml2:AttributeValue ... xsi:type="xs:string">West</saml2:AttributeValue>
  </saml2:Attribute>
  <saml2:Attribute Name="sso_admin_org_1_2">
    <saml2:AttributeValue ... xsi:type="xs:string">Department A</saml2:AttributeValue>
  </saml2:Attribute>
  <saml2:Attribute Name="sso_admin_org_2_1">
    <saml2:AttributeValue ... xsi:type="xs:string">East</saml2:AttributeValue>
  </saml2:Attribute>
  <saml2:Attribute Name="sso_timezone">
    <saml2:AttributeValue ... xsi:type="xs:string">America/Los_Angeles</saml2:AttributeValue>
  </saml2:Attribute>
  <saml2:Attribute Name="sso_missionstatement">
    <saml2:AttributeValue ... xsi:type="xs:string">Be awesome</saml2:AttributeValue>
  </saml2:Attribute>
  <saml2:Attribute Name="sso_password">
    <saml2:AttributeValue ... xsi:type="xs:string">password123</saml2:AttributeValue>
  </saml2:Attribute>
</saml2:AttributeStatement>
```
3.3 Sample SAML Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<saml2p:Response xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol"
Destination="http://abccorp.torchlms.com/saml/SSO" ID="a6f3bb21-c5c6-466f-9bd2-b43e49181023" IssueInstant="2015-10-08T13:08:17.841Z" Version="2.0">
  <saml2:Issuer xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">
    [Your Entity ID]
  </saml2:Issuer>
  <saml2p:Status>
  </saml2p:Status>
  <saml2:Assertion xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:xs="http://www.w3.org/2001/XMLSchema" ID="77c69ac3-a513-4328-94c7-4e58688017da"
IssueInstant="2015-10-08T13:08:17.810Z" Version="2.0">
    <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="#77c69ac3-a513-4328-94c7-4e58688017da">
          <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#">
              <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#"
PrefixList="xs" />
            </ds:Transform>
          </ds:Transforms>
        </ds:Reference>
        <ds:SignatureValue>UsJ+ISqTZkExMTNZ1ygjrIFcPYKCYwi9HuMihB0FW9hhpv7zCFRvXXhd138isE2vlacPj</ds:SignatureValue>
      </ds:SignedInfo>
      <ds:SignatureValue>
      </ds:SignatureValue>
    </ds:Signature>
  </saml2:Assertion>
</saml2p:Response>
```
</saml2:AuthnStatement>
<saml2:AttributeStatement>...See section 3.2</saml2:AttributeStatement>
</saml2:Assertion>
</saml2p:Response>