Kaltura MediaSpace™
SSO Integration Guide

Version: 4.5

Kaltura
open source video

Kaltura Business Headquarters

5 Union Square West, Suite 602, New York, NY, 10003, USA

Tel.: +1 800 871 5224

Contents

Preface

This preface contains the following topics:

About this Guide

This document describes how to implement single sign-on (SSO) in Kaltura MediaSpace™.

NOTE: Please refer to the official and latest product release notes for last-minute updates. Technical support may be obtained directly from: Kaltura Support.

Contact Us:

Please send your documentation-related comments and feedback or report mistakes to knowledge@kaltura.com.

We are committed to improving our documentation and your feedback is important to us.

Audience

This document is intended for Kaltura partners, community members, and customers who wish to understand and deploy SSO authentication and authorization in MediaSpace.

To understand this document, you need to be familiar with authentication and authorization terminology.

Document Conventions

Kaltura uses the following admonitions:

- Note
- Workflow

NOTE: Identifies important information that contains helpful suggestions.

Workflow: Provides workflow information.

1. Step 1
2. Step 2

Related Documentation

In addition to this guide, the following product documentation is available:

- Kaltura MediaSpace
- Kaltura MediaSpace Setup Guide
- Kaltura MediaSpace: Introduction to Authentication and Authorization Solutions

NOTE: Please remember to review all product release notes for known issues and limitations.

Understanding the MediaSpace Single Sign-on Authentication and Authorization Gateway

Understanding MediaSpace Authentication and Authorization

MediaSpace includes a low-level authentication mechanism. When a user attempts to access MediaSpace, MediaSpace checks whether the user is logged in. If the user is not logged in or the login is expired, the user is redirected to another URL.

As part of the authentication mechanism, MediaSpace determines the user's *application role* following a successful login. The application role determines the MediaSpace actions that the user is authorized to do. To learn more about the MediaSpace application role, refer to Understanding Application Roles in the Kaltura MediaSpace Setup Guide.

Understanding the SSO Gateway

In the SSO Gateway, an expired login causes the user to be redirected to the SSO Gateway's login page.

On the login page, the user logs into *your* user management system using your methodologies (you are responsible for writing the login code).

When a user logs in successfully, the login page generates a unique session key using a *secret* shared between the login page and MediaSpace (which you define for MediaSpace on the Configuration Management panel of the Kaltura MediaSpace Administration Area. Refer to Configuring SSO Gateway Authentication and Authorization in the Kaltura MediaSpace Setup Guide). The session key also securely encapsulates user information that is passed to MediaSpace.

After successfully logging into your system, the user is redirected to the MediaSpace authentication URL. The session key is passed as a URL parameter.

The MediaSpace authentication URL uses the secret to validate the session key. If the session key is valid, the MediaSpace authentication URL logs the user into MediaSpace. If the session key is not valid, the MediaSpace authentication URL redirects the user back to your login page.

NOTE: A sample login page (and related files) is available from your project manager upon request.

Constructing the MediaSpace Authentication URL

In the SSO Gateway, the MediaSpace authentication URL logs in a user based on the customer user management's session key. The following is the URL structure:

http://*mediaspace_domain*/*mediaspace_path*/user/authenticate/sessionKey/{secure hash string}

The URL consists of the following elements:

- **mediaspace_domain** – the host name of the server that hosts your MediaSpace. Replace *mediaspace_domain* with the actual value in your environment.
- **mediaspace_path** – the URI of MediaSpace, when MediaSpace is installed under a subfolder. Replace *mediaspace_path* with the actual value in your environment.
- **user** – the controller that handles all user-related actions in MediaSpace. This element is hardcoded.
- **authenticate** – the action within the user controller that handles the authentication. This element is hardcoded.
- **sessionKey** – the parameter name in the URI for the secure hash string. This element is hardcoded.
- **secure hash string** – the token that your login page creates and then passes to MediaSpace in the redirect URL**.**

Generating and Validating a Secure Hash String

A secure hash string is generated in your login page and is passed to MediaSpace in the redirect URL.

As part of the MediaSpace authentication process, MediaSpace validates the secure hash string received from the login page.

Sample Code for Generating a Secure Hash String

The following pseudo code demonstrates how to generate a secure hash string.

```
info = “userId;userRole;extraUserInfo;expiry;random”
salt = {MEDIA_SPACE_LOGIN_SECRET}
signature = sha1(salt+info)
stringToHash = signature+”|”+info
hashedString = base64_encode(stringToHash)
```

- `userId` – the ID of the user in your system as you use want it to be identified in Kaltura.
- `userRole` – the application role that you assign the user in MediaSpace.
- `extraUserInfo` – a key-value string in which pairing is specified using colons (:) and pairs are separated by commas (,). For example:

  `extraUserInfo = "display_name:Gonen,age:30,hobby:surfing"`
- `expiry` – a Unix-timestamp after which the hashed string is no longer valid.
- `random` – a randomly generated number from 0-32000.
- `MEDIA_SPACE_LOGIN_SECRET` – the secret string shared by MediaSpace and the login page.

Sample Code for Validating a Secure Hash String

The following pseudo code demonstrates how the secure hash string passed in the sessionKey URL parameter is validated.

```
decodedString = base64_decode(hashedString)
separate decoded string by |, first part is signature second is info
salt = {MEDIA_SPACE_LOGIN_SECRET}
validateSignature = sha1(salt, info)
IF validateSignature = signature AND expiry < current unix time
    hash is valid
ELSE
    hash is not valid
```

The validation code mirrors the token generation actions in reverse order:

1. The secure hash string is decoded from its base64 encoding.
2. The decoded string is separated into two parts: the signature and the information.
3. The code creates a signature using the information provided and the SSO secret, which is defined in the MediaSpace configuration.

If the signature created by the code matches the signature provided by the redirect URL and the expiration date did not pass, the secure hash string is valid and the user is successfully authenticated.

Configuring the SSO Gateway in MediaSpace

To learn how to configure the SSO Gateway for authentication and authorization in MediaSpace, refer to Configuring SSO Gateway Authentication and Authorization in the Kaltura MediaSpace Setup Guide.