Kaltura Application Framework Integration Guide

Version: 5.x

Kaltura
open source video

Kaltura Business Headquarters

250 Park Avenue South, New York, NY. 10003, USA

Tel.: +1 800 871 5224

Contents

Preface

This preface contains the following topics:

About this Guide

This guide introduces the Kaltura Application Framework (KAF) and provides information about integrating it into other applications to quickly and effectively leverage Kaltura's rich media platform according to the workflows you need to implement in your own application.

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 want to understand how to integrate the Kaltura Application Framework and build applications and extensions that leverge the available functionality and use cases.

To understand this document, you need to be familiar with Kaltura and the Kaltura APIs.

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

\

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

SECTION 1

Understanding the Kaltura Application Framework

The Kaltura Application Framework (KAF) is an extensible, feature rich, UI based configurable framework that streamlines the integration of Kaltura's rich media capabilities into different publishing applications.

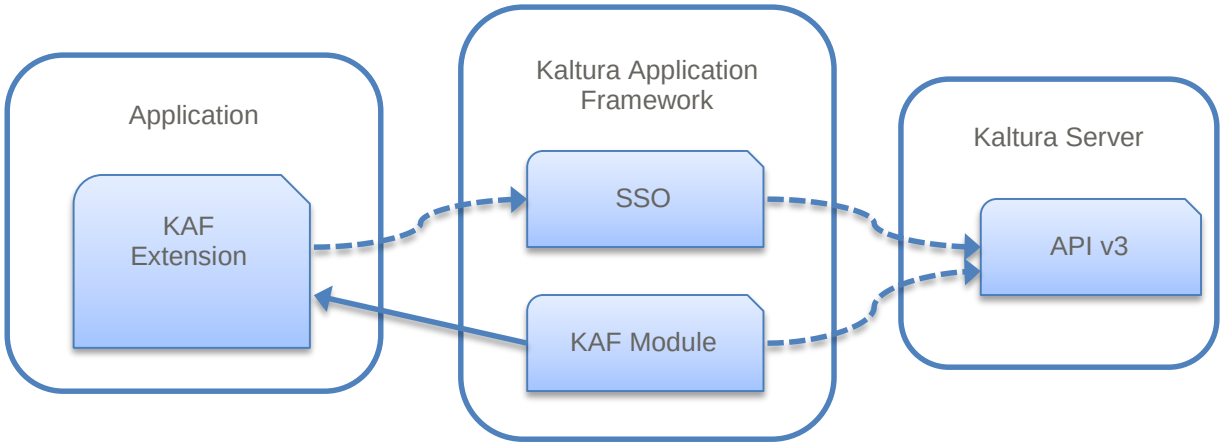
The framework is constructed of modules that provide a packaged workflow and functionalities that can be easily embedded in another application as an iFrame, instead of integrating directly with the Kaltura APIs. The framework can decrease the integration time with your application dramatically and allow you to always get up to date new functionality by decoupling the added features from the integration itself.

The embedded iFrames are all based on a responsive design to ensure that the integrated pages are displayed properly on any given area.

High Level Component Diagram

The following diagram illustrates a high level description of the different components in a typical application integration and how they interact.

The extension, within an application, loads an iFrame of a specific KAF module through a single-sign-on (SSO) authentication.



The KAF module uses Kaltura API's to display, add, or change content depending on the module available and configured functionality. The display is rendered in HTML that can be directly used by the user in the integrated application.

SECTION 2

Integrating Kaltura's Media Capabilities Into Different Publishing Applications

This section describes the following topics:

- Authentication and Authorization
- Framework Modules

Authentication and Authorization

To load a KAF module in an iFrame, an authentication process must occur so that the KAF module can identify the user and determine the user's privileges.

KAF provides two methods of authentication: KS-based SSO and LTI.

KS-Based Single Sign-On Implementation

In KS-Based SSO, the KAF end-point (URL of a specific module) must include a security token.

The security token expected by KAF is a KS (Kaltura Session) that includes, in its list of privileges, relevant user information, session information and other security-oriented or functionality-related privileges.

A KS can be constructed using any of the Kaltura API Client libraries, with the internal method that is called *generateSession* (or a similarly named function).

The following is an example for generating a session in PHP using the client library:

```
$client->generateSessionV2(
   $adminSecret,
   $userId,
   $sessionType,
   $partnerId,
   $expiration,
   $privileges
);
```

The following table explains the parameters expected by the *generateSession* method.

Admin Secret	The admin secret of the Kaltura account. Can be retrieved from the KMC, under "Settings" tab, in "Integration Settings".
User ID	The unique ID of the user that is being presented with the KAF module.
Session Type	Type of Kaltura Session. For KAF integration, the Session Type must be USER session.
Partner Id	The Kaltura account ID. Can be retrieved from the KMC, under "Settings" tab, in "Integration Settings".

Expiration	Expiration length of the KS **in seconds.** For KAF integration, the Expiration values hould be up to 60 seconds.
Privileges	Comma-separated list of privileges. Some privileges are of type key-value pair, in which case they are represented as key:value. For the maximum security, the required privilege is "actionslimit:-1" to make the KS useless for any API calls.

The following is a practical example for generating a KS for loading a KAF module:

```
// prepare privileges
$privileges= array();
$privileges[] = "actionslimit:-1";
$privileges[] = "firstName:John";
$privileges[] = "lastName:Doe";
$privileges[] = "role:viewerOnly";

// transform array privileges to string
$privilegesStr = implode(",", $privileges);

// prepare additional parameters
$adminSecret = "-the-string-you-copied-from-kmc-";
$userId = "john.doe";
$partnerId = 12345;

// generate the KS using all the above parameters
$ks = $client->generateSessionV2(
   $adminSecret,
   $userId,
   KalturaSessionType::USER,
   $partnerId,
   20,
   $privilegesStr
);

// build iFrame URL with KS
$iframeUrl = 'https://url.to.kaf.com/hosted/index/my-media/ks/' . $ks;

// render iFrame to page
echo '<iframe src="' . $iframeUrl . '"></iframe>';
```

NOTE: You can see the list of required and optional privileges, as well as the available KAF modules using the "kaftestme" module. Contact your Kaltura account manager to gain access to the "kaftestme" module.

Roles and Permissions for KS-Based SSO

KAF has two types of roles:

- Applicative role – a role that determines the list of allowed actions that the user is allowed to perform in the KAF application.
- Contextual role – a role that determines a user's capabilities within a specific given context.

When loading a KAF module, the KS must specify the user's applicative role.

Depending on the KAF module loaded, the KS might be required to also specify the user's contextual role in the loaded context.

In the example below:

- The applicative role of the user is "adminRole" . The adminRole user can upload content as well as publish content in different contexts (galleries).
- The contextual role of the user is "manager".. In the loaded context, the user should be considered a manager of that gallery, and is given some capabilities that are only allowed for this role.

NOTE: Contextual role values are the available constants of KalturaCategoryUserPermissionLevel

```
$privileges= array();
$privileges[] = "actionslimit:-1";
$privileges[] = "firstName:John";
$privileges[] = "lastName:Doe";
$privileges[] = "role:adminRole";
$privileges[] = "userContextualRole:0";

...

// build iFrame URL with KS
$iframeUrl = 'https://url.to.kaf.com/hosted/index/course-gallery/ks/' . $ks;

...
```

NOTE: The list of *applicative roles* can be seen in the "kaftestme" module.

LTI-Based Authentication and Authorization

If your application supports LTI (administrator capabilities for configuring tools, or internal code-based API for rendering LTI tools) you can choose to integrate your system to KAF based on LTI.

NOTE: At this time, the "kaftestme" module does not show how LTI integration should be constructed.

The following lists the mandatory parameters expected in an LTI launch, including comments on relevancy for a specific KAF module. The entire set of required LTI parameters is not listed. The table contains the specific parameters required for KAF integration.

Parameter	Use/Scope	Comments
contextId		Currently required for all KAF modules, even if some modules are not loaded in a specific context (course).
context_title	Used in the "Gallery Module" to display the name of the gallery,	Recommended.
user_id		Should contain a unique identifier of the user, preferably one that the user, or you as an organization, would recognize An alternative attribute name can be configured in the "hosted" module.

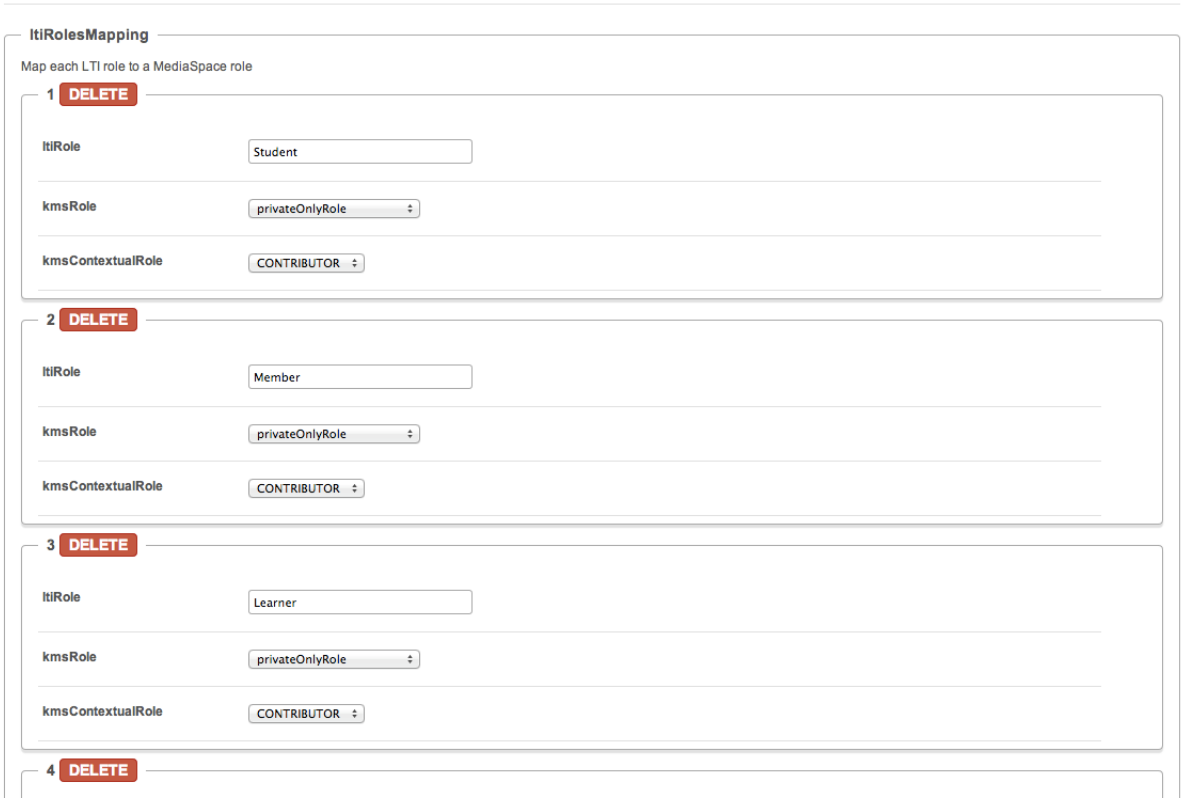
Parameter	Use/Scope	Comments
lis_outcome_service_url	Used by the "browse and embed" module to return data of the selected content to this URL.	An alternative attribute name can be configured in the "hosted" module of your KAF instance. Refer to content-extension for the structure of the returned data. You can also use the "kaftestme" module to see the return data structure. There is no difference between KS-based and LTI-based SSO.
lis_person_contact_email_primary		
lis_person_name_family		
lis_person_name_given		
roles	See Roles and Permissions for LTI-Based SSO.	Should contain LIS roles as mentioned in LTI specification appendix.
launch_presentation_locale		

Additionally, you can choose to pass custom parameters in the LTI launch, if you have your own dedicated KAF module to use those parameters.

Roles and Permissions for LTI-Based SSO

Since the roles passed to the KAF application in LTI-based authentication are LIS roles, and not the expected KAF roles, KAF provides the ability to map LIS roles to KAF roles.

The mapping may be done in the configuration of the "hosted" module.



For each LIS role you can configure the applicative role in KAF that the user will be assigned.

For each LIS role you can also configure a contextual role that the user will be assigned.

KAF includes a default mapping that can be modified and/or extended to map additional LIS roles.

NOTE: The role that the user is granted is the highest one of all matches.

Framework Modules

This section describes the following KAF modules:

- Gallery Module
- My Media Module

Gallery Module

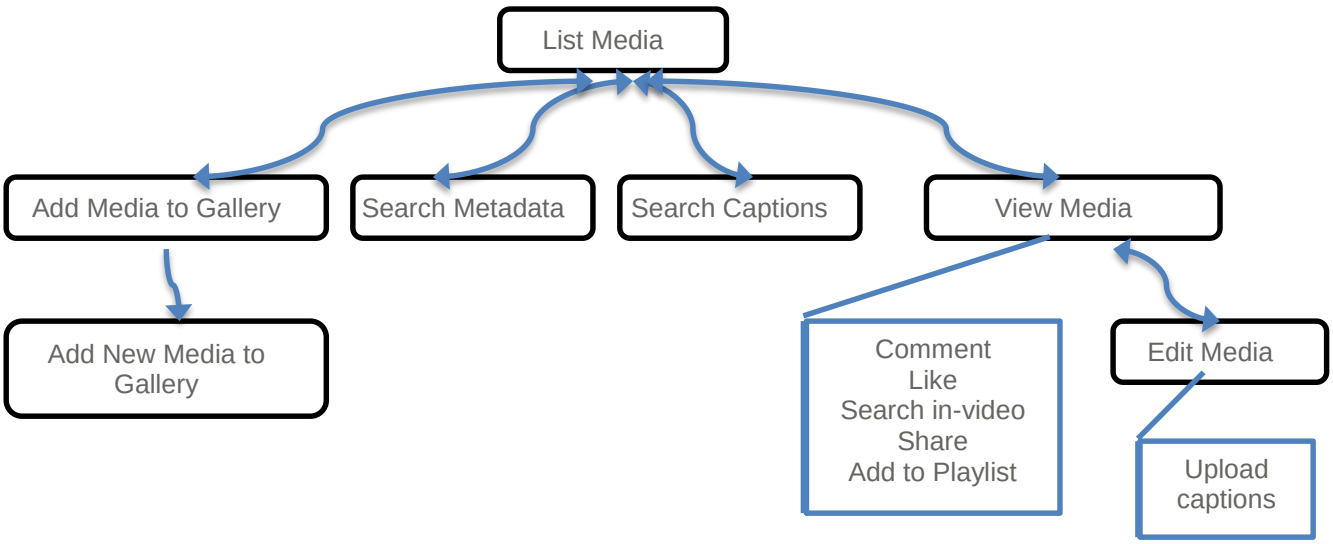
The Gallery Module is used to create, display and contribute to a dedicated media gallery that can be associated with a group in your application. For example, a media gallery for a specific forum, a course, a community, or other context.

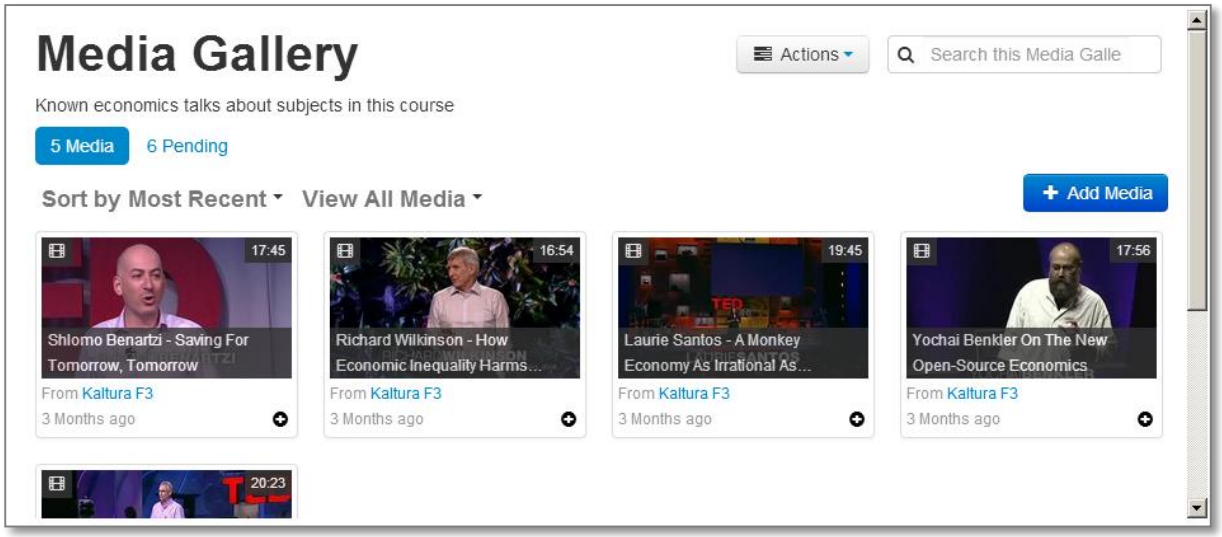
The Gallery Module includes the following functionality:

- Display of media that is published in the gallery.
- Viewing and engaging with the published media. According to the configuration, a user may: Play the media, search in-video, comment, like, add to playlist, and other actions that are configured.

- Search metadata and transcription / closed caption files
- Contribute to a gallery (according to user's role)
- Upload new content and publish to the gallery (according to user's role)
- Moderate content in gallery (optional)

The following diagram describes the user flow between the different available pages and functionality within this module:





My Media Module

The My Media Module displays the personal media library of a authenticated user.

Specific functionality in this module includes:

- Displaying media that the user owns (contributed by the logged user).
- Viewing / Editing media
- Search on metadata and transcription / closed caption files

- Uploading new content and publish to the gallery (according to user's role)
- Bulk publishing to a specific gallery or playlist

The following diagram describes the user flow between the different available pages and functionality within the My Media Module: