# Understanding the Kaltura Application Framework (KAF)

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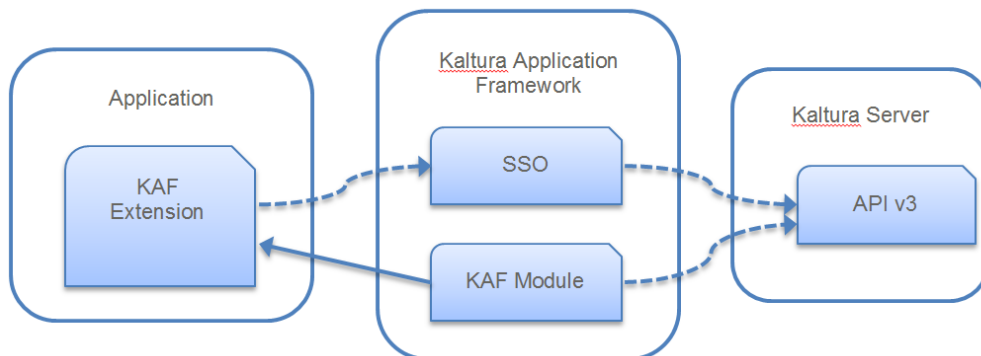
The KAF module uses Kaltura API's to display, add, or change content depending on the module available and configured functionality. The display is rendered in HTML that can be directly used by the user in the integrated application.

Integrating Kaltura's Media Capabilities Into Different Publishing Applications

This section describes the following topics:

- Authentication and Authorization

-

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:

```php
$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 should 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:

```php
// prepare privileges

$privileges= array();

$privileges[] = "actionslimit:-1";

$privileges[] = "firstName:John";

$privileges[] = "lastName:Doe";

$privileges[] = "role:viewerRole";


// 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=""></iframe>';
```

The list of KAF privileges, that are required, depends on the KAF widget you want to use.

To understand the privileges that are required or optional, visit http://kaf-integration-demo.kaf.kaltura.com/kaftestme/index/index . You can choose between the available KAF widgets (End Point) and see a detailed list of required and optional privileges per widget. After submitting your information you can simulate the KAF widget's behavior according to the values passed.

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.

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;



...
```
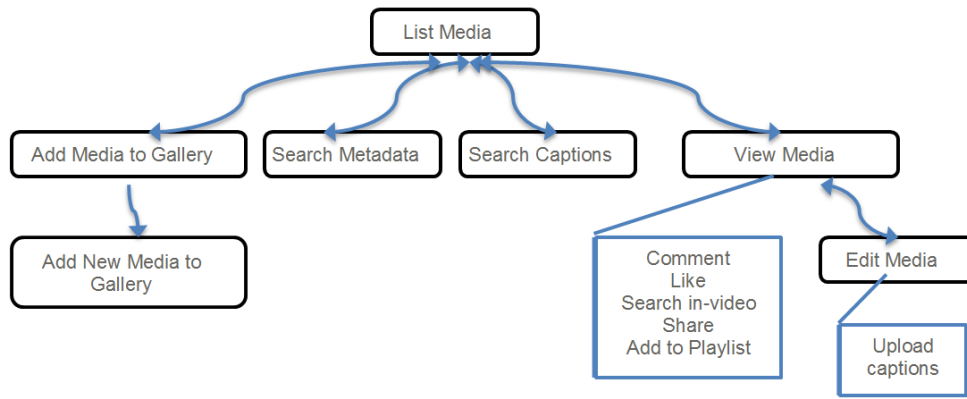
Gallery Module

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:
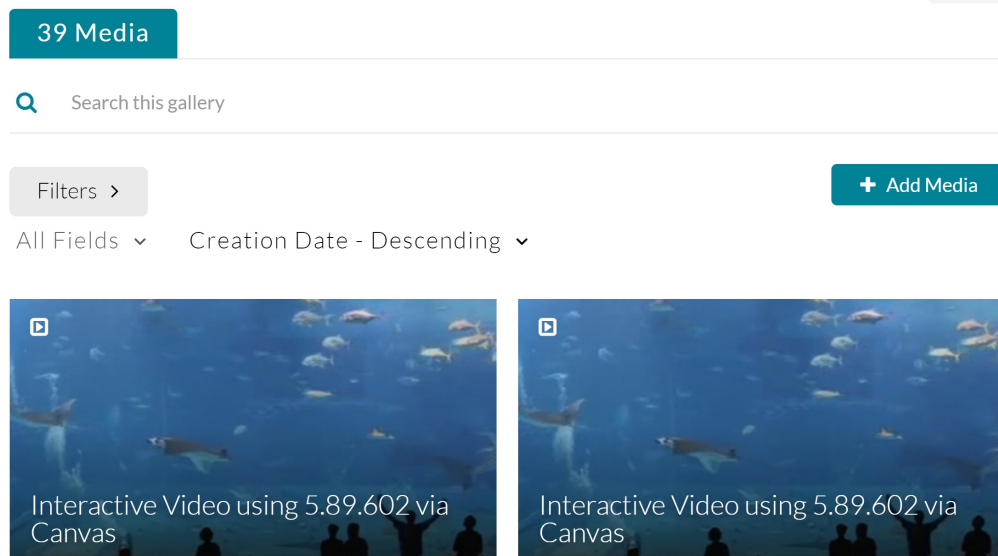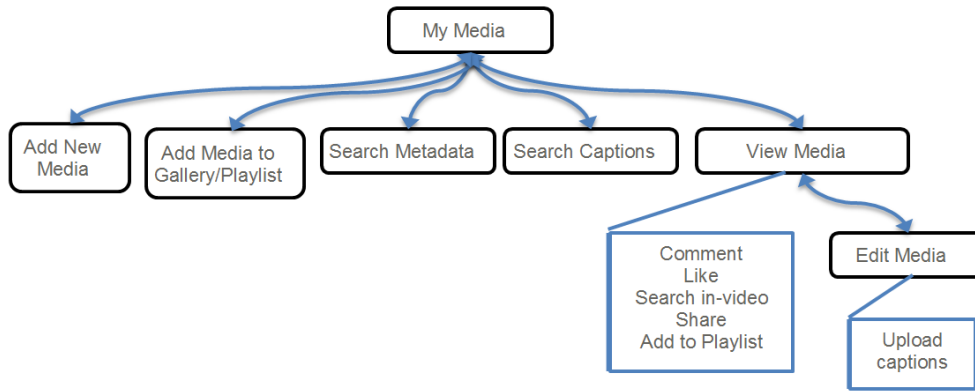
My Media Module

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
- Upload 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:

My Media

 My Dashboard   My Schedule

Search My Media

Filters > | All Fields ∨ | Creation Date - Descending ∨   ACTIONS ∨

☐

☐
RAPT-193 interactive via KAF by Adi - Canvas
RAPT-193 interactive via KAF by Adi - Canvas

Published

liat.noy@kaltura.com

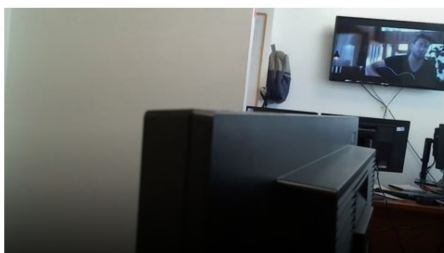Updated April 15th, 2019

💬 0

☐
Sweet Harmony - The Beloved - Quiz
Sweet Harmony

Published

liat.noy@kaltura.com

Updated September 17th, 2017

💬 0

☐
Kaltura Classroom recording

Published

liat.noy@kaltura.com

Updated September 14th, 2017

💬 0

[template("cat-subscribe")]