

# Webhooksiteregistration module

Last Modified on 12/23/2025 4:22 pm IST

 This article is designated for administrators.

## About

The **Webhooksiteregistration module** lets you create users from external sources (like Marketo or Facebook) using a webhook. The app gets a call from the external service via the webhook and creates the new user based on the data provided.



This module is for the Video Portal only and requires the [Theming module](#) to be enabled.

## Configure

1. Go to your Configuration Management console, and click the **Webhooksiteregistration** module. You can also navigate directly using a link: [https://{your\\_site\\_URL}/admin/config/tab/webhooksiteregistration](https://{your_site_URL}/admin/config/tab/webhooksiteregistration).

<b>Modules /</b>
<b>siteRegistration</b>
Basesiteregistration
Inviteusers
Siteregistration
<a href="#">Webhooksiteregistration</a>

The 'Webhooksiteregistration' page displays.

## Configuration Management

**Server Tools**

- Backup Configuration
- Player replacement tool

**Global**

- Application
- AddNew
- Auth
- Categories
- Channels
- Client
- Debug
- EmailService
- Email Templates
- Gallery
- Header
- Languages
- Login
- MediaCollaboration
- Metadata
- Moderation
- My-media
- Navigation
- Player
- PlaylistPage
- Recaptcha
- Search
- Security
- SharedRepositories
- Widgets
- TimeAndDate

### WebhooksiteRegistration

**Module Info**

**Description**  
The webhooksiteRegistration module provides the ability to create users to the application from external sources (e.g. marketing, facebook) via a webhook. The application receives a call from the external service to a webhook (action), and creates the new user according to the data provided by that service.

**Note**  
KMS only Module. Depends on the [Theming](#) module.

enabled	<input type="button" value="Yes"/>	Enable the WebhooksiteRegistration module.
registrationKey	<input type="text" value=""/>	<a href="#">Generate key</a> key to validate request.
printToLog	<input type="button" value="No"/>	Print all the request information to log
getRegistrationKeyFromHeader	<input type="button" value="No"/>	The key need to be sent in authorization header from type Bearer Token
sendErrorResponseAsJson	<input type="button" value="No"/>	Send error in json format
allowGetRequestStatusCheck	<input type="button" value="No"/>	This is used, to allow GET request to check the status of the endpoint, it will return json with status true
postContentType	<input type="button" value="Form Data"/>	Select form data for key value pairs or Json to send in raw format
allowedAddresses	<input type="text" value="192.58.252.1-192.58.252.25"/> <input type="button" value="X"/>	provide IPs or ranges to allow access to the webhook registration action. if none configured, access is allowed to all IPs. Examples: 192.168.1.1 192.168.1.1/24. Requires module iprestriction to be enabled

[+ Add "allowedAddresses"](#)

## 2. Configure the following:

### General configurations & request's header

**enabled** - Set to 'Yes' to enable the module.

**registrationKey** - Click **Generate Key** to validate request. If a key already exists, leave the existing one.

**printToLog** - This is set to 'No' by default and is recommended to remain unchanged.

**getRegistrationKeyFromHeader** - Leave as 'No' unless a Kaltura representative tells you otherwise. If set to 'Yes', an additional section displays:

**keyHeaderInfo**

**authorizationHeaderType**

Bearer

Bearer

API Token

Specify which authorization method is used: API Key (more common) or Bearer token.

If **Bearer**, continue to **sendErrorResponseAsJson** field.

If **API Token**, additional fields display:

keyHeaderInfo
Collapse

authorizationHeaderType
API Token

apiTokenKeyName

getApiTokenFrom
Header
Header
Query Params

- **apiTokenKeyName** - give the name of the header field that holds the key.
- **getApitokenFrom** - Header or Query params ('header' is more common)

**sendErrorResponseAsJson** - Set to 'No' unless you want to get all responses in JSON format.

**allowGetRequestStatusCheck** - Set to 'No' unless you require validate endpoint using a GET request.

## Request's body

**postContentType** - Select **Form Data** for key value pairs or **JSON** to send in raw format. If you choose JSON, additional fields display:

**postContentJsonKey** - Enter name of object's wrapping key.

**postContentJsonRegInfoKeys** - Fill in the key name that holds user registration info (separated with a comma), for example, `firstName, lastName, email, company, jobTitle`

**allowedAddresses** - Leave empty. (This setting allows you to specify IP addresses or ranges that are permitted to access the webhook registration action. If no addresses are configured, access is open to all IPs by default. For example, you can enter a specific IP like `192.168.1.1` or a range like `192.168.1.1/24`. Note: The **iprestriction module** must be enabled for this feature to work.)

**defaultRole** - Choose a role from the following options:

- Viewer role
- privateOnly role
- admin Role
- unmoderatedAdmin Role

**defaultGroupId** - Add the group ID that will include the Webhook users / default group for newly created users (without an invite code).



Avoid using spaces in the group ID, as this will cause errors.

- If Kaltura event → “ep\_attendees\_xxxxxx” [default, do not change].
- If non-Kaltura event→ Create a new group called “Webhook” and add the ID of this group in the defaultGroupId field.

## Fields and languages mapping

Configure the mappings for user fields and the language parameter that KMS will ingest and process.



All items configured here must be included in the payload of the Webhook call to the Video Portal, allowing it to capture the information and apply the mapping logic.

### mandatoryFieldsMapping

Map the webhook fields to the mandatory Kaltura user object keys.

**emailWebhookKey** - Enter the webhook field name for `email`

**firstNameWebhookKey** - Enter the webhook field name for `firstName`

**lastNameWebhookKey** - Enter the webhook field name for `lastName`

### fieldsMapping

Configure the mappings for the user fields that Kaltura video portal will ingest and handle.

Click **+Add "fieldsMapping"** to display an additional section:

fieldsMapping

Map the webhook fields to Kaltura registration form and user object keys.

DELETE

webhookKey	<input type="text"/>	Enter the webhook Field name
userObjectKey	<input type="text"/>	Enter the user object field key. Leave empty to not map to a user object.
registrationFormKey	<input type="text"/>	Enter a registration form ID. Leave empty to use the webhook field name in the registration info.

Complete the following fields:

**webhookKey** - Enter the webhook field name.

**useObjectKey** - Enter the user object field key. Leave empty (Kaltura default field).

**registrationFormKey** - Enter a registration form ID (industry, companySize, interests, etc.). Leave empty to use the webhook field name in the registration info.

## languageMapping

Click **+Add "languageMapping"** to configure the field on which KMS will map users to the correct language and group.



Ensure that the language field from the webhook is mapped to the Kaltura language field in the user object. This enables KMS to load in the correct locale when the user accesses it via the set password link. If the specified language code isn't configured in KMS (under Application > Language Configuration), a fallback language from the configured options will be used.

Complete the following fields:

**externalLanguageCode** - Enter the language code from the webhook form.

**formkmsLanguage** - Choose a language from the drop-down list.

**groupId** - (Optional) Enter the group ID to add users to.

fieldsMapping

Map the webhook fields to Kaltura registration form and user object keys.

DELETE

webhookKey

company

userObjectKey

registrationFormKey

company

Must configure all three fields as 'company' is both a user object and a custom RegForm field

DELETE

webhookKey

jobTitle

userObjectKey

title

registrationFormKey

Must configure all three fields as 'title' is both a user object and a custom RegForm field

DELETE

webhookKey

industry

userObjectKey

registrationFormKey

industry

'Industry' is exclusively a custom RegForm field

DELETE

webhookKey

companySize

userObjectKey

companySize

registrationFormKey

Must configure registrationFormKey since 'companySize' is exclusively a custom RegForm field

## Screens & emails

Configure 'Registration Complete' screen and the emails for your Webhook registration flow:

**completeTitle** - The default is set to *Password has been set.*

**completeText** - The default is set to *Thank you for setting your password.*

**completeSubText** - The default is set to *Sign in to the site using your email and password.*

**completeButtonText** - The default is set to */user/login*

**completeButtonUrl** - The default is set to *Password has been set.*

**sendEmail** - The default is set to 'Yes'. If you set it to no, skip to the **webhookConfirmedEmailTemplate**.

**webhookEmailType** - Choose to send an email or a calendar invitation post registration as the invitation email to the user. Choose from the following options:

- Send email only
- Send both email and calendar invite
- Send calendar invite only

Once selected, make sure to choose the email template to use from the fields below. To use **calendar invitation**, the **loginLogFrom** field in the [basesiteregistration module](#) must be filled in.

**webhookSetPassEmailTemplate** - Choose a template for the initial email asking the user to setup their password for the application.

**iCalEmailTemplate** - Choose an email template to use in the calendar invitation. Time and date will be taken automatically from the [basesiteregistration module loginLogFrom](#) field (Click the link to navigate to the [emailTemplates module](#) to configure email templates).

**webhookConfirmedEmailTemplate** - Choose a template for post confirmation email letting the users know that their account is all confirmed and ready to login to the application.

To setup a Webhook Registration flow where no emails are involved, you must have Magic Link as the attendees method of authentication. For more details, please talk to your Kaltura representative.

The following optional fields are used only after the event ends.



Please note the post-event email fields

**webhookSetPassEmailTemplatePostEvent** and **webhookConfirmedEmailTemplatePostEvent** appear only after an event end time is defined in the **loginLogTo** field in the [BaseSiteRegistration module](#).

webhookSetPassEmailTemplatePostEvent

Choose a template ▼

Choose a template for the initial email asking the user to setup their password for the application - post event ([Configure email templates](#)).

webhookConfirmedEmailTemplatePostEvent

Choose a template ▼

Choose a template for post confirmation email letting the users know that their account is all confirmed and ready to login to the application - post event ([Configure email templates](#)).

Save

**webhookSetPassEmailTemplatePostEvent** - Choose a template for the initial email asking the user to set up their password for the application – post event.

**webhookConfirmedEmailTemplatePostEvent** - Choose a template for post confirmation email letting the users know that their account is all confirmed and ready to log in to the application – post event.

3. When you have finished configuring the desired settings, click **Save**.

---