

Webhookregistration module

Last Modified on 11/08/2024 7:20 pm IST

 This article is designated for administrators.

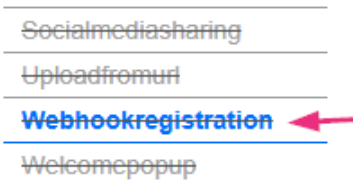
About

The Webhookregistration module is set up to enable external registration providers to create new users in the video portal, grant them access to specific entries or channels, and send a confirmation email to the registrant. This email includes a magic link, allowing direct access to the entry, channel, or homepage.

Configure

1. Go to your Configuration Management page and click the **Webhookregistration** module. You can also navigate directly using a link:

https://{your_site_url}/admin/config/tab/webhookregistration.



The Webhookregistration page displays.

MANAGE CONFIGURATION	MANAGE USERS	MANAGE GROUPS	KNOWLEDGE BASE	CLEAR THE CACHE	GO TO SITE
----------------------	--------------	---------------	----------------	-----------------	------------

Configuration Management

Server Tools

- Backup Configuration
- Player replacement tool

Global

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

Webhookregistration

Module Info

description	The module allows external registration providers to create new users in KMS, grant them access to an entry or a channel and then send a confirmation email to the registrant email address with magic link to access the entry/channel/home page	
version	1.0.2	

enabled Enable the Webhookregistration module.

createPreRegisteredUser When enabled, a user will be created immediately when the webhook is being sent. The user attendanceInfo will be with status 'Pre-registered' and the user status will be Blocked.

registrationKey key to validate request.

userPool

cognito user pools		
userPoolId	<input type="text"/>	Cognito user pool id
clientId	<input type="text"/>	Cognito user pool client id
region	<input type="text"/>	region used in the user pool

2. Configure the following:

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

createPreRegisteredUser - When set to 'Yes,' a user is created as soon as the webhook is sent. The user's attendanceInfo will show a status of "Pre-registered," and the user's status will be set to "Blocked."

registrationKey - This is the key to validate the request.

userPool

To set up cognito user pools, complete the following fields:

userPoolId - Enter the cognito user pool id.

clientId - Enter the cognito user pool client id.

region - Enter the region used in the user pool.

allowedAddresses - Provide specific IP addresses or ranges to restrict access to the webhook registration action. Click **+Add "allowedAddresses"** to add additional IPs.

If no IPs are configured, access is allowed to all IPs. Examples of acceptable formats include:

- Single IP: 192.168.1.1
- Subnet Mask: 192.168.1.1/255.255.0.0
- CIDR Notation: 192.168.1.1/24
- IP Range: 192.168.1.0-192.168.1.240

magicLinkExpiry - The duration, in seconds, for which the link will be accessible (starting from the moment the email is generated).

setPasswordLinkExpiry - The duration, in seconds, for which the link will be accessible (starting from the moment the email is generated).

emailAction - Select from the drop-down menu either **Magic Link** or **Set Password**. When set to **Magic Link** the email will include a KS allowing the user to login and watch the event; when set to **Set Password**, the link will allow the user to set their Cognito password and login.

defaultRole - Select the user application role to be set for newly created users. Choose from the following options:

- viewerRole

- privateOnlyRole
- adminRole
- unmoderatedAdminRole

defaultJoinLinkExpiry - Event Join Link expiry in seconds

eventName - Enter the event name. This name will be used in email notifications if the provided eventId does not resolve to a valid entry ID.

registerLink - Enter link to the event registration form.

defaultConfirmationEmailSubject - You can use {FNAME}, {LNAME} and {EVENT_NAME} to dynamically populate the registration info.

defaultConfirmationEmailBody - You can use {FNAME}, {LNAME}, {EVENT_NAME} and {MAGIC_LINK} to dynamically populate the registration info. HTML tags are supported.

registrationEmailFrom - The email that registration email messages are sent from.

sendCognitoLanguage -

fieldMapping

Click **+Add "fieldMapping"** to add field mapping. An additional section displays.

name - Enter a name.

options - Write every option on a new line.

emailLanguageParamName - Enter the name of language parameter received by set password request.

setPasswordSuccessEmailLocal

Set the list of email templates per language for the set-password-success email.

Click **+Add "tokens"**.

An additional section displays.

setPasswordSuccessEmailLocal Collapse

The list of email templates per language for the set-password-success email.

tokens

DELETE

tokenName	<input type="text"/>	the token name as appears in the email template.
tokenValue	<input type="text"/>	the token value.

[+ Add "tokens"](#)

Complete the following fields:

tokenName - Enter the token name as appears in the email template.

tokenValue - Enter the token value.

languages

Create the list of email templates per language.

Click **+Add "languages"**.

An additional section displays.

languages

The list of email templates per language.

DELETE

language	<input type="text"/>	The language value as sent via the webhook.
templateName	<input type="text"/>	The email template name for this language.

[+ Add "languages"](#)

Complete the following fields:

language - Enter the language value as sent via the webhook.

templateName - Enter the email template name for this language.

setPasswordFailEmailLocal

Set the list of email templates per language for the set-password-fail email.

Click **+Add "tokens"**.

An additional section displays.

setPasswordFailEmailLocal Collapse

The list of email templates per language for the set-password-success email.

tokens

DELETE

tokenName	<input type="text"/>	the token name as appears in the email template.
tokenValue	<input type="text"/>	the token value.

[+ Add "tokens"](#)

Complete the following fields:

tokenName - Enter the token name as appears in the email template.

tokenValue - Enter the token value.

languageMapping

Click **+Add "languageMapping"**.

An additional section displays.

languageMapping

DELETE

externalLanguageCode	<input type="text"/>
language	<input type="text" value="English - en"/>

[+ Add "languageMapping"](#)

Complete the following fields:

externalLanguageCode - Enter the language code.

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

3. Click **Save**.