How to create an AppToken authentication via API?

Last Modified on 11/09/2021 1:20 pm IST

Administrators may use AppToken to generate a custom privileged KS for users who access the Kaltura API. To learn more about AppToken authentication, please visit https://developer.kaltura.com/api-docs/VPaaS-API-Getting-Started/application-tokens.html.

This guide is a supplement of the steps given in https://developer.kaltura.com/workflows/Generate_API_Sessions/App_Token_Authentication.

To create an AppToken authentication via API

1. Open the API console at https://developer.kaltura.com/console/ (Sign In) or http://www.kaltura.com/api_v3/testme.
   1.1 In case Testme Console is used, first, you will need to generate a KS session as demonstrated on How to Start a Kaltura Session using the TestMe Console .
   1.2 Grab the KS string from the output which will be needed in the next step.
2. Get an AppToken ID
   2.1 Set the resulted KS string from step 1 and paste it into the **Kaltura API session** field.
   2.2 Select **appToken.add**
   2.3 Select **KalturaAppToken** > **hashType**. Select the desired hash type (SHA-1 is the default hash function used with Application Tokens.)
   2.4 Select **Send**
   2.5 Grab the id and token from the output which will be needed in the next steps.
3. Start a Widget Session
   3.1 Select **Session.startWidgetSession**
   3.2 Set **widgetId.** The widgetId should be a partnerId (PID), with an underscore at the beginning, e.g. `_1234567`.
   3.3 Select **Send**
   3.4 Grab the KS string from the output which will be needed in the next steps.
4. Compute the Hash
   4.1 Create a string that is compiled from the KS (generate in step 3) and the token (generated in step 2). These 2 values should be concatenated (KS+token, in this order).
   4.2 Take this string and hash it using **the same hash type** as the one used for creating the App Token in step 2.3.
   You may use an online hash generator as http://www.sha1-online.com or https://emn178.github.io/online-tools/sha256.html.
   4.3 Grab the '**tokenHash**' output which will be needed in the next step.

5. Start the App Token Session
   5.1 Set the resulted KS string from step 3 and paste it into the **Kaltura API session** field.
   5.2 Select **appToken.startSession**
   5.3 Set the token **id** string found in step 2.
   5.4 Set the **tokenHash** string found in step 4.
   5.5 (Optional) Set the desired **userId** and/or **type** and/or **expiry** and/or **sessionPrivileges**.
   5.6 Select **Send**
   5.7 The returned output result will reveal the privileged KS.