



CRYSTAL DELTA  
MELBOURNE | CHENNAI



# Loree

Kaltura functionality using App-token

Last Updated on 17<sup>th</sup> July 2020

# Table of Content

1. Introduction
2. Existing Configuration
3. Kaltura App-token
  - a. Required details for Kaltura config via Loree-admin Dashboard
4. Steps for generating App token Using API
5. Benefits of Using SHA-256 hash type



## Introduction

This document refers that the Kaltura setup on Loree Dashboard has to be configure using Kaltura App token and this document explained on how the App token can be generated and what will the benefits for using App Token instead of Kaltura Secret.

## Existing Configuration

In Loree Admin Dashboard, Kaltura account has been configured using Kaltura Partner ID and Secret.

As we received various feedbacks from many Loree users that the recommendation of using App token instead of Kaltura secret, we did more analyse on that and to overcome the Kaltura privacy we finally decided to move the Kaltura configuration from Kaltura secret to Kaltura App token.

## Kaltura App-token

An Application Token is useful in cases where different applications with varying permissions need access to your Kaltura account, without using your Admin Secret. It enables clients to provide their development partners or internal technical teams with restricted access to the Kaltura API.

The appToken is **created and customized by the account administrator**, and then used by the developers to generate Kaltura Sessions for their respective applications. This allows access to the API to be revoked at any time with the deletion of the appToken.

## Required details for Kaltura config via Loree-admin Dashboard

1. App-token id
2. App-token value
3. Hash type (optional)

## Steps for generating App token Using API

<https://developer.kaltura.com/>



Step 1: Click on the above link

Step 2: Sign-in using the Kaltura user-id and password.

Step 3: Choose the API-Docs option

Step 4: In search bar search “`apptoken.add`” and choose the operation on the right side

Step 5: Once you click on the app-token operation, you will see “**Try it out**” button.

Kaltura API Documentation > apptoken.add  
Add new application authentication token  
**Try it out**

POST https://www.kaltura.com/api\_v3/service/apptoken/action/add

**Request Parameters**

Body Query

Global Parameters

body:object

Schema Example

```
object
  appToken:object:KalturaAppToken
  createdAt:integer
  createdAt:integer
  readonly Creation time as Unix timestamp (In seconds)
  description:string
  expiry:integer
  expiry:integer
  Expiry time of current token (unix timestamp in seconds)
  hashType:string
  Enum: MD5 [MD5], SHA1 [SHA1], SHA256 [SHA256], SHA512 [SHA512]
  Enum Type: KalturaAppTokenHashType
  id:string
  readonly The id of the application token
  partnerId:integer
  readonly
  sessionDuration:integer
  Expiry duration of KS (Kaltura Session) that created using the current token (in seconds)
  sessionPrivileges:string
  Comma separated privileges to be applied on KS (Kaltura Session) that created using the current token
  sessionType:integer
  Enum: USER [0], ADMIN [2]
```

**Response**

200

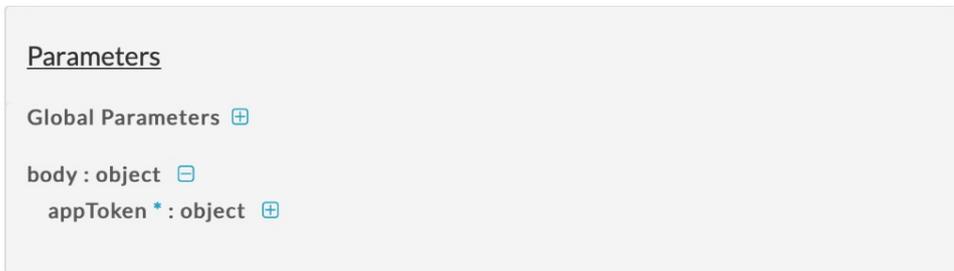
Success

Schema Example

```
object:KalturaAppToken
  createdAt:integer
  readonly Creation time as Unix timestamp (In seconds)
  description:string
  expiry:integer
  Expiry time of current token (unix timestamp in seconds)
  hashType:string
  Enum: MD5 [MD5], SHA1 [SHA1], SHA256 [SHA256], SHA512 [SHA512]
  Enum Type: KalturaAppTokenHashType
  id:string
  readonly The id of the application token
  partnerId:integer
  readonly
  sessionDuration:integer
  Expiry duration of KS (Kaltura Session) that created using the current token (in seconds)
  sessionPrivileges:string
  Comma separated privileges to be applied on KS (Kaltura Session) that created using the current token
  sessionType:integer
  Enum: USER [0], ADMIN [2]
  Enum Type: KalturaSessionType Type of KS (Kaltura Session) that created t
  sessionUserId:string
```

Step 6: Once the Try-out button is clicked You will find a **Send Request** button.

Step 7: Below the Send Request button you will find app Token with “+” icon.



Step 8: Click on the “+” Icon.

Step 9: Fill in the details Such as

“expiry: ‘ ’ ”

“hash Type: SHA256”

“session Type: Admin” (The session type should be admin, since we have accessed the video according to the user in specific admin.)

“sessionId: KalturaUserId”

Note: Please give the expiry as NULL (Don't give any values to this field) Since the app-token won't expiry soon. Where the default time is 24 hours once its expired you have re-generate it, and update again in dashboard.

Step 10: Once all the details are given, click the send request button.

Step 11: You will find a non-expired App-token with Id, value and its SHA-Type.

Alternative ways to create appToken: Request Kaltura support to generate an app token for your Account.

## Benefits of Using SHA-256 hash type

### Kaltura suggests using the SHA-256 for the app-token generation

- collisions are very unlikely to occur
- Block size indicator (byte): 64.
- Maximum allowed message length (bytes): 33.
- The number of iterations in one cycle: 64.



To know more about Application tokens please refer the below link:

<https://developer.kaltura.com/api-docs/VPaaS-API-Getting-Started/application-tokens.html>

