

## How to create a Kaltura session

Last Modified on 03/19/2025 10:42 am IST

 ${}^{\textcircled{a}}_{\textcircled{a}}$  This article is designated for administrators.

## About

To make API calls in Kaltura, you need a Kaltura Session (KS) - a secure key that authenticates your requests. This guide walks you through how to create a KS using the Developer Console or TestMe Console. Once you have your KS, you can use it to interact with Kaltura's API and manage your media programmatically.

•

Accessing the Kaltura APIs requires a Publisher Account. If you don't have such an account, see start a free Kaltura.com trial.

### Via the Developer Console

- 1. Go to https://developer.kaltura.com/api-docs/service/session/action/start
- 2. Click Try it out

Start a session with Kaltura's server.

The result KS is the session key that you should pass to all services that requires a ticket.

Try it out

3. Enter the Partner ID, Administrator Secret, and select type Admin.

Retrieve the Partner ID and Administrator Secret information from your KMC application:

- Click the **Settings** button and then **Integration settings.**
- Copy the following parameters:
  - A. Partner ID
  - B. Administrator Secret.

\*The numbers and letters are different for each Partner.\*

4. Click **Send Request**. The KS string output is displayed on the right side.



Kaltura API Documentation > session.start

Start a session with Kaltura's server.

The result KS is the session key that you should	pass to all services that requires a ticket.

Send Request 🕨	Hide	Expand	
Parameters			
Global Parameters	Ð		
body:object 😑			
secret *: string			
•••••	•••••	•••••	
expiry : integer			
86400			
partnerId : integ	er		
TEDIOSZ			
privileges : string	g		
type : integer			
ADMIN [2]			
userId : string			

Success	
🕀 Head	ers
djJ8NDgzl	NDAzMnxAhwOLos-o2AphZabvz61C 36PHK- EvOzOD60EcaCDaJ3CYsO6FmKgrtvB484IKm2Di5iZ0PiemM-fJkuSCp4

## Via the TestMe Console

- Go to: http://www.kaltura.com/api\_v3/testme for US SaaS accounts or select the TestMe console by your region, such as:
  - EU https://api.eu.kaltura.com/api\_v3/testme/
  - AP https://api.ap.kaltura.com/api\_v3/testme/
  - SG https://api.sg.kaltura.com/api\_v3/testme/
  - CA https://api.ca.kaltura.com/api\_v3/testme/
  - DE https://api.de.kaltura.com/api\_v3/testme/
- 2. Select Session from the Select Service drop-down list.

# 🔆 kaltura

Test Console API Documentation	XML Schema API Client Libraries
History:	
Select request 🔹	
Format (int):	
XML	
oliantTeg (string):	
testme	
3.3.0	
land and and and and and and	
Impersonated partner id (int):	
Kaltura API session (string):	
Select service:	
session	
Select action:	
start 🔻 🎯	

### 3. Select Start from the Select action drop-down list.

Test Console	API Documentation	XML Schema	API Client Libraries
1 Entrany			
Select request	•		
Concernequeer			
Format (int):			
XML	Ψ.		
clientTag (string):			
testme			
apiVersion (string):			
3.3.0			
Impersonated partne	erid (int):		
Kaltura API session	(string):		
	0		
Select service:			
session	• @		
Salact action:			
Select action.			
start	•		

- 4. Retrieve the Partner ID and Administrator Secret information from your KMC application:
  - Click the **Settings** button and then **Integration settings**.
  - Copy the following parameters:



- A. Partner ID
- B. Administrator Secret.

\*The numbers and letters are different for each Partner.\*

⋇	CONTENT	STUDIO	ANALYTICS SERVI	CES DASHBOARD		
			ACCOUNT SETTINGS	INTEGRATION SETTINGS	ACCESS CONTROL	TRANSCODING SETTINGS
			Account Info	)		
			Partner ID	147		
			Sub Partner ID	14	0	
			Administrator Se	ecret eb3bb	170	96967b0
			User Secret	5dccf62	15t	o0f7f3

5. In the TestMe Console, paste the Partner ID and Administrator Secret information, and change the type field from **USER** to **ADMIN**.

# 🔆 kaltura

Test Console	API Docu	mentation	XML Schem	a API Client Libraries
History:		_		
Select request		•		
Format (int):				
XML				
clientTea (string);				
testme				
apiVersion (string):				
3.3.0				
Impersonated partne	rid (int):			
Kaltura API session (	(string):			
Select service:		0		
session		•		
Select action:				
start		<b>v</b> 0	1	
secret (string):		1200h 🖋		
94		12000		
<u> </u>				
userld (string):				
0				
type (KalturaSession	Type):			
ADMIN		▼ 🗹		
0				
partnerld (int):				
1 2		<b>e</b>		
0				
evening (ant):				
expiry (int):				
0				
20				
privileges (string):				
privileges (string):				
privileges (string):				
<ul> <li>privileges (string):</li> <li>Send</li> </ul>				

6. Click Send.

The output results should contain the KS, a Kaltura Session key.

<mark>g7™joetNike 41P</mark>2Mzk4LjM5NDs7Ozs=</mark>K/result>

7. Copy the KS string and put it in the Kaltura API session (string) field for your next API call.

Copyright © 2024 Kaltura Inc. All Rights Reserved. Designated trademarks and brands are the property of their respective owners. Use of this document constitutes acceptance of the Kaltura Terms of Use and Privacy Policy.



That's it! Your API console is ready to make new API calls now.

## •

If you notice that you cannot find all entries/categories in the API call output, this is likely to be related to the **Default entitlements Enforcement** set on your account.



To accommodate this, you may set the ultimate privilege when generating the KS string and bypass this restriction.

To do so, when generating the KS as described in this guide, please add the following string to the "privileges (string):" field:

### all:\*,disableentitlement .



Select service:		
session	<b>∨</b> ⊌	
Select action:		
start	<b>∨</b> 0	
secret (string):		
Θ		
userld (string):		
@kaltura.com		
Θ		
ype (KalturaSessionType):		
ADMIN	~ <mark>~</mark>	
Θ		
partnerld (int):		
Θ		
expiry (int):		
Θ		
privileges (string):	-	
disableentitlement		
0		
Sand		
Sellu		

### Notes

The Kaltura API is **REST** based and stateless. Every call (request) made to the Kaltura API requires an authentication key, the Kaltura Session (aka KS), identifying the account on which the action is to be carried, the authenticated user and its role.

A Kaltura Session can be initiated in 3 different ways:

1. Calling the session.start action, providing the account Partner ID and an Admin or User secret key.

```
$ks = $client->session->start ($adminSecret, $anyUserName, KalturaSessionType::ADMIN, $partnerId);
$client->setKS($ks);
```

2. Calling the user.loginByLoginId action, providing a User login ID (email), its Password and the account Partner ID.



\$ks = \$client->user->loginByLoginId(\$loginId, \$password, \$partnerId); \$client->setKS(\$ks);

3. Creating a session locally - Combine all the Kaltura Session details, and sign them using the shared secret.

\$ks = \$client->generateSession(\$adminSecret, \$userId, \$type, \$partnerId); \$client->setKS(\$ks);

The above examples are using the Kaltura PHP5 Client Library.

### **Important Security Notes:**

### A

ADMIN type KS provides super admin priveliges to the Kaltura account. If you're creating an application where the session will be exposed to the end-user, make sure that you are using a USER type KS and not ADMIN type. Exposing an ADMIN type KS in non-administrative context will expose your Kaltura account to risks of being used by malicious users with unrestricted access.

Sharing the account API secret keys with 3rd party vendors should be avoided, as secret keys can not be regenerated or blocked for access. Kaltura API based application developers and 3rd party application vendors should build their application to leverage the user.loginByLoginId API and ask the publisher for their email, password and account Id (aka partnerId). Users can be easily created, removed or blocked and their password can easily be changed.

To learn more about the KS and its usage read: Kaltura's API Authentication and Security: The Kaltura Session.