Kaltura Player Support for Chromecast

Overview

Player Compatibility

The Kaltura Player supports Chromecast on the following versions:

- Mobile SDK V3
  - iOS: iOS 9, 10
  - Android: Android 4.2.2+
- Web player version 2.46+

Cast API

The Kaltura Player supports Google's Cast V3 APIs.

Introduction to Chromecast

Chromecast enables end users to cast media from their tablets, smartphones and chrome browsers to their TVs during viewing. Therefore, the assumption is that the end user owns and has attached a Chromecast device to their television.

How does It Work?

The Chromecast system consists of the following:

- A **Sender application** that is integrated within the player and is responsible for sending streaming requests to the Chromecast Receiver,
- A **Receiver application**, which is a separate HTML page that runs on the Chromecast dongle attached to the end user's TV, and is responsible for

receiving the streaming requests from the Sender application and presenting it on the TV. Note that for Chromecast to work, the end user's application and the Chromecast device must be connected to the same wireless network.

Figure 1:

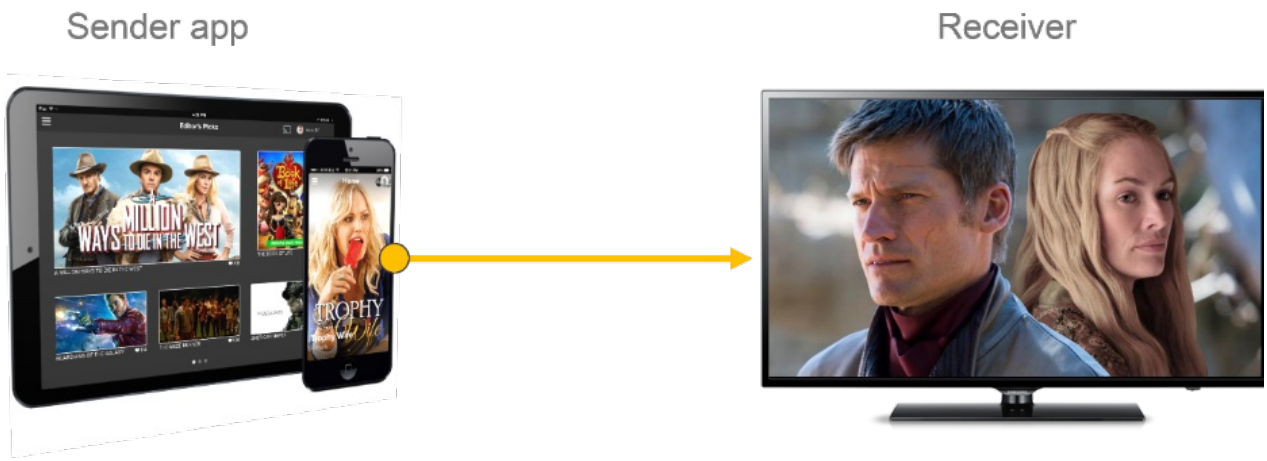


Figure 2:

As pictured in Figure 2 above, the Chromecast receiver handles the communication between itself and:

- The Player
- The Kaltura Backend
- The Google cloud

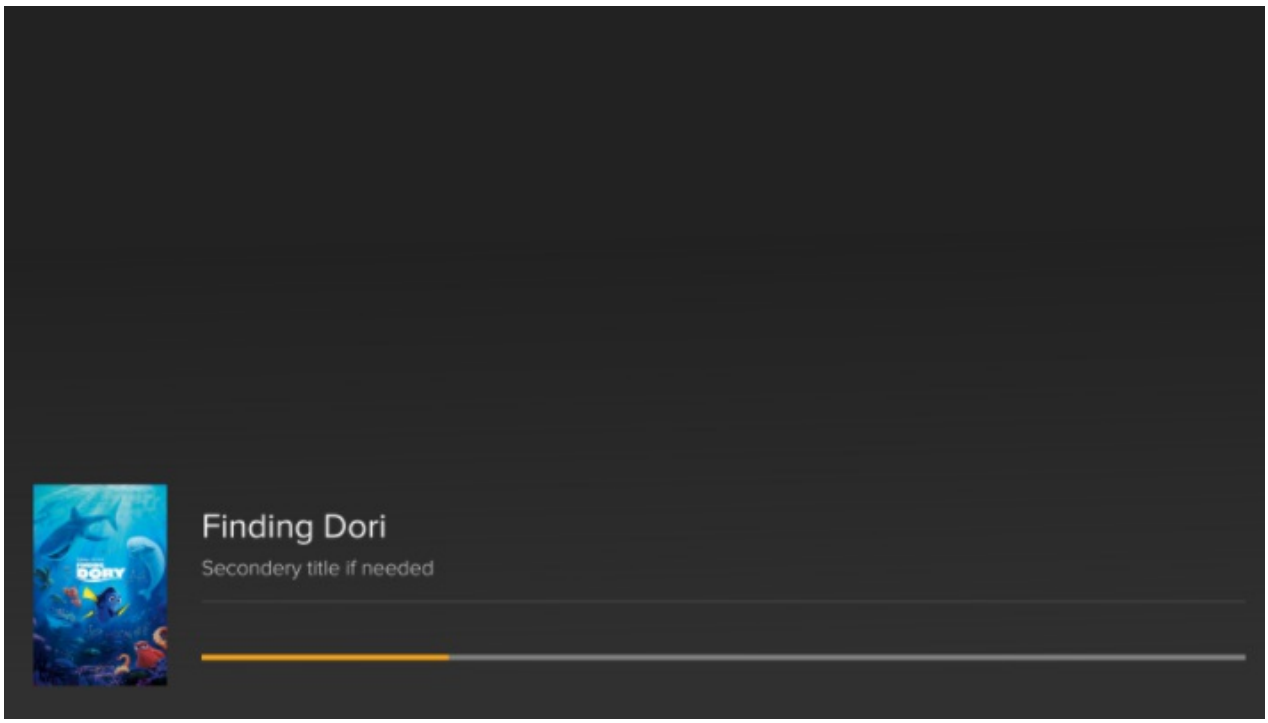
Kaltura's Chromecast Sender App

The Chromecast Sender application represents the application that is responsible for sending streaming requests to the Receiver (the Chromecast device). The Sender application enables mobile and web users to cast content from the player onto a TV screen via the Chromecast device. The Web Player and the Mobile SDKs have separate sender applications. The Sender applications implemented in the player work with a *custom* receiver.

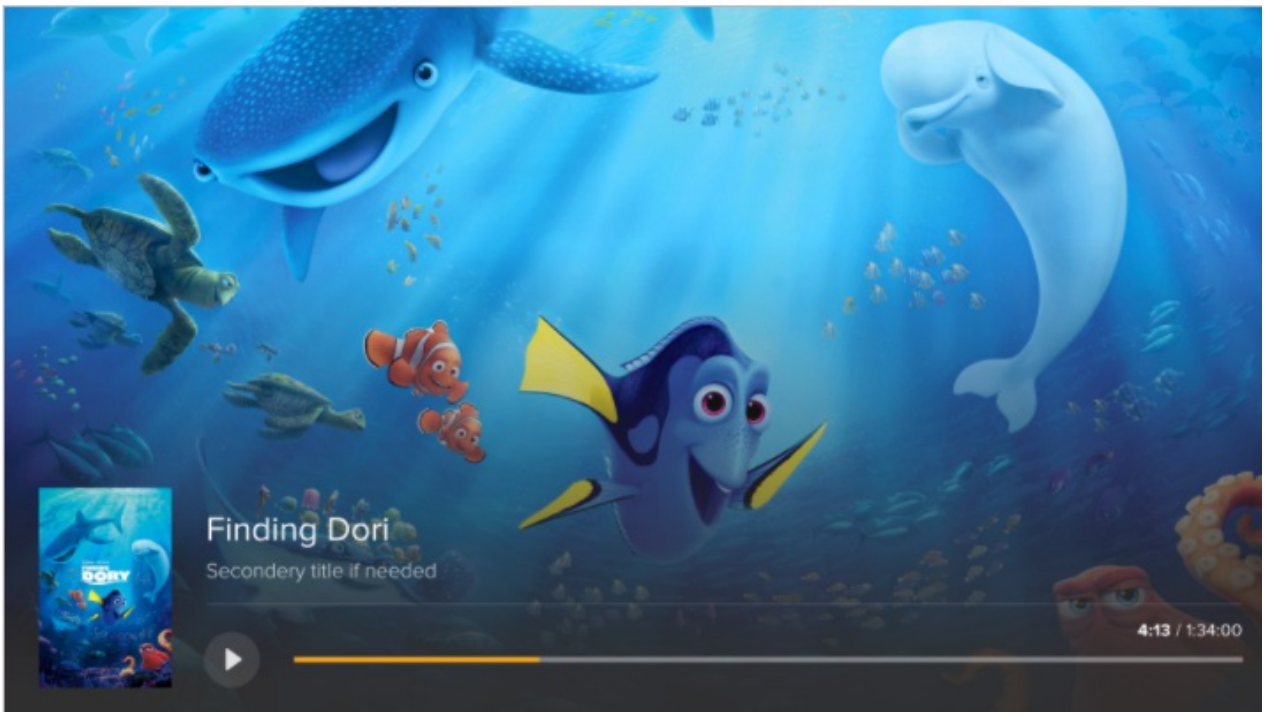
Kaltura's Chromecast Receiver Application

Kaltura uses a Custom Receiver, which is a custom-built HTML5 that can also stream files that use DRM. The receiver TV, depending on the status of the Chromecast connection, displays the following screens:

Content Loading – The Receiver application is loaded and connected and is now loading the video



Content Playing – The Receiver application is loaded, connected and playing



Customizing the Receiver Application

Some of the components on the screen can be configured, including:

Field	Description	Default
spinnerFillColor	Fill color of the loading spinner	#59BAF3
'progressFillColor'	Color of the progress bar	#F8A61A
TitleFont	Title Font	proxima nova
TitleSize	Title Size	32
SubtitleFont	Subtitle Font	proxima nova
SubtitleSize	Subtitle Size	32
AudioTracksMessage	This message appears while switching audio tracks	"Please wait..."
AudioTracksMessageColor	Audio Tracks Message color	#59BAF3
	The time (in seconds) that takes the	

Field	Description	Default
FadeOutPauseTime	media artwork to disappear in case of pause media during playback	5
launchingTimeout	The time (in minutes) until the receiver is shut down when in a launching state	5
LoadingTimeout	The time (in minutes) until the receiver is shut down when in a loading state	5
pausedTimeout	The time (in minutes) until the receiver is shut down when in a pause state	20
idleTimeout	The time (in minutes) until the receiver is shut down when in an idle state	5

Connecting to Chromecast

There are two ways end users can start a casting session:

1. Connect & play: Connect to a casting receiver before playing content.
2. Play & connect: Connect to a casting receiver while playing content.

To start the connection process and open the Cast Section on the mobile device, the end user touches the Chromecast icon. While using the Web player, the end user can easily access the Chromecast icon from the chrome browser or the player itself.

The Chromecast button has three states:

- Not connected (or disconnected)  – This state is displayed when the Sender application is not connected to a Chromecast device.
- Connecting: While the receiver is connecting, the button animates the waves in the icon progressively  - The Connecting (animated) state appears when the connection is being established. Once connected, the Receiver application launches.
- Connected  – This state is displayed when the Sender application is

connected to a Chromecast device (but not necessarily casting).

While using the Mobile SDKs, the entire player UI is implemented by the application (and not the player) and, therefore, it is up to the application to implement the Cast button.

Additional Capabilities

Delivery Profiles

Videos on the Chromecast receiver will playback with HLS or DASH. while the sender app supports HLS, DASH and progressive. Multiple audio tracks and multiple captions are fully supported.

Advertisements

The Kaltura Player supports Pre-roll ads on Chromecast. The player leverages Google's IMA plugin to support any VAST compliant ad server.

DRM

Widevine CENC DRM is supported on the Chromecast plugin.

Analytics

The Kaltura Player tracks player events on the receiver and can therefore provide statistics on Chromecast video playing. Refer to the article Creating and Tracking Statics for more information on Kaltura statistics.

Chromecast Queues

The Player Mobile SDKs support Queuing of VOD assets. Users can be given access to queue controls, letting them add, remove and re-order clips they wish to view. This functionality is currently not supported on the web player.

Disconnections

If the Sender application gets disconnected accidently (e.g., the device battery dies), the content will continue to be shown on the receiver (the TV). The Sender application, once reloaded, should return to its last state. The queue will be maintained. If the Receiver application gets accidently disconnected (e.g., the Wi-Fi issues for instance), the ed user will need to reconnect the application to Chromecast and the queue will be reset.

Mobile SDKs – Additional Documentation

For additional documentation on the Mobile SDKs support of Chromecast, please see:

- Android
  - Android Casting – Get started
  - Code Examples
- iOS
  - iOS Casting – Get Started
  - Code samples