

Advanced player setting: Post message bridge

Last Modified on 04/16/2026 4:19 pm IDT

 This article is designated for all users.

About

The **Post Message Bridge** plugin lets you forward Kaltura HTML5 player events from the player iframe to the embedding page via `window.postMessage` so host applications can enrich and route them to analytics or backends.

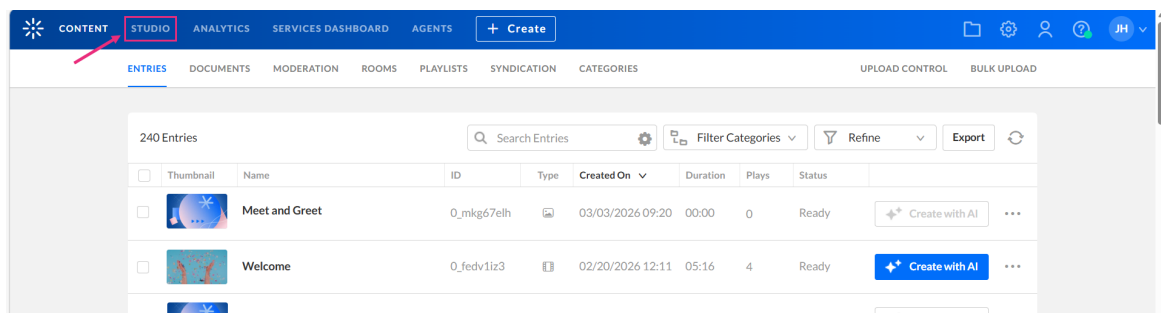


You can enable this plugin through the player's **Advanced settings**. This configuration is not exposed in the standard Player Studio interface.

Access the Advanced settings

VOD **AUDIO** **REELS**

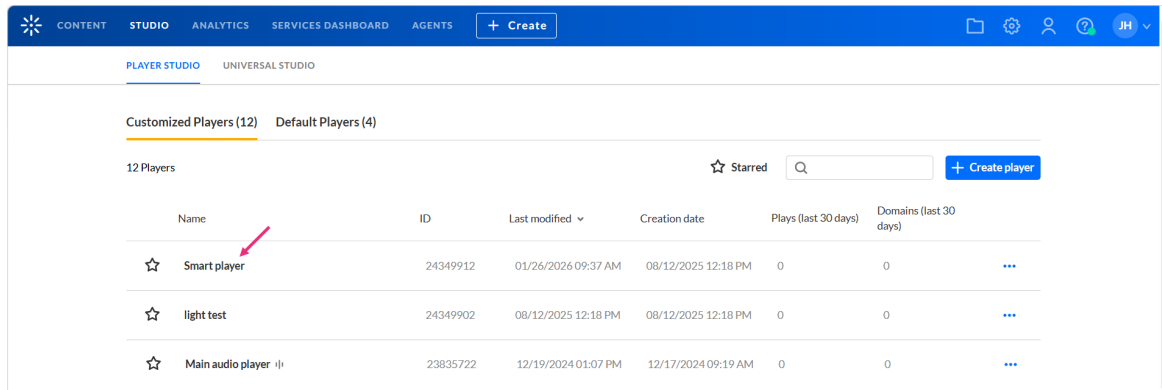
1. Log in to your KMC and click the **Studio** tab.



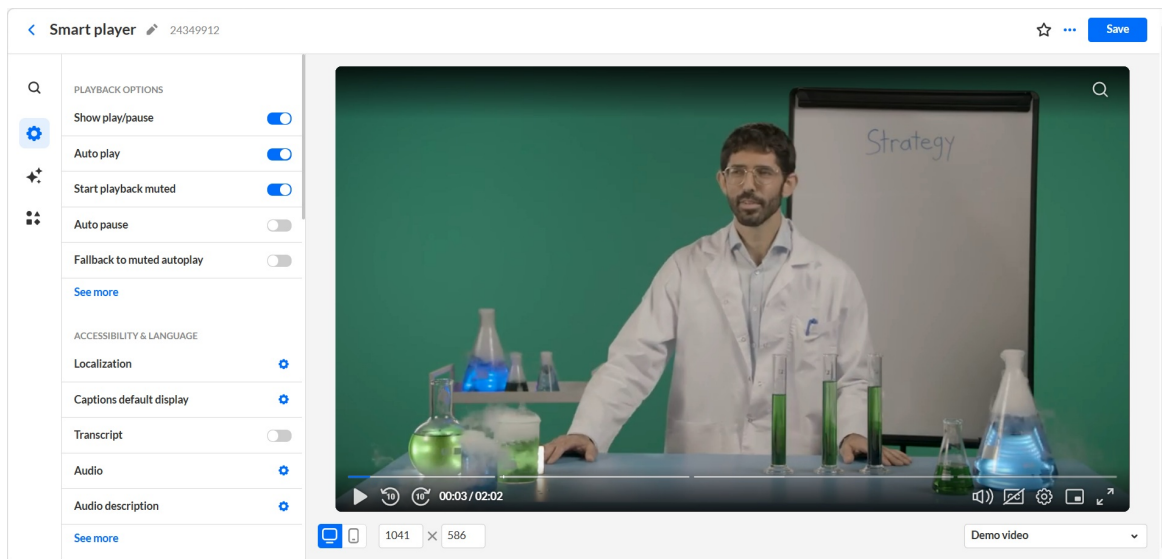
2. The **Player Studio** page displays.
3. Click the player you want to edit from the list (or use the search bar in the top right).



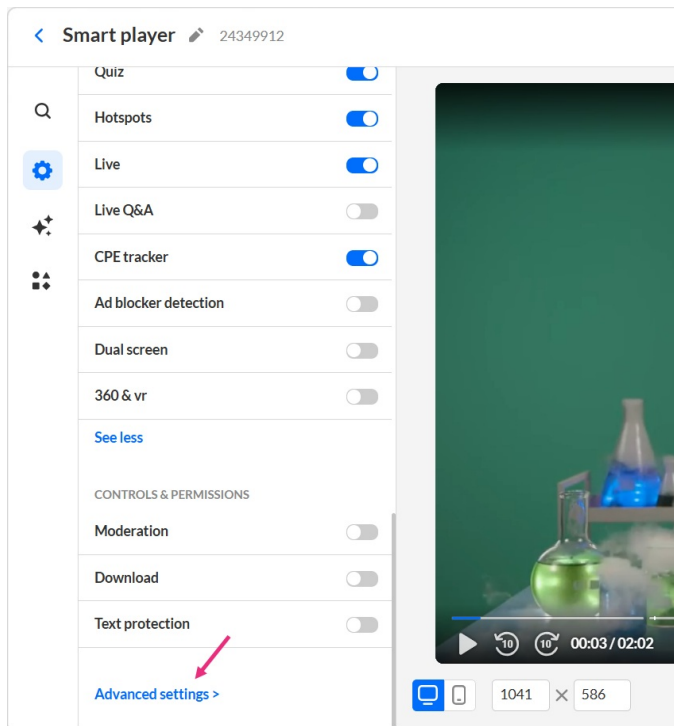
If you haven't created a player yet, check out our article [Create a player](#) for guidance.



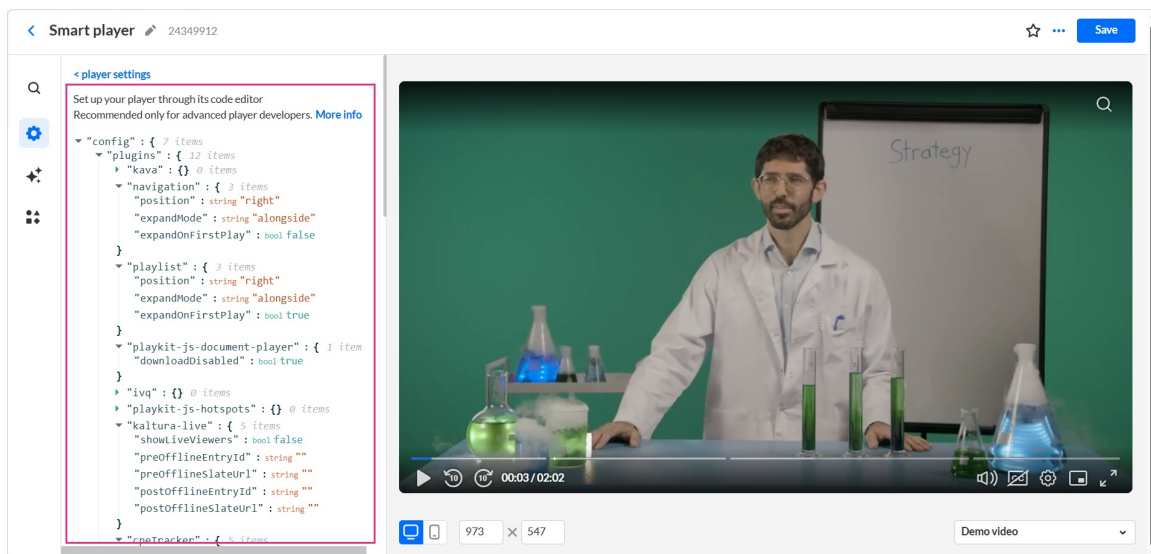
4. The [player settings](#) display.



5. Scroll to the bottom of the player settings and click **Advanced settings**.



The Advanced settings editor displays.



Configure Post Message Bridge

You may configure the Post Message Bridge plugin in the player configuration (uiconf) by adding/enabling the plugin block and its options.

Enable the plugin

1. In the [Advanced Settings](#) tab, hover over "plugins" then click the **green + icon**.

```

"config" : { 7 items +
  "plugins" : { 12 items +
    "kava" : {} 0 items
    "navigation" : { 3 items

```

- In the Key Name box, enter `postMessageBridge` then click the green checkmark or press 'Enter' on your keyboard.

```

"config" : { 7 items
  "plugins" : {
    "kava" : {}
    "navigation" : {
      "position" : string "right"
      "expandMode" : string "alongside"

```

You'll see the new plugin listed and enabled and labeled as "null".

```

"postMessageBridge" : NULL

```

- Hover over the plugin name and click the **green edit icon**.

```

"postMessageBridge" : NULL

```

A box containing a "null" value displays.

```

"postMessageBridge" :
null
NULL

```

- Replace "null" with "{}" then click the second small green checkmark next to the {...}.

```

"postMessageBridge" :
{}
{...}

```

- The new plugin name is now followed by an empty array.

```

"postMessageBridge" : {} 0 items

```

- Click **Save** at the top right before exiting.

Configure which events to send

To send all events:

1. Hover over "postMessageBridge" then click the **green + icon**.

```

}
  ▶ "postMessageBridge" : {} 0 items + ×

```

2. In the Key Name box, enter `sendAllEvents` , then click the green checkmark or press 'Enter' on your keyboard.

```

"config" : { 7 items
  ▶ "plugins" : { 1 item
    ▶ "kaltura" : { 1 item
      ▶ "call" : { 1 item
        ▶ "messages" : [ 1 item

```

Key Name:

✓

You'll see the new setting listed and labeled as "null".

```

}
  ▶ "postMessageBridge" : { 1 item
    "sendAllEvents" : NULL
  }
}

```

3. Hover over "sendAllEvents" and click the green edit icon.

```

}
  ▶ "postMessageBridge" : { 1 item + ×
    "sendAllEvents" : NULL ✎ ×
  }
}

```

A box containing a "null" value displays.

```

}
  ▶ "postMessageBridge" : { 1 item
    "sendAllEvents" :

```

null

✎ ✓

```

    NULL ✓
  }
}

```

4. Replace "null" with "true" then click the small green checkmark.

```

}
▼ "postMessageBridge" : { 1 item
  "sendAllEvents" :
  true
  bool true
}

```

The item is now labeled as "true".

```

}
▼ "postMessageBridge" : { 1 item
  "sendAllEvents" : bool true
}

```

When `sendAllEvents` is true, the plugin listens to all supported PlayKit HTML5 events (including custom and ad events) and forwards them to the host page via `window.postMessage`.

5. Click **Save** at the top right before exiting.

To select which events are forwarded (optional):

1. If "sendAllEvents" has been added, set it to "false".
 - a. Hover over "sendAllEvents" and click the **green edit icon**.

```

}
▼ "postMessageBridge" : { 1 item
  "sendAllEvents" : bool true
}

```

A box containing a "true" value displays.

```

}
▼ "postMessageBridge" : { 1 item
  "sendAllEvents" :
  true
  bool true
}

```

- b. Replace "true" with "false" then click the small green **checkmark**.

```

}
▼ "postMessageBridge" : { 1 item
  "sendAllEvents" :
  false
  bool false
}

```

The item is now labeled as "false".

```

}
  "postMessageBridge" : { 1 item
    "sendAllEvents" : bool false
  }
}

```

c. Click **Save** at the top right before exiting.

2. Add "eventsToSend" and select events.

a. Hover over "postMessageBridge" then click the **green + icon**.

```

}
  "postMessageBridge" : { 1 item  
    "sendAllEvents" : bool false
  }
}

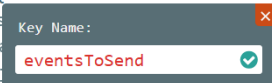
```

b. In the Key Name box, enter `eventsToSend`, then click the green checkmark or press 'Enter' on your keyboard.

```

"config" : { 7 items
  "plugins" : {
    "kav" : {
      "call" : {
        "messages" : [ 1 item
          "eventsToSend" : NULL
        ]
      }
    }
  }
}

```



You'll see the new setting listed and labeled as "null".

```

}
  "postMessageBridge" : { 2 items
    "sendAllEvents" : bool false
    "eventsToSend" : NULL
  }
}

```

c. Hover over "eventsToSend" and click the green edit icon.

```

}
  "postMessageBridge" : { 2 items  
    "sendAllEvents" : bool false
    "eventsToSend" : NULL  
  }
}

```

A box containing a "null" value displays.

```

}
  "postMessageBridge" : { 2 items
    "sendAllEvents" : bool false
    "eventsToSend" :
      null
  }
}

```



- d. Enter groups, event names, or a mix, as a comma-separated string then click the green checkmark or press 'Enter' on your keyboard. In the following example, we are adding one event group - playbackMilestones.

```
}
└─ "postMessageBridge" : { 2 items
  "sendAllEvents" : bool false
  "eventsToSend" :
    playbackMilestones ✕ ✓
}
```

The Events are listed.

```
}
└─ "postMessageBridge" : { 2 items
  "sendAllEvents" : bool false
  "eventsToSend" :
    string "playbackMilestones"
}
```

- e. Click **Save** at the top right before exiting.

Supported event groups:

- `core` – Core playback events such as `media_loaded`, `play`, `pause`, `resume`, `seek`, `ended`
- `playbackMilestones` – Milestone events such as `play_reached_25`, `play_reached_50`, `play_reached_75`, `play_reached_90`, `play_reached_100`
- `errorEvents` – Error events such as `error`
- `adEvents` – Ad plugin events mapped to `ad_start`, `ad_complete`, `ad_click` (and any other ad events supported by the player)

Add optional context fields (optional)

You can include additional identifiers in every message using the optional `contextId` and `channel` fields. These values are passed through as-is in the event payload and can be used by the host page to route or filter events.

```
"plugins": {  
  "hostPageEventBridge": {  
    "sendAllEvents": true,  
    "contextId": "my-company-site-or-page-context",  
    "channel": "company"  
  }  
}
```

- `contextId` – optional string that identifies the context of the player (for example, a site, page, or template identifier).
- `channel` – optional string that identifies the “channel” or namespace for this bridge (for example, `"company"` if you use it in multiple integrations).

Event payload

Example payload:

```
{  
  "source": "kaltura-player",  
  "version": "1.0",  
  "event": "play",  
  "timestamp": "2026-02-26T12:34:56.789Z",  
  "entryId": "1_abcd1234",  
  "playerId": "1234567",  
  "partnerId": 123456,  
  "sessionId": "sess_01",  
  "isLive": false,  
  "positionSeconds": 12.34,  
  "durationSeconds": 300,  
  "playbackRate": 1.0,  
  "contextId": "prod:/content/site/page-x",  
  "channel": "aem"  
}
```

Notes:

- `source` and `version` are **always** sent
- Other fields are sent **only when available** (if a value is missing, that field is omitted)
- `event` matches the logical event name selected by `sendAllEvents` / `eventsToSend`