Record and save videos with IndiVideo components

Content updated: 10/21/2026

> This article is designated for administrators.

About

With **IndiVideo** components, your application can let users record themselves directly into a video, starting from a template or an existing video. You load the template or video, select the scene to record, show the recorder, add the recording to the scene, and save the result as a draft.

Before you start

Set up the IndiVideo component library and authentication in your application. You also need one of these:

- A template code and template language, to create a video from a template
- A media code, to create a new version of an existing video

You can retrieve template codes with `GET /rest/v1/module/template` and video media codes with `GET /rest/v1/module/project`. Use the API base URL and authentication headers for your environment.

Create a video from a template

1. Load the template assembly with its template code and language.
2. Wait for the load to complete, then select the scene to record.
3. Show the recorder component after the scene is selected.

```
window.IndivideoService.loadAssembly({
  templateCode: '<templateCode>',
  templateLang: '<templateLanguage>'
}).subscribe({
  next: () => {
    const sceneIndex = window.IndivideoService.getRecordingSceneIndex();
    window.IndivideoService.selectAssemblyScene({ index: sceneIndex });
    // Show the recorder after the scene is selected.
  },
  error: (err) => console.error('Failed to load template', err)
});
```

`loadAssembly()` returns an Observable. Call `selectAssemblyScene()` only after the assembly has loaded. This example finds the recording scene with `getRecordingSceneIndex()`, which returns `null` if no scene is flagged for recording. You

Copyright © 2025 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.

can also select a scene by its index or context.

To show a script on the teleprompter, get it with `GET /rest/v1/script`, and pass it to `updateVideoScript()` after you select the scene.

```
// script: the script object you got from GET /rest/v1/script
// sceneIndex: the index of the scene you selected
window.IndivideoService.updateVideoScript({ scriptData: script, index: sceneIndex });
```

Add the recorder component to your page:

```
<iv-video-recorder></iv-video-recorder>
```

When the user accepts a recording, the component emits a `videoRecordUpdate` event. The scene doesn't update automatically. Pass the recording's `mediaBlobUrl` to `updateAssemblySceneMedia()`, and wait for it to complete before you save. It returns a cold Observable, so it doesn't run until you subscribe. This method requires component library 29.3.4 or later.

```
document.querySelector('iv-video-recorder').addEventListener('videoRecordUpdate', (event) => {
  const { mediaBlobUrl } = event.detail;
  window.IndivideoService.updateAssemblySceneMedia({ mediaBlobUrl }).subscribe({
    next: () => console.log('Recording added. You can save the video now.'),
    error: (err) => console.error('Failed to add the recording', err)
  });
});
```

Create a new version of an existing video

Load the existing video with its `mediaCode` to create a new version in the same project. Wait for the load to complete before selecting a scene or making changes.

```
window.IndivideoService.loadAssembly({ mediaCode: '<mediaCode>' }).subscribe({
  next: () => {
    const sceneIndex = window.IndivideoService.getRecordingSceneIndex();
    window.IndivideoService.selectAssemblyScene({ index: sceneIndex });
    // Continue with edits, then record and save the new version.
  },
  error: (err) => console.error('Failed to load video', err)
});
```

After making the changes and adding any new recording to the assembly, save the video.

Save the video

After the recording is added to the assembly, save the video. From a template, `saveVideo()` creates the first version of the video. From an existing video, it creates a new version. Either way, the video is saved as a draft. Subscribe to the returned

Copyright © 2025 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.

Observable, and keep the `mediaCode` for later operations.

```
window.IndivideoService.saveVideo({ videoTitle: 'My video' }).subscribe({
  next: (result) => {
    const mediaCode = result.mediaItem.mediaCode;
    console.log('Video saved as draft:', mediaCode);
  },
  error: (err) => console.error('Failed to save video', err)
});
```

To publish a video or get its download links, see Publish and download IndiVideo videos.

Copyright © 2025 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.