Kaltura's JS plugin boilerplate

Last Modified on 09/15/2025 2:06 pm IDT

This article is designated for administrators.

About

The PlayKit JS plugin boilerplate is a preconfigured development template for building custom Kaltura Player plugins. It includes support for ES6 and TypeScript, a local development server, automated linting and testing, and a set of UI services to simplify player customization.

The plugin groups several UI-related services into a clean, developer-friendly API. Each service manages a specific area of the player UI and leverages the Kaltura Player Core API for integration.

The project is written in ECMAScript6 (.js) and TypeScript (.ts), and transpiled to ECMAScript5 using Babel and the TypeScript compiler. Webpack is used to build the distribution bundle and serve the local development environment.

Features

Add your plugin-specific features here. For example: custom buttons, overlays, metadata display, or player layout tools.

Getting started with the development

```
# First, checkout the repository and install the required dependencies
git clone https://github.com/kaltura/playkit-js-plugin-example.git
# Navigate to the repo dir
cd playkit-js-plugin-example
# Run dev-server for demo page (recompiles on file-watch, and write to actual dist fs artifacts)
npm run dev
# Before submitting a PR - Run the pre commit command
npm run pre:commit
# this command will run:

# 1. types check
# 2. lint check
# 3. generate/update types
# 4. generate/update docs
```

The dev server will host files on port 8000. Once started, the demo can be found running at http://localhost:8000/.

Before submitting a PR, please see our contribution guidelines.

Linter (ESlint)

Run linter:

```
npm run lint:check
```

Run linter with auto-fix mode:

```
npm run lint:fix
```

Formatting Code

Run prettier to format code

```
npm run prettier:fix
```

Type Check

Run type-check to verify TypeScript types

```
npm run types:check
```

Automated tests (Mocha/Karma)

Run all tests at once:

```
npm test
```

Run unit tests in watch mode:

```
npm run test:watch
```

Design

An overview of this project's design, can be found here.

Usage guide

Usage guide

Demo

https://kaltura.github.io/playkit-js-plugin-example/demo/index.html

Compatibility

playkit-js-plugin-example is only compatible with browsers supporting MediaSource extensions (MSE) API with 'video/MP4' mime-type inputs.

playkit-js-plugin-example is supported on:

- Chrome 39+ for Android
- Chrome 39+ for Desktop
- Firefox 41+ for Android
- Firefox 42+ for Desktop
- IE11 for Windows 8.1+
- Edge for Windows 10+
- Safari 8+ for MacOS 10.10+
- Safari for ipadOS 13+

License

playkit-js-plugin-example is released under Apache 2.0 License