 [Features](#) [Explore](#) [Pricing](#)

This repository

[Sign in](#) or [Sign up](#)

ctd1500 / **videojs-hotkeys**

[Watch](#) 7 [Star](#) 62 [Fork](#) 45

[Code](#)

[Issues](#) 4

[Pull requests](#) 2

[Projects](#) 0

[Pulse](#)

[Graphs](#)

Adds more hotkey support to video.js

videojs

video-js

hotkeys

🕒 113 commits	🌿 2 branches	📦 19 releases	👤 12 contributors	📄 Apache-2.0
Branch: master ▾ New pull request			Find file	Clone or download ▾
ctd1500 Adding Readme doc for enableInactiveFocus ... Latest commit ae36b51 28 days ago				
.editorconfig	Initial commit.	3 years ago		
.gitignore	Release 0.2.0	2 years ago		
Gruntfile.js	Updating keywords	7 months ago		
LICENSE.md	Updating license	a year ago		
README.md	Adding Readme doc for enableInactiveFocus	28 days ago		
bower.json	Updating keywords	7 months ago		
example.html	Adding number key override to example.html	4 months ago		
package.json	Release 0.2.17	4 months ago		
videojs.hotkeys.js	Adding Readme doc for enableInactiveFocus	28 days ago		
videojs.hotkeys.min.js	Release 0.2.17	4 months ago		

📖 README.md

videojs-hotkeys

- [Introduction](#)
- [Usage](#)
 - [CDN version](#)
 - [Self hosted](#)
 - [Enable the plugin](#)
- [Options](#)
- [Custom Hotkeys and Overrides](#)
 - [Override existing hotkeys](#)
 - [Create new custom hotkeys](#)

Introduction

A plugin for Video.js that enables keyboard hotkeys when the player has focus.

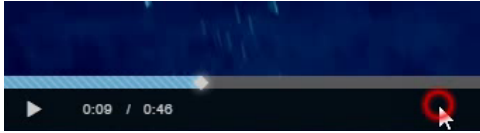
- Space bar toggles play/pause.
- Right and Left Arrow keys seek the video forwards and back.
- Up and Down Arrow keys increase and decrease the volume.
- M key toggles mute/unmute.
- F key toggles fullscreen off and on. (Does not work in Internet Explorer, it seems to be a limitation where scripts cannot request fullscreen without a mouse click)

- Double-clicking with the mouse toggles fullscreen off and on.
- Number keys from 0-9 skip to a percentage of the video. 0 is 0% and 9 is 90%.

Note: clicking any of the control buttons such as Play/Pause, Fullscreen, or Mute, will remove focus on the player which appears to "break" the hotkeys. This is for accessibility reasons so that people who do not use or know about the hotkeys can still properly use the `Tab` key to highlight the control buttons and press `space` to toggle them.

To restore focus, just click on the video, or an empty part of the control bar at the bottom of the video player.

To override this behaviour, set the flag `alwaysCaptureHotkeys` to `true`. This will "fix" hotkeys. For accessibility, the `Tab` key may be used in combination with the `Enter / Return` key to navigate and activate control buttons.



Usage

Include the plugin:

CDN version

You can either load the current release:

```
<script src="//cdn.sc.gl/videojs-hotkeys/0.2/videojs.hotkeys.min.js"></script>
```

Or always load the latest version:

```
<script src="//cdn.sc.gl/videojs-hotkeys/latest/videojs.hotkeys.min.js"></script>
```

Self hosted

```
<script src="/path/to/videojs.hotkeys.js"></script>
```

Enable the plugin

Add hotkeys to your Videojs ready function. Check the [Options](#) section below for the available options and their meaning.

```
videojs('vidId').ready(function() {  
  this.hotkeys({  
    volumeStep: 0.1,  
    seekStep: 5,  
    enableModifiersForNumbers: false  
  });  
});
```

Options

- `volumeStep` (decimal): The percentage to increase/decrease the volume level when using the Up and Down Arrow keys (default: `0.1`)
- `seekStep` (integer): The number of seconds to seek forward and backwards when using the Right and Left Arrow keys (default: `5`)
- `enableMute` (boolean): Enables the volume mute to be toggle by pressing the `M` key (default: `true`)
- `enableVolumeScroll` (boolean): Enables increasing/decreasing the volume by scrolling the mouse wheel (default: `true`)
- `enableFullscreen` (boolean): Enables toggling the video fullscreen by pressing the `F` key (default: `true`)
- `enableNumbers` (boolean): Enables seeking the video by pressing the number keys (default: `true`)
- `enableModifiersForNumbers` (boolean): Enables the use of `Ctrl/Alt/Cmd` + Number keys for skipping around in the video, instead of switching browser tabs. This is enabled by default due to backwards compatibility [PR #35](#) (default: `true`)

- `alwaysCaptureHotkeys` (boolean): Forces the capture of hotkeys, even when control elements are focused. The **Enter/Return** key may be used instead to activate the control elements (default: `false`)
- `enableInactiveFocus` (boolean): This reassigns focus to the player when the control bar fades out after a user has clicked a button on the control bar (default: `true`)
- `enableJogStyle` (boolean): Enables seeking the video in a broadcast-style jog by pressing the Up and Down Arrow keys. `seekStep` will also need to be changed to get a proper broadcast-style jog. This feature and the changes for `seekStep` are explained a bit more in [PR #12](#) (default `false`) (**Note:** This isn't a feature for everyone, and enabling `JogStyle` will disable the volume hotkeys)

There are more options specifically for customizing hotkeys described below.

Custom Hotkeys and Overrides

There are 2 methods available here. Simply overriding existing hotkeys, and creating new custom hotkeys.

Override existing hotkeys

Override functions are available for changing which, or adding additional, keys that are used as hotkeys to trigger an action.

Any override functions that you build **must** return a boolean. `true` if the matching key(s) were pressed, or `false` if they were not.

- `playPauseKey` (function): This function can be used to override the Play/Pause toggle hotkey (Default key: **Space**)
- `rewindKey` (function): This function can override the key for seeking backward/left in the video (Default key: **Left Arrow**)
- `forwardKey` (function): This function can override the key for seeking forward/right in the video (Default key: **Right Arrow**)
- `volumeUpKey` (function): This function can override the key for increasing the volume (Default key: **Up Arrow**)
- `volumeDownKey` (function): This function can override the key for decreasing the volume (Default key: **Down Arrow**)
- `muteKey` (function): This function can override the key for the volume muting toggle (Default key: **M**)
- `fullscreenKey` (function): This function can override the key for the fullscreen toggle (Default key: **F**)

These allow you to change keys such as, instead of, or in addition to, "F" for Fullscreen, you can make Ctrl+Enter trigger fullscreen as well. Example usage:

```
videojs('vidId').ready(function() {
  this.hotkeys({
    volumeStep: 0.1,
    fullscreenKey: function(event, player) {
      // override fullscreen to trigger when pressing the F key or Ctrl+Enter
      return ((event.which === 70) || (event.ctrlKey && event.which === 13));
    }
  });
});
```

Create new custom hotkeys

- `customKeys` (object): Create an object containing 1 or more sub-objects. Each sub-object must contain a `key` function and `handler` function
 - `key` (function): This function checks if the chosen keys were pressed. It **must** return a boolean, `true` if the keys match.
 - `handler` (function): This function runs your custom code if the result of the `key` function was `true`.

```
videojs('vidId').ready(function() {
  this.hotkeys({
    volumeStep: 0.1,
    customKeys: {
      // Create custom hotkeys
      ctrlIdKey: {
        key: function(event) {
          // Toggle something with CTRL + D Key
          return (event.ctrlKey && event.which === 68);
        },
        handler: function(player, options, event) {
          // Using mute as an example
          if (options.enableMute) {
```

```
        player.muted(!player.muted());  
    }  
  }  
}  
});  
});
```

There are more usage examples available in the source code of the [example file](#).

