WEBPACK

Hot Module Replacement

Sreekanth M. E.

Freelance Trainer & Consultant

http://www.SreekanthME.com

Hot Module Replacement (HMR) is a feature of webpack that watches changes to source files, and signals the browser to replace updated modules, but not reload the entire page

module just refers to each of the source code files (JS, CSS etc.).

The major benefit of HMR is that app state is not lost.

For example, in case of an SPA, developer would have performed a number of actions to get to a specific state. HMR will not perform a full page refresh and lose the history of those actions.

Benefits of HMR are:

- Retain application state during code changes.
- Improve developer productivity.
- Effective tweaking of CSS in real-time.

Webpack installs the HMR Runtime into the output bundle.

HMR runtime runs in the browser and receives module updates.

HMR can be used during development as a better alternative to Live Reloading.

webpack-dev-server which supports Live Reloading is also used for HMR.

There are three steps to enable HMR in webpack.

- Adding HotModuleReplacementPlugin.
- Setting WDS hot flag.
- Adding HMR code to source files.

STEP-1

Add the HotModuleReplacementPlugin to webpack.config.js, like so:

```
plugins: [
  new webpack.HotModuleReplacementPlugin(),
]
```

STEP-2

Run WDS with the devServer.hot key set to true.

STEP-3

Tell webpack which file (and all its dependencies) can be replaced by using the HMR JS API aka module.hot API:

```
if (module.hot) {
  module.hot.accept();
}
```

HotModuleReplacementPlugin writes logs to the console. It logs whenever a module is updated.

To make HMR work with CSS files. Only additional thing needed is to use style-loader.

Style-loader uses module.hot.accept behind the scenes to patch <style> tags when CSS dependencies are updated.

END OF CHAPTER

APPENDIX