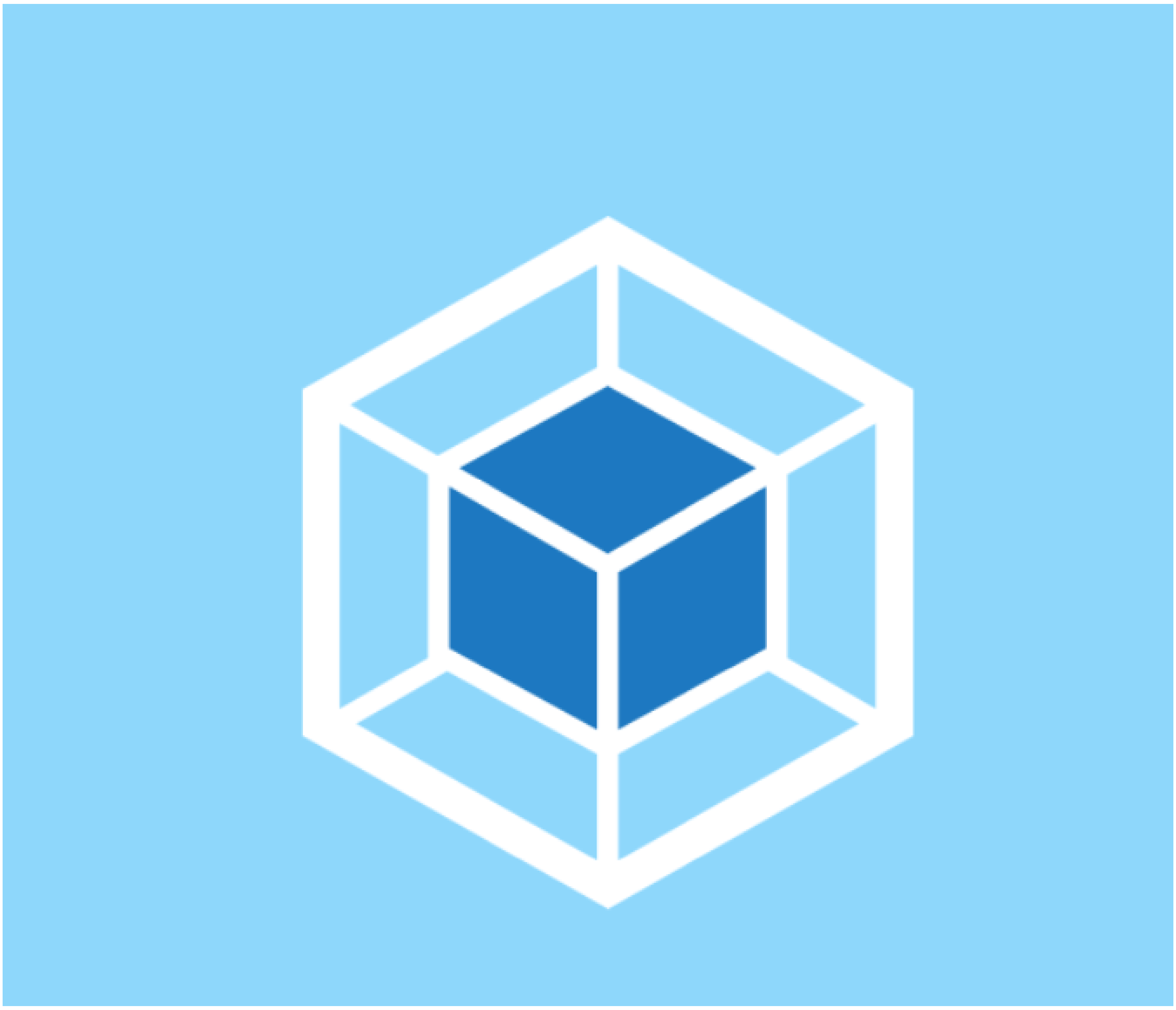
How to hot reload grapphql using webpacks Hot Module Replacement Plugin

Craig Godden-Payne
Nov 12, 2019 · 3 min read ★



Hot Module Replacement will exchange, add or remove a part of your application, such as modules or in our case graphql schema and inner workings, without the need of a full reload.

There are many reasons why you would want this over killing a server and restarting

- You can retain an application state, which would normally be lost during a full reload.
- Reloading the modifications has significant speed improvement over rebuilding the application and restarting.
- Prevents the annoying task of having to keep performing a kill, build, start process manually.

Using webpacks HotModuleReplacementPlugin

I recently was working on a graphql service, which was using webpack so HotModuleReplacementPlugin was a natural choice.

To set this up was very straightforward. I here is an example of my webpack config.

```
const webpack = require("webpack");
const path = require("path");
const nodeExternals = require("webpack-node-externals");
const StartServerPlugin = require("start-server-webpack-plugin");

module.exports = {
  entry: ["webpack/hot/poll?1000", "./src/index"],
  watch: true,
  target: "node",
  node: {
    __filename: true,
    __dirname: true
  },
  devtool: 'eval-source-map',
  externals: [nodeExternals({ whitelist: ["webpack/hot/poll?1000"]
})],
  module: {
    rules: [
      {
        test: /\.js?$/,
        use: [
          {
            loader: "babel-loader",
            options: {
              babelrc: false,
              presets: [["env", { modules: false }], "stage-0"],
              plugins: ["transform-regenerator", "transform-
runtime"]
            }
          }
        ],
        exclude: /node_modules/
      }
    ]
  },
  plugins: [
    new StartServerPlugin("server.js"),
    new webpack.NamedModulesPlugin(),
    new webpack.HotModuleReplacementPlugin(),
    new webpack.NoEmitOnErrorsPlugin(),
    new webpack.DefinePlugin({
      "process.env": { BUILD_TARGET: JSON.stringify("server") }
    })
  ],
  output: { path: path.join(__dirname, "dist"), filename:
"server.js" }
};
```

If you notice the plugin section you will see that I have included plugins which I need to enable the hot reloading.

The plugin will allow us to access a property on module, which will allow us to reload certain parts of the code when it detects a change.

I am then able to reload by removing the listener on the server, and swapping it out with the reloaded module as so:

Webpack entry point (index.js)

```
import http from 'http';
import app from './server';
const server = http.createServer(app);
let currentApp = app;

server.listen(8888, () => {console.log(`GraphQL-server listening on
port 8888.`)});
if (module.hot) {
  module.hot.accept(['./server', './schema'], () => {
    server.removeListener('request', currentApp);
    server.on('request', app);
    currentApp = app;
  });
}
```

Grapqhl Server setup (server.js)

```
import 'core-js/stable';
import AWS from 'aws-sdk';
import 'regenerator-runtime/runtime';
import {SchemaBuilder} from './schema';
import expressPlayground from 'graphql-playground-middleware-
express'
import cors from 'cors'
import express from 'express';
import bodyParser from 'body-parser';
import { ApolloServer } from 'apollo-server-express';
const app = express();
const schemaBuilder = new SchemaBuilder().build;

AWS.config.update({region: 'eu-west-1'});
const server = new ApolloServer({ schema: schemaBuilder});
server.applyMiddleware({ app });
app.use(cors())
app.get('/', (res) => res.status(200).json({status:'healthy'}))
app.get('/playground', expressPlayground({ endpoint: '/graphql' }))
app.use('/graphql', bodyParser.json());
app.use(function (err, req, res, next) {
    console.error(err.stack);
    res.status(500).send('Something broke!');
});
export default app;
```

More information on hot reloading can be found within the webpack documentation, here: https://webpack.js.org/guides/hot-module-replacement/

Written on November 7, 2019, original here: https://craig.goddenpayne.co.uk/hot-reloading-graphql/

JavaScript Webpack GraphQL