

Настройка окружения

№ урока: 2 **Курс:** React Advanced

Средства обучения: Текстовый редактор или IDE, браузер, Node.js

Обзор, цель и назначение урока

В данном уроке мы с вами произведем настройку окружения, используя Webpack2, для того, чтобы в следующем уроке начать познавать азы Redux уже на практике.

Изучив материал данного занятия, учащийся сможет:

- Настраивать окружение самостоятельно, используя Webpack2

Содержание урока

1. Создание всех необходимых файлов и директорий.
2. Описание конфига Webpack.
3. Проверка работоспособности нашего конфига на примере.

Резюме

Совсем недавно был релиз Webpack 2й версии, также вместе с ним обновился и webpack-dev-server. У них на данный момент разные версии, но пусть это вас не смущает – все отлично работает. В нашем проекте мы будем использовать именно их.

Webpack2 хорошо прооптимизировали, улучшили производительность и добавили некоторые вкусные штучки, вроде:

- Нативная поддержка ES6 модулей
- Code Splitting при помощи ES6
- Новый более удобный синтаксис объявления правил
- ES6 специфичные оптимизации

К сожалению, нет одного большого change-log со всеми изменениями, но в интернете есть много статей с обзором новых фишек в Webpack2 и сравнением с 1й версией.

К тому же эти ребята создали новый сайт с прекрасной документацией - <https://webpack.js.org/>
В отличие от 1-й версии, информация на сайте представлена более структурировано и понятно, а также там есть отличные гайды – вроде миграции с 1-й версии на 2-ю.

Описание свойств конфига и их назначение:

“entry” – место старта приложения. Есть возможность перечислить в массиве несколько entry-points (например, отдельный бандл для вендоров).

“output” – место, куда будут помещаться наши бандлы.

“module -> rules” – перечень правил, которые указывают Webpack, как обрабатывать те или иные файлы. Во 2-й версии правила описываются с помощью нового синтаксиса, обратите на это внимание, если вы работали с 1-й версией.

“plugins” – перечень плагинов. Можно отметить WebpackHtmlPlugin, который позволяет автоматически встраивать <script> с нашими бандлами в наш index.html, делает он это на

основании нашего “output”. Также нам в разработке будет очень сильно помогать встроенный в webpack плагин под названием HotModuleReplacement, о нем я уже упоминал в предыдущем уроке.

В свойстве “devServer” мы указываем все необходимые настройки для нашего Webpack-dev-server, такие как, например, порт, исходная директория и пр.

В файле `.babelrc` мы конфигурируем наш Babel, указываем ему плагины и пресеты, которые нужно использовать при транспиляции JS кода.

Закрепление материала

- Почему мы используем Webpack версии 2.x?
- Как сконфигурировать Babel?
- Что делает HtmlWebpackPlugin?
- Как выглядит новый синтаксис для описания правил?
- Зачем использовать url-loader?
- Можно ли использовать несколько entry-points в Webpack?

Рекомендуемые ресурсы

Сайт с документацией по Webpack 2.x

<http://webpack.js.org>