

Лабораторная работа № 2

Инструменты сборки. Grunt и Gulp

Время выдачи работы: 4-я неделя.

Срок защиты работы: 7-я неделя.

1. ЦЕЛИ РАБОТЫ

Выработать навыки использования инструментов сборки при разработке клиентской части веб-приложения.

2. ЗАДАНИЕ

Для веб-приложения, разработанного в ходе выполнения лабораторной работы № 1, необходимо создать скрипт сборки (используемый инструмент сборки Grunt/Gulp необходимо согласовать с преподавателем). Скрипт должен обеспечивать:

- сборку **версии для разработчика**. В этой версии в целевую папку (и/или ее подпапки) копируются все ресурсы (HTML-страницы, стили, скрипты и проч.) приложения, а также все ресурсы подключенных Bower-библиотек;
- сборку **промышленной версии**. В этой версии в целевую папку копируются все ресурсы приложения и подключенных Bower-библиотек, при этом осуществляется их оптимизация. *Необходимо* реализовать: объединение и минификацию стилей и скриптов. *Дополнительно можно* реализовать: минификацию HTML, удаление неиспользуемых стилей, оптимизацию изображений, «встраивание (inlining)» стилей и/или изображений, автоматическое формирование тегов `<script>` для подключаемых скриптов и тегов `<link>` для подключаемых стилей и проч.

Также в скрипте можно дополнительно реализовать:

- запуск **режима «наблюдения (watch)»** для разработчика. В этом режиме осуществляется сборка версии для разработчика, а после этого при любом изменении в папке с исходными кодами осуществляется автоматическое обновление соответствующей части собранного приложения;

- запуск простейшего **HTTP-сервера** для отдачи статических файлов (HTML, CSS, JavaScript). Также можно реализовать возможность автоматического обновления открытых страниц в браузерах при изменении файлов.

3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Установка Grunt/Gulp и их плагинов осуществляется средствами менеджера пакетов NPM. Причем для самого Grunt/Gulp необходимо выполнить как глобальную, так и локальную установку. Глобальная установка предоставляет возможность выполнение команд интерфейса командной строки, а локальная – обеспечивает доступность ядра Grunt/Gulp в рамках проекта.

Установка Grunt:

```
$ npm i -g grunt-cli
$ npm i -D grunt
```

Установка Gulp:

```
$ npm i -g gulp
$ npm i -D gulp
```

Установка плагинов (примеры):

```
$ npm i -D grunt-contrib-uglify
$ npm i -D gulp-cssso
```

Конфигурация сборки описывается в файле Gruntfile.js для Grunt (описание структуры см. <http://gruntjs.com/getting-started#the-gruntfile>) и в файле Gulpfile.js для Gulp (пример см. <https://github.com/gulpjs/gulp#sample-gulpfilejs>).

Примеры плагинов, которые можно использовать для решения задач лабораторной работы:

Задача	Grunt	Gulp
Удаление файлов/папок	grunt-contrib-clean	Del
Копирование	grunt-contrib-copy	<i>из коробки</i>
Получение ресурсов Bower	main-bower-files	bower-main-files
Объединение ресурсов	grunt-contrib-concat, grunt-usemin	gulp-concat, gulp-useref
Минификация JavaScript	grunt-contrib-uglify	gulp-uglify
Минификация HTML	grunt-contrib-htmlmin	gulp-minify-html
Минификация CSS	grunt-contrib-cssmin	gulp-cssso
Удаление неиспользуемых стилей	grunt-uncss	gulp-uncss
Добавление vendor-префиксов к стилям	grunt-autoprefixer	gulp-autoprefixer
Встраивание стилей	grunt-inline-css	gulp-inline-css

Задача	Grunt	Gulp
Импорт в HTML	grunt-usemin	gulp-inject + gulp-order, gulp-useref
Оптимизация изображений	grunt-contrib-imagemin	gulp-imagemin
Построение sourcemaps	–	gulp-sourcemaps
Удаление отладочных команд	grunt-strip-debug	gulp-strip-debug
Режим наблюдения (watch)	grunt-contrib-watch	<i>из коробки</i>
HTTP-сервер	grunt-contrib-connect	browser-sync, connect, gulp-webserver
Упрощение конфигурации сборки	load-grunt-config	gulp-load-plugins

Запуск задачи Grunt (примеры):

```
$ grunt
$ grunt foo
$ grunt foo:bar
```

Запуск задачи Gulp (примеры):

```
$ gulp
$ gulp foo
```

Среда разработки JetBrains WebStorm из коробки обеспечивает интеграцию с инструментами Grunt (начиная с 8-й версии IDE, см. <https://www.jetbrains.com/webstorm/help/using-grunt-task-runner.html>) и Gulp (начиная с 9-й версии IDE, см. <https://www.jetbrains.com/webstorm/help/using-gulp-task-runner.html>). Поддерживается как отображение всего списка сконфигурированных задач, так и выполнение задачи.

4. ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ

1. Опишите назначение инструментов сборки. Приведите примеры инструментов сборки для платформ Java, .NET.
2. Охарактеризуйте инструмент сборки Grunt.
3. Опишите порядок установки и конфигурации Grunt.
4. Поясните значение понятий «задача (task)» и «цель (target)», используемых при работе с Grunt.
5. Как выполнить задачу Grunt?
6. Охарактеризуйте инструмент сборки Gulp. Раскройте значение понятий «поток (stream)» и «конвейер (pipe)».
7. Опишите порядок установки и конфигурации Gulp.

8. Как выполнить задачу Gulp?
9. Назовите основные типовые шаги, которые выполняются перед публикацией веб-приложений.
10. Поясните, как при разработке веб-приложений используется режим «наблюдения (watch)».

5. СОДЕРЖАНИЕ ОТЧЕТА

Отчет по лабораторной работе должен содержать:

1. титульный лист установленного образца с указанными ФИО студента и номером варианта;
2. постановку задачи;
3. описание хода выполнения работы с указанием всех команд (установка инструмента сборки, плагинов, сборка проекта в разных режимах и т.п.);
4. листинг манифеста NPM;
5. листинг конфигурации сборки;
6. выводы.

В бумажном виде отчет предоставляется студентом на защиту лабораторной работы. Кроме того, в информационную систему «eCampus» необходимо загрузить отчет в формате PDF и архив, содержащий:

1. отчет в формате DOCX (MS Office Word 2007 и выше) или DOC (MS Office Word 2003);
2. отчет в формате PDF;
3. и полученное решение (папку с проектом).

Архив необходимо назвать в формате «ФИО, группа» (например: «*Константинов И.П., ИТД-111*»), при загрузке нужно указать теги: «*Отчет 2; 2015*».

6. СПИСОК ОСНОВНОЙ ЛИТЕРАТУРЫ

1. Matt Bailey. Руководство по использованию Grunt для начинающих: пересмотренное издание. — Frontender Magazine. — <http://frontender.info/a-beginners-guide-to-grunt-redux/>
2. Сапегин А. Grunt 0.4: система сборки для фронтенд-разработчиков. — <http://nano.sapegin.ru/all/grunt-0-4>

3. Grunt: Getting started. — <http://gruntjs.com/getting-started>
4. Grunt: plugins. — <http://gruntjs.com/plugins>
5. Пестов И.С. Самые нужные плагины для Grunt. — <http://habrahabr.ru/post/251157/>
6. Travis Maynard. Gulp — как глоток свежего воздуха после Grunt. — Frontender Magazine. — <http://frontender.info/no-need-to-grunt-take-a-gulp-of-fresh-air/>
7. Nicolas Bevacqua. Gulp или Grunt, да всё равно. — Frontender Magazine. — <http://frontender.info/gulp-grunt-whatever/>
8. Gulp documentation. — <https://github.com/gulpjs/gulp/tree/master/docs>
9. Gulp.js plugin registry. — <http://gulpjs.com/plugins/>
10. Пестов И.С. Самые нужные плагины для Gulp. — <http://habrahabr.ru/post/252745/>
11. Set Up Your Build Tools. — Google Developers. — <https://developers.google.com/web/tools/setup/workspace/setup-buildtools>
12. Addy Osmani. Grunt- и Gulp-таски для оптимизации производительности. — Frontender Magazine. — <http://frontender.info/performance-optimization>