

Tools for JavaScript Development

Unleash the Power of JavaScript Tooling

Telerik Software Academy

Learning & Development

<http://academy.telerik.com>

- ◆ JavaScript Editors
 - ◆ JetBrains WebStorm
 - ◆ Sublime Text 2/3
- ◆ Project tools
 - ◆ Package Management: NPM & Bower
 - ◆ Scaffolding: Yeoman
 - ◆ Task Automation: Grunt

JavaScript Editors

- ◆ JavaScript IDE suitable for both client-side JavaScript and server (Node.js) development
 - ◆ Livereload
 - ◆ Build
 - ◆ Code highlighting/Intellisense for:
 - ◆ LESS/SASS/Stylus CSS preprocessors
 - ◆ CoffeeScript/TypeScript JavaScript preprocessors
 - ◆ Emmet Coding
 - ◆ Has only 30-days-long free trial

JetBrains WebStorm

Live Demo

- ◆ Editor for scripting languages
 - ◆ JavaScript, PHP, Python, Ruby, etc...
 - ◆ Basic code highlighting with vanilla installation
 - ◆ Endless number of configurable plugins
 - ◆ Free product
 - ◆ Paid only if used commercially
 - ◆ With plugins support for:
 - ◆ LESS/SASS/Stylus
 - ◆ CoffeeScript/TypeScript

Sublime Text 2/3

Live Demo

Project Tools

No matter the Editor

- ◆ NPM & Bower
 - ◆ Install Node.js packages or client libraries
- ◆ Grunt
 - ◆ Tasks runner
 - ◆ Create different tasks for build/development/test cases
- ◆ Yeoman
 - ◆ Scaffolding of applications
 - ◆ One-line-of-code to create a project template with views/routes/modules/etc...

Package Management

Bower & NPM

Package Management: NPM

- ◆ Node.js Package Management (NPM)
 - ◆ Package manager for Node.js modules
 - ◆ \$ npm init in CMD (Win) or Terminal (MAC/Linux)
 - ◆ Initializes an empty Node.js project with package.json file

```
$ npm init
//enter package details
name: "NPM demos"
version: 0.0.1
description: "Demos for the NPM package management"
entry point: main.js
test command: test
git repository: http://github.com/user/repository-name
keywords: npm, package management
author: doncho.minkov@telerik.com
license: BSD-2-Clause
```

Package Management: NPM (2)

◆ Installing modules

- ◆ \$ npm install package-name [--save-dev]
 - ◆ Installs a package to the Node.js project
 - ◆ --save-dev suffix adds dependency in package.json

```
$ npm install express --save-dev  
$ npm install jade --save-dev
```

◆ Before running the project

- ◆ \$ npm install
 - ◆ Installs all missing packages from package.json

```
$ npm install
```

Package Management: NPM

Live Demo

Package Management: Bower

- ◆ Bower is a package management tool for installing client-side JavaScript libraries
 - ◆ Like jQuery, KendoUI, AngularJS, etc...
 - ◆ It is a Node.js package and should be installed first

```
$ npm install -g bower
```

- ◆ \$ bower init in CMD (Win) or Terminal (Mac/Linux)

```
$ bower init
```

- ◆ Asks for pretty much the same details as \$ npm init
- ◆ Creates bower.json file to manage libraries

Package Management: Bower (2)

◆ Searching for libraries

```
$ bower search kendo
```

Search results:

```
kendo-ui git://github.co  
angular-kendo-ui git://g  
kendo-backbone git://git  
knockout-kendo git://git  
kendo-global git://githu
```

```
bower cached git://gith  
bower validate 2014.1.318  
i.git#*  
bower cached git://gith  
bower validate 2.0.3 again  
.0.3  
bower resolution Removed un  
bower install kendo-ui#2.0  
bower install jquery#2.0  
  
kendo-ui#2014.1.318 bower_com  
└─ jquery#2.0.3  
  
jquery#2.0.3 bower_components\
```

◆ Installing libraries

```
$ bower install kendo-ui
```

Package Management: Bower

Live Demo

Grunt



Tasks Runner

- ◆ Grunt is a Node.js task runner
 - ◆ It can runs different tasks, based on configuration
 - ◆ Tasks can be:
 - ◆ Concat and minify JavaScript/CSS files
 - ◆ Compile SASS/LESS/Stylus
 - ◆ Run jshint, csshint
 - ◆ Run Unit Tests
 - ◆ Deploy to Git, Cloud, etc...
 - ◆ And many many more

- ◆ Why use a task runner?
 - ◆ Task runners gives us automation, even for different profiles:

DEVELOPMENT

jshint
stylus
csshint
connect
watch

TEST

jshint
stylus
csshint
mocha

BUILD

jshint
stylus
csshint
concat
uglify
copy
usemin

Configuring Grunt

- ◆ To configure grunt, create a `Gruntfile.js` file in the root directory of your application
 - ◆ It is plain-old Node.js
 - ◆ Grunt is configured programmatically
 - ◆ Create an module that exports a single function with one parameter – the grunt object

```
module.exports = function (grunt) {  
  //configure grunt  
};
```

Configuring Grunt (2)

- ◆ All the configuration is done inside the module
 - ◆ First execute the `grunt.initConfig()` method and pass it the configuration

```
module.exports = function (grunt) {  
  grunt.initConfig({  
    ...  
  });  
};
```

Configuring Grunt Plugins

- ◆ To use a plugin in grunt:
 - ◆ Install the plugin

```
$ npm install grunt-contrib-jshint --save-dev
```

- ◆ Load the plugin

```
//inside the grunt module
grunt.loadNpmTasks('grunt-contrib-jshint');
```

- ◆ Configure the plugin

```
//inside the grunt.initConfig()
grunt.initConfig({
  jshint: {
    app: ['Gruntfile.js', 'path/to/scripts/**/*.*js']
  }
});
```

Configuring Grunt Plugins

Live Demo

Grunt Plugins

Grunt Plugins: Build

- ◆ **jshint (grunt-contrib-jshint)**
 - ◆ Runs jshint for specified files
- ◆ **csslint(grunt-contrib-csshint)**
 - ◆ Runs csslint for specified files
- ◆ **stylus (grunt-contrib-stylus)**
 - ◆ Compiles STYL files into CSS files
- ◆ **uglify (grunt-contrib-uglify)**
 - ◆ Minifies configured JavaScript files
- ◆ **concat (grunt-contrib-concat)**
 - ◆ Concats configured JavaScript files

Grunt Plugins for Build

Live Demo

Grunt Plugins: Development

- ◆ **connect (grunt-contrib-connect)**
 - ◆ Starts a Web server on a given port and host
- ◆ **watch (grunt-contrib-watch)**
 - ◆ Watches for changes to configured files
 - ◆ Can run other tasks on file changed

Grunt Plugins for Development

Live Demo

Yeoman

Application Scaffolding

- ◆ Yeoman is a Node.js package for application scaffolding
 - Uses bower & NPM to install the js package
 - Has lots of generators for many types of applications:
 - MEAN, AngularJS, Kendo-UI, WebApp, WordPress, Backbone, Express, etc...
 - Each generators install both needed Node.js packages and client-side JavaScript libraries
 - Generated Gruntfile.js for build/test/serve

- ◆ Install Yeoman:

```
$ npm install -g yo
```

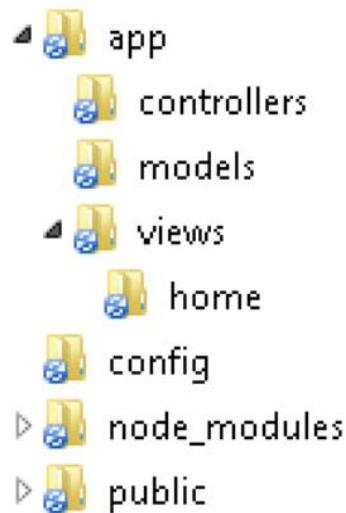
- ◆ Install Yeoman generator:

```
$ npm install -g generator-express
```

- ◆ Scaffold Express application:

```
$ cd path/to/app/directory  
$ yo express
```

- ◆ Generates:



Yeoman Scaffolding

Live Demo

Tools for JavaScript Development

Questions?

1. Create a project with a Gruntfile.js and three folders – DEV folder, APP folder and DIST folder

- Write some CoffeeScript, Jade and Stylus in APP
- Register the following grunt tasks:
 - Serve:
 - Compiles the CoffeeScript to JS and puts them into DEV/scripts
 - Runs jshint on the compiled JS files
 - Compiles the Jade to HTML and puts them into DEV
 - Compiles the Stylus to CSS and puts them into DEV/styles
 - Copies every image from the APP/images to DEV/images
 - Connect a server on port 9578 and show the contents of DEV
 - Watch for changes to the CoffeeScript, Stylus and Jade files, and if changed – reload the page into the browser

Homework (2)

1. *cont. Create a project with a Gruntfile.js and three folders – DEV folder, APP folder and DIST folder
 - Write some CoffeeScript, Jade and Stylus in APP
 - Register the following grunt tasks:
 - Build:
 - Compiles CoffeeScript, Stylus and Jade
 - Runs jshint and csslint
 - Concats all CSS files into one file and minify it into DIST/styles
 - Concats all JS files into one file and uglify it into DIST/scripts
 - Uglifies the HTML files into DIST
 - Copies all images into DIST/images