Software Automation using Yeoman

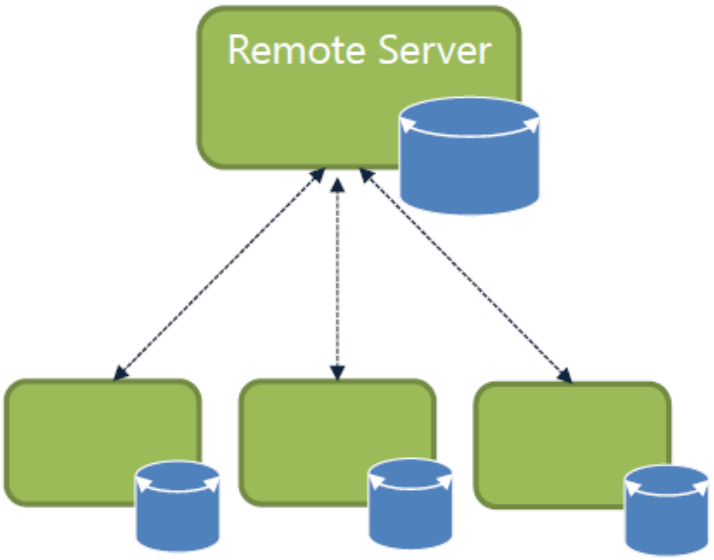
Frank Walsh, Diarmuid O’Connor

Agenda

- Software Package Management
  - Node Package Management
  - Bower
- Automation and Build
  - Grunt
- Scaffolding
  - Yeoman

Aside: git

- Distributed Version Control
- Directory Content Management
- Tree Based History
- Everybody has complete history
- Perhaps becoming seminal code management tool
  - Provides basis for other tools such as Phonegap/cordova, Node, NPM and Bower



Node Package Manager()

- Software package manager for most web/javascript projects
  - Jquery, AngularJS, …
- Install from https://www.npmjs.com/
- Can create a node project using:
  - npm init on command line/terminal
  - 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
```

NPM

- Installing Modules is easy

```
npm install package-name [--save-dev]
```

- Installs a package to the Node.js project
  - "--save-dev" suffix adds dependency to package.json

```
npm install express --save-dev
```

- Before running a proje

```
npm install
```

Installs all missing packages from package.json

Package Management: Bower

- Bower is a package management tool for installing client-side JavaScript libraries
  - E.g. Jquery, AngularJS, Backbone, …
  - Requires Node and NPM.
- Installed as follows:

```
npm install -g bower
```

"-g" suffix installs module in a global context (i.e. can run from command line)

- Initialisation very similar to NPM

  bower init
- Asks for project details and creates a "bower.json" file to manage dependencies
  - Gets you to follow best practices with versioning, author etc.

Package Management: Bower

- To install a package, say jquery:

  bower install jquery
- To search for a common package/library, e.g. bootstrap:

  bower search bootstrap
- Other good commands include:
  - bower list …list all dependencies
  - Bower update …update dependencies from source.

- Good tutorial here

Task Management

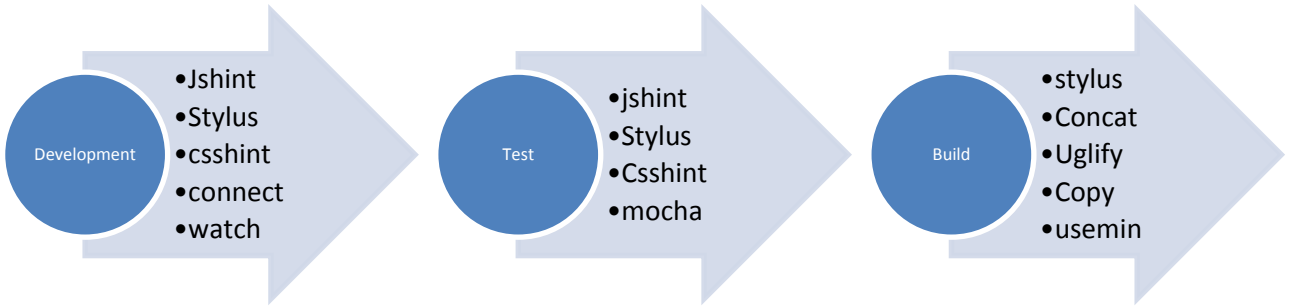
GRUNT

Task Management: Grunt

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

Task Management: Grunt

- Why use Task Manager:
  - Automate routine tasks
  - Save time and make less mistakes.
  - Can apply different automation profiles for various stages of software dev process



Grunt: possible tasks

- Jshint
  - tool that helps to detect errors and potential problems in JavaScript code
- Stylus
  - efficient, dynamic, and expressive way to generate CSS
- Csshint
  - tool that helps to detect errors and potential problems in css.
- Connect
  - ability to run a web server on a local file system and interact with the files using a web browser
- watch
  - when it detects any of the files specified have changed, it will run the tasks you specify, in the order they appear.
- Concat
  - Concatinates all js files into one file. Often followed by uglify.
- Uglify
  - minify JavaScript files.

Configuring Grunt

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

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

Configuring Grunt

- 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({
    ...
  });
};
```

YEOMAN

Yeoman

- Node.js package for application scaffolding and workflows
- Key components:
  - Yo
  - Bower
  - Grunt
- Many build in generators for common types of applications
    - AngularJS, MEAN, Express…
    - Each generators install both needed Node.js packages and client-side JavaScript libraries
    - Generated Gruntfile.js for build/test/serve
    - Takes care of structuring application

Yo Generators

- Node Modules
- Available from NPM

  npm search some-generator
- Can write your own custom generators also

Yeoman Installation

- To Install:

```
npm install –g yo
```

- To Install a generator:

```
npm install –g generator-express
```

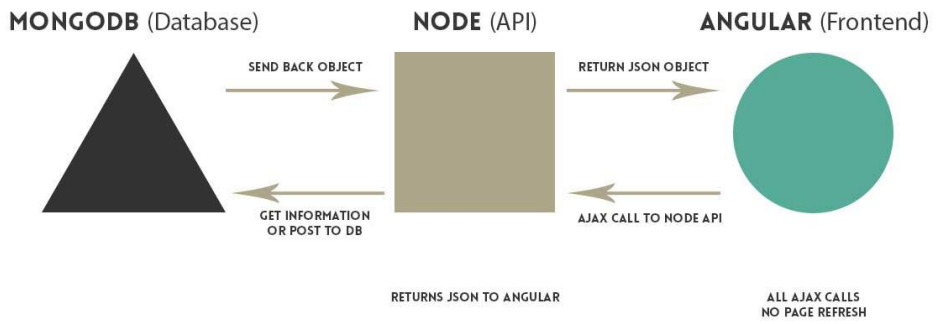
- Create a scaffold for express application(run this in the application top level directory)

```
yo express
```

DEMO

Angular Application with Express

- We will use a Yeoman generator for creating MEAN stack applications, using MongoDB, Express, AngularJS, and Node
- Lets you quickly set up a project following best practices.
- Details are here



Initial Set up

1. Install git on your machine

   See here
2. Install MongoDB on your machine

   See here to install and start a Mongodb instance on your machine.
3. Install Yo, Grunt and Bower:

   npm install –g yo grunt-cli bower
4. Install the generator that you require:

   npm install -g generator-angular-fullstack
5. Create a new project folder called **hackerMongo** and change directory to it

Scaffold your Project

- To create the '*vanilla*' scaffold, run the following in the project folder:
  yo angular-fullstack hacker
- You'll be asked for some configuration and build details. Answer as follows:
  - Scripts: Javascript
  - Markup: HTML
  - CSS: CSS
  - Routing: ngRouting
  - Bootstrap: Yep
  - MongoDB/Mongoose:Yep
- Say **no** to everything else…

Examine what's generated

- Server and client with skeleton web page and web api.
- Gruntfile includes the following tasks/plugins
  - Watch
  - Test
  - Build
    - Generates web app in dist folder.
  - serve

Configuring MongoDB

- You can configure the location of the DB you want to use for each workflow:
  - i.e. for development, ideally use local
  - Can specify for different “targets”. You will have a different DB for production
  - Configured in server/config/environment

Web API Scaffolding

- Generate a generic scaffolding for the Web API post endpoint:

```
yo angular-fullstack:endpoint post
```

- Check out generated resources in /server/api/post/…
- Replace model with “post” model from previous module

Seeding

- Just as in previous labs, you can seed the database in development/testing.
- Open server/config/seed.js and enter the seed data…
- Test the endpoint.

Generate Client

- generate the posts route scaffolding:

```
yo angular-fullstack:route posts
```

- This generates new view and associated routing

Update Client Files

- Update the following client files
  - client/app/posts/posts.html
  - client/app/app.css
  - client/app/posts/posts.controller.js

Scaffold a Service

- Create a service scaffold:

  ```
  yo angular-fullstack:factory Post
  ```

- `Update c`lient/app/Post/Post.service.js and corresponding controller.