Community Experience Distilled

Learning Yeoman

Design, implement, and deliver a successful modern web application project using three powerful tools in the Yeoman workflow

Jonathan Spratley

[PACKT] PUBLISHING open source* community experience distilled

Learning Yeoman

Design, implement, and deliver a successful modern web application project using three powerful tools in the Yeoman workflow

Jonathan Spratley

[PACKT] PUBLISHING open source community experience distilled

BIRMINGHAM - MUMBAI

Learning Yeoman

First published: August 2014

Production reference: 1120814

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-78398-138-0

www.packtpub.com

Cover image by Tony Shi (shihe99@hotmail.com)

Credits

Author
Jonathan Spratley

Reviewers
Lauren Bridge
Dejan Markovic
Matt Momont

Commissioning Editor
Edward Gordon

Acquisition Editors
Sam Birch
Ellen Bishop

Content Development Editor
Rikshith Shetty

Technical Editor
Neha Mankare

Copy Editors
Janbal Dharmaraj
Karuna Narayanan
Alfida Paiva

Project Coordinators
Melita Lobo
Kartik Vedam

Proofreaders
Simran Bhogal
Lauren Harkins
Lawrence Herman
Joanna McMahon

Indexers
Hemangini Bari
Tejal Soni

Graphics
Ronak Dhruv

Production Coordinator
Arvindkumar Gupta

Cover Work
Arvindkumar Gupta

About the Author

Jonathan Spratley, or as his friends call him, Jonnie, is currently working at GE Software and actively uses technologies such as Yo, Bower, Grunt, Node, AngularJS, and Backbone. He leveraged his knowledge of creating Yeoman web applications to write this book. He started developing with HTML, CSS, and JavaScript around 2004 and has spent several years designing and implementing web applications for both mobile and desktop browsers. He is a full-stack developer with experience in both server- and client-side technologies. His passion for tools and technologies that streamline a developer's productivity has driven him to become an autodidact. He has written articles for Safari Books Online Blog, and Flash & Flex Developer's Magazine. This is his first time as an author.

I would like to dedicate this book to all those who believed in me.

About the Reviewers

Lauren Bridge is a software engineer on the Predix platform team at GE Software. She earned her BS in Computer Science from the University of Michigan and is working on her MCS at the University of Illinois at Urbana-Champaign. She mostly develops in JavaScript and Java, and she enjoys learning more frontend web technologies every day.

I would like to thank Jonnie for teaching me about Yeoman.

Dejan Markovic is an accomplished web developer who enjoys working on both frontend and backend technologies. Since 2003, he has added great value to numerous projects for small- and medium-sized businesses as well as major corporations such as Rogers Media and Softchoice. He is the co-owner of NYTO Group (New York City/Toronto Group), a premium web development company in Toronto, Canada.

NYTO Group's portfolio is available at `http://nytogroup.com/portfolio/`. NYTO Group is always looking for new projects and partnerships.

Matt Momont is a full-stack developer at GE Software working on the Industrial Internet—the Internet of (really big) Things. He holds a Computer Science degree from the University of Notre Dame and is currently pursuing his Masters in Computer Science at the University of Illinois at Urbana-Champaign. He likes Yeoman because it helps backend developers quickly scaffold out frontend web applications.

www.PacktPub.com

Support files, eBooks, discount offers, and more

You might want to visit `www.PacktPub.com` for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at `www.PacktPub.com` and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at `service@packtpub.com` for more details.

At `www.PacktPub.com`, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.

`http://PacktLib.PacktPub.com`

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print and bookmark content
- On demand and accessible via web browser

Free access for Packt account holders

If you have an account with Packt at `www.PacktPub.com`, you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

Table of Contents

Preface

Now is the time to start using a workflow that can keep up with the fast pace of the development world. Software changes so fast that keeping your project libraries updated and using the latest code has always been a manual, tedious process. Well, not anymore, thanks to modern tooling that has taken my development productivity to greater levels! I have been using Yeoman since the early versions, where Yeoman was the one tool that could do it all.

Since the Yeoman project grew, it has evolved into something I have always wanted in the web development community, such as code generators that can quickly scaffold out working applications that are in alignment with the best practices of that specific framework or language. Now, the time has come and Yeoman is going to take the development world by storm and grow into something that will become a standard in creating modern web applications.

This book is a compilation of using the most popular Yeoman generators on npm. We explore the options that each tool has to offer and use them to create various types of projects, ranging from AngularJS applications to Node.js modules. This book provides examples and information regarding the tools in Yeoman.

What this book covers

Chapter 1, *Modern Workflows for Modern Webapps*, is an overview of the three core tools used in the Yeoman workflow—Yo, Bower, and Grunt. We cover how to use these tools in development and how to incorporate the workflow into new or existing projects, followed by an example of each of the features in Yeoman.

Chapter 2, *Getting Started*, begins with installing Yeoman for development and an overview on the AngularJS, Backbone.js, Ember.js, and webapp generators, the options and subgenerators it uses, and examples of using each generator to start the project.

Chapter 3, *My Angular Project*, starts out with covering the concepts of Angular and the anatomy of an AngularJS application. We will use the generator-angular to scaffold an extendable AngularJS application that uses directives, services, and factories. We will cover setting up a CRUD application with unit tests that use the Karma runner.

Chapter 4, *My Backbone Project*, covers the anatomy of a Backbone.js project and the concepts behind the library. We create a Backbone application to perform CRUD operations on a data source that is unit tested using Jasmine. The project uses CoffeeScript, Require.js, and AMD to create a well-structured app ready for extending.

Chapter 5, *My Ember Project*, starts out by creating a new Ember.js project. We then cover how an Ember application is structured and the concepts around the framework, configuring a test environment that is used to run both unit and integration tests.

Chapter 6, *Custom Generators*, covers the Yeoman generator API and the common methods used when developing generators. We also cover installing and invoking the generator-generator to create a custom Yeoman generator with option prompts that scaffold a custom application based on users' feedback. We cover how to handle testing the generators using nodeunit and then we publish the generator to npm.

Chapter 7, *Custom Libraries*, covers using Yeoman to create custom libraries that are deployed to either Bower or npm. We learn how to use the Node.js generator and the CommonJS generator to create a Node module, followed by a client-side jQuery plugin that handles sending CRUD operations to a Node REST API server.

Chapter 8, *Tasks with Grunt*, starts out by covering all the available options when using the Grunt command. We install two Yeoman Grunt generators: the Gruntfile generator that enables adding Grunt to existing projects, and the Grunt plugin generator. We cover creating a custom Grunt task that is then deployed to npm along with unit tests using the nodeunit framework.

Chapter 9, *Yeoman Tips and Tricks*, aims to cover the holes from the Yeoman generators and specific projects. We cover adding code coverage to a Backbone.js project, as well as setting up Protractor to run end-2-end testing for our Angular project.

What you need for this book

You will need to have the following installed software on your development box in order to properly run the examples and tutorials in the chapters:

- Node 0.10.24
- NPM 1.4.7
- Git 1.8.5.2
- Ruby 1.9.2
- Text editor of some sort
- Google Chrome

Who this book is for

This book is for newbies and intermediate web developers looking to speed up the process when it comes to creating web applications of various frameworks. You should have basic knowledge of HTML, CSS, and JavaScript. The examples in this book will use jQuery style selectors and methods, so some jQuery experience is needed. The tools in this book involve the command line, so having basic knowledge about using the shell to invoke commands on a system is required. As long as you understand basic principles about structuring HTML and OO JavaScript applications, you should have no problem following the step-by-step examples in the chapters.

Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "The `bower.json` file is how Bower manages the project's dependencies."

A block of code is set as follows:

```
<div class="header">
  <ul class="nav nav-pills pull-right">
    <li ng-repeat="item in App.menu"
      ng-class="{'active': App.location.path() === item.href}">
      <a ng-href = "#{{item.href}}"> {{item.title}} </a>
    </li>
  </ul>
  <h3 class="text-muted"> {{ App.sitetitle }} </h3>
</div>
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
require 'scripts/config'
LearningYeomanCh5 = window.LearningYeomanCh5 =
  Ember.Application.create(
  LOG_VIEW_LOOKUPS: true
  LOG_ACTIVE_GENERATION: true
  LOG_BINDINGS: true
  config: window.Config
)
```

Any command-line input or output is written as follows:

```
$ npm install -g generator-webapp
```

New terms and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "See for yourself; open Chrome Developer Tools and click on the **Network** tab."

Warnings or important notes appear in a box like this.

Tips and tricks appear like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to `feedback@packtpub.com`, and mention the book title via the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on `www.packtpub.com/authors`.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the example code

You can download the example code files for all Packt books you have purchased from your account at `http://www.packtpub.com`. If you purchased this book elsewhere, you can visit `http://www.packtpub.com/support` and register to have the files e-mailed directly to you.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting `http://www.packtpub.com/submit-errata`, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from `http://www.packtpub.com/support`.

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at `copyright@packtpub.com` with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

Questions

You can contact us at `questions@packtpub.com` if you are having a problem with any aspect of the book, and we will do our best to address it.

1
Modern Workflows for Modern Webapps

This chapter will cover the three core tools that make up the Yeoman workflow, how to use these tools in development, and how to incorporate this workflow into new or existing projects.

In this chapter, you will learn the following topics:

- The Yeoman tools and architecture
- Downloading and installing Yeoman
- Features of Yeoman
- Using the Yeoman tools

An overview of Yeoman

The term **modern webapps** is a relatively new thing, as the Web is still in its infancy stage. As the Web matures, so does the need for developer tools and workflows, thanks to some modern-day Web pioneers over at Google. Paul Irish and Addy Osmani have developed a modern workflow that goes by the name of **Yeoman**.

The Yeoman workflow is a collection of three tools to improve developers' productivity when building web applications: **Yo** is the scaffolding tool, **Grunt** is the build tool, and **Bower** is the package tool.

- Yo is used to scaffold things such as projects and files from templates
- Grunt is used for task management, testing, code linting, and optimization
- Bower is used for package management and to manage client-side dependencies

Yeoman's architecture

The Yeoman toolset runs in the Node.js environment and is invoked from the command line. Each tool is installed using **Node's package manager** (**npm**) and uses the npm repository to manage all plugins.

Node's package manager

Node.js is a platform that is built on Chrome's JavaScript runtime engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight, efficient, and perfect for real-time applications that run across distributed devices.

The official package manager for Node.js is npm. From Node versions 0.6.3 and up, npm is bundled and installed automatically with the environment. The npm package manager runs through the command line and manages the application dependencies that are available on the `npm` registry.

The current Node.js version used in this book is v0.10.28.

Features of Yeoman

Before we dig deep into using each tool of the workflow, let's take a look at some of the Yeoman tooling features that will help you in your next project:

- **Quick install**: Easily installs all three tools from the npm repository using one command
- **Scaffolding**: Fast and easy-to-use command-line tool to create new projects or files from templates that individual generators provide
- **Build process**: Tasks for concatenation, minification, optimization, and testing
- **Preview server**: Connect LiveReload server to preview your application in the browser
- **Package management**: Search, install, and manage project dependencies via the command line
- **Code linting**: Scripts are run against JSHint to ensure language best practices
- **Automation**: A simple watch task to compile CoffeeScript, LESS, or SASS, and reload the browser upon changes

- **Testing**: Executes JavaScript code in multiple real browsers with the Karma runner
- **Optimization**: Images are optimized using OptiPNG and JPEGtran, and HTML is optimized using the HTML minifier

The preceding features are dependent on what the individual generators provide via Grunt tasks. By default, the Angular, Backbone, Ember, and other webapp generators provide tasks to perform all the features listed.

Grunt tasks are individual plugins that perform specific operations on files or folders.

Quick installation

Modern tools usually mean more tools to learn, but learning the tools of the Yeoman workflow is easier than you think. To demonstrate by example, here is how easy it is to get a modern web application up and running, all from the command line.

Installing Yeoman and friends

To install all three tools in the Yeoman workflow, just execute the following command in the terminal:

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

The command will install Yo, Grunt, and Bower into your systems path as follows:

- The `-g` option flag specifies the installation to be globally available in your path, allowing the `yo` command to be invoked from anywhere
- If using the latest versions of Node and Git, Yeoman will automatically install Bower and Grunt while installing Yo

The `-g` flag installs globally and requires an administrator user.

If you run into any errors during the initial installation process, you can install the `envcheck` module to ensure that your system is ready for all of Yeoman's features; just execute the following command:

```
$ npm install -g envcheck
```

Installing a generator

To install generators for Yo, use `npm`. Let's install the generic webapp generator; open a terminal and execute the following command:

```
$ npm install -g generator-webapp
```

The preceding command will install the webapp generator globally on your system, easily letting you create new web projects within any directory of your choice.

Scaffolding with Yo

Yeoman includes a powerful command-line utility that can scaffold files based on individual generator templates, allowing you to save time creating files from scratch. There are over 700 community generators on npm.

- To search for generators, add the `generator-` prefix before the name, as follows:

  ```
  $ npm search generator-[name]
  ```

- To install generators, use `npm install` passing in the name of the package, as follows:

  ```
  $ npm install generator-[name]
  ```

The `npm` attribute is the package manager for Node.js and comes bundled with it.

Creating the project

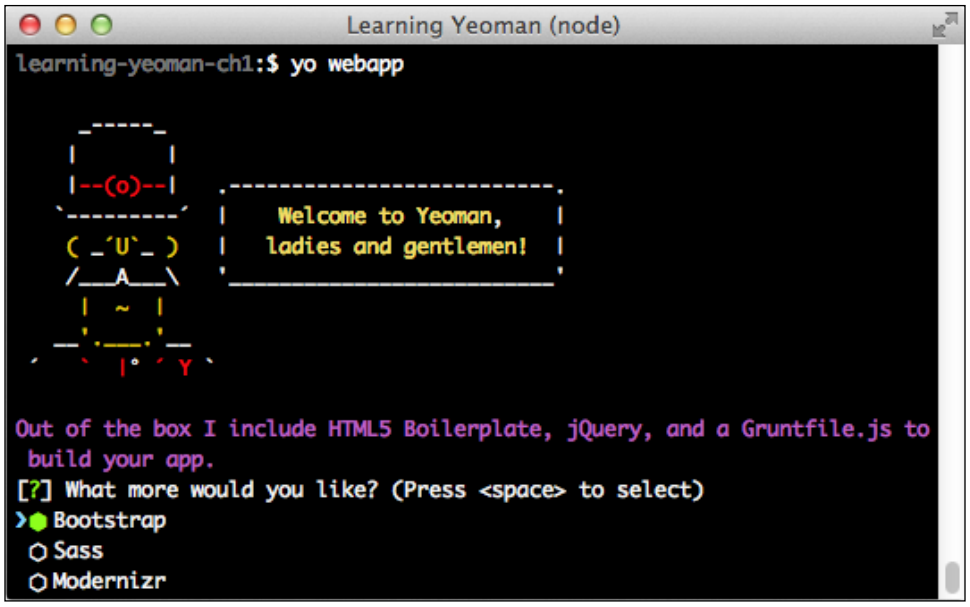
All Yeoman commands work off the current directory, so create a new folder named `learning-yeoman-ch1`, open the terminal, and `cd` to that location into the newly created directory.

Invoking the generator

Yo, the scaffold tool, will easily create and set up project configuration, files, and folders needed for a modern web application. Execute the following command:

```
$ yo webapp
```

Generators can be invoked with different options; in the preceding command, we use the generators' default options that include Mocha as the test framework and JavaScript as the scripting language. You will get an output similar to the following screenshot:



The preceding command does many things. First off, it's going to ask you a few questions about your new project, such as whether to include Twitter Bootstrap with or without Compass SASS and whether to include Modernizr.

Select the first option (Bootstrap), and press *Enter*; you will see the output to the terminal, and Yeoman is performing all the magic right before your eyes.

Directory structure

Do not be overwhelmed by the number of files generated; take a minute and examine the directory structure that Yeoman produces. You will notice how organized the directory structure is. It looks as follows:

```
├── Gruntfile.js
├── app
|   ├── 404.html
|   ├── bower_components
```

```
|   |   ├── bootstrap
|   |   └── jquery
|   ├── favicon.ico
|   ├── images
|   ├── index.html
|   ├── robots.txt
|   ├── scripts
|   |   └── main.js
|   └── styles
|       └── main.css
├── bower.json
├── node_modules
├── package.json
└── test
    ├── bower.json
    ├── bower_components
    ├── index.html
    └── spec
```

Just think of Yeoman as a helpful robot that does all the hard work for you, creating all the necessary files and folders to get started with development.

The build process

Yeoman includes Grunt, a task-based command-line tool for JavaScript projects. It is used to perform various build tasks on projects and exposes several useful tasks that you will want to use in your workflow. Yeoman automatically creates and configures `Gruntfile.js`, which specifies the configuration of the tasks and targets.

The following order of commands is used for a seamless development workflow:

```
$ yo webapp                  #scaffold application
$ bower install jquery       #install dependency
$ grunt serve                #start preview server
$ grunt test                 #run unit tests
$ grunt                      #create optimized build
```