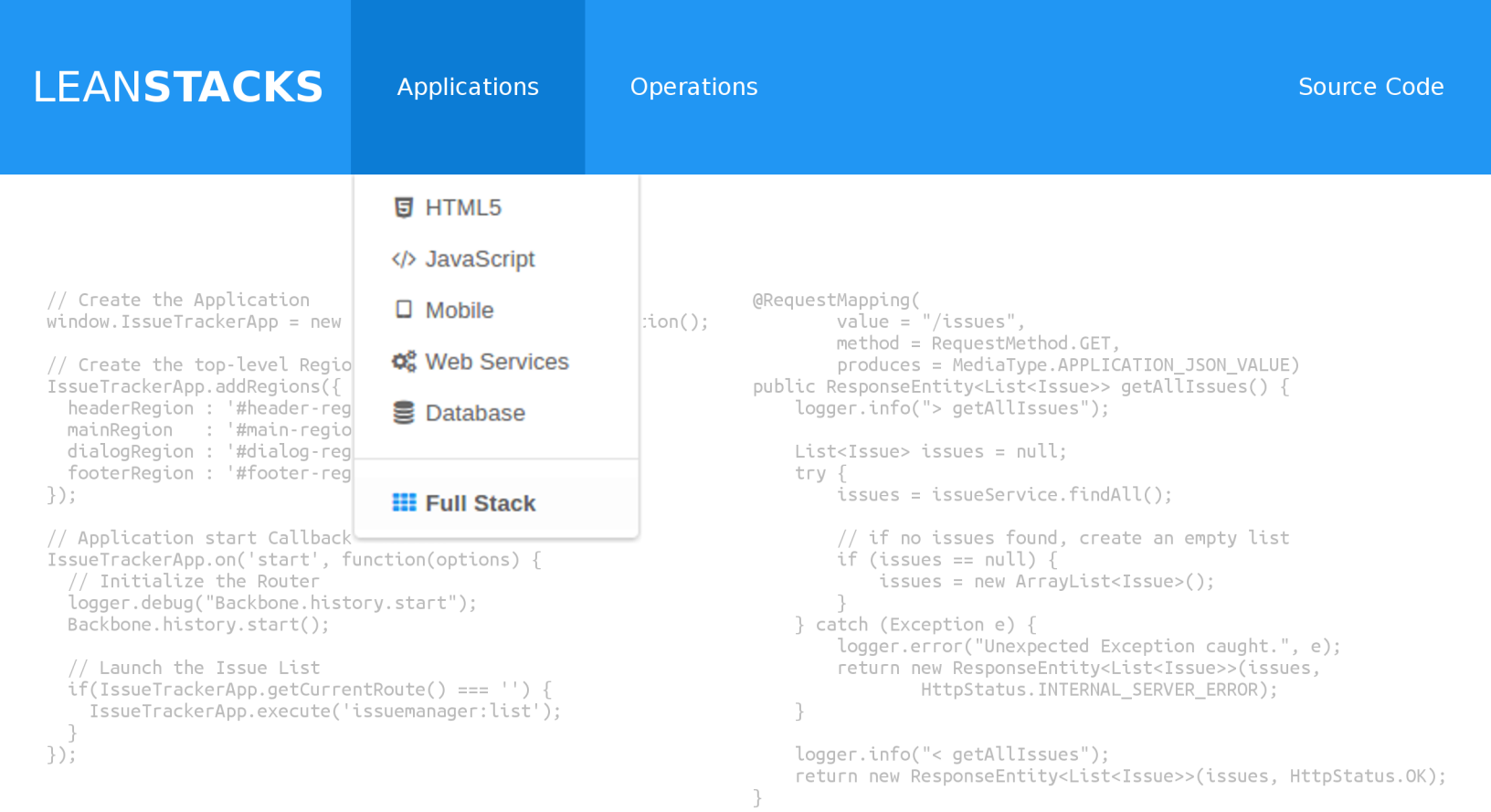

Lean Application Engineering featuring Backbone.Marionette.js and the Spring Framework

Learn lean application engineering techniques building a single page web application and RESTful web services using Backbone.Marionette.js and the Spring Framework.

by **Matthew Warman**

Lean Application Engineering

Featuring Backbone.Marionette and the Spring Framework

Matthew Warman

This book is for sale at http://leanpub.com/leanstacks-marionette-spring

This version was published on 2015-06-29

Tweet This Book!

Please help Matthew Warman by spreading the word about this book on Twitter!

The suggested hashtag for this book is #leanstacks.

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

https://twitter.com/search?q=#leanstacks

Contents

Part I. Introduction

Paradigm Shift

In the 1990's savvy marketing and business executives challenged information technology teams to move their mainframe and desktop applications to the Internet. Technology executives soon realized the benefits of a browser-based user interface. Desktop and mainframe user interfaces were tightly coupled to the back-end business logic. Releasing a software update inevitably required changes to the user interface. Desktop user interfaces were often deployed to thousands of client machines. Rolling out updates to non-browser-based user interfaces was tremendously costly and complex. Moving the user interface to the browser nearly eliminated the costly software upgrade process associated with traditional user interfaces.

Throughout the 1990's open-source technologies began emerging that serve as the foundation for web applications. In 1995 Netscape introduced JavaScript which allowed programmers to perform basic tasks such as form input validation or dynamically show and hide portions of HTML pages. In 1999, the Java introduced a Servlet specification revision which ushered in the concept of the server-side web application.

For the next decade, server-side web applications replaced the desktop client as the standard user interface technology. Server-side web applications are characterized by their presentation logic residing on the server that hosts the user interface. When a user performs an action in their browser, the browser submits a request to the server and blocks the user from taking further action. The server processes the request and creates a new HTML page which it returns to the user's browser. The browser displays the new page provided by the server.

In the mid-2000's, Asynchronous JavaScript And XML, or AJAX, technologies had matured and were introduced into many commercial web applications. By the late 2000's and into the 2010's a new trend in web applications emerged, the Single Page Application, or SPA. In a Single Page Application, the presentation layer logic is created entirely in a client-side scripting language like JavaScript. When a user accesses a SPA, their browser loads the application in one HTML page, hence the name Single Page Application. As the user interacts with the application, the client-side script performs the behavioral logic that previously required a round trip to the server to complete. When information needs to be acquired from or submitted to the server, AJAX is used to send the requests to the server asynchronously. Unlike a server-side web application, the page does not block the user from continuing to use the web application. The asynchronous nature of AJAX allows the browser to communicate with the server in the background while the user continues to interact with the web application.

In a Single Page Application, the server-side component is no longer responsible for presentation behavior; however, it still receives HTTP requests and responds to them. The server-side component in a SPA application stack consists of web services that implement the business logic for the resources they manage, decoupling them from a specific client. No longer responsible for presentation logic, these web services are completely reusable throughout the technology ecosystem.

Industry standard technologies are evolving and a new wave of frameworks are emerging that propel the Single Page Application application technology stack to the forefront of web application design paradigms.

About the Book

In this book, two technology frameworks are employed to construct a full stack Single Page Application with RESTful web services. The Spring Framework has been a popular, reliable, and feature rich Java technology for more than a decade. The web services component of the application is constructed using the Spring Boot project, leveraging the core Spring Framework and the Spring Data projects. Readers will create the Single Page Application client-side user interface component using the popular Backbone and Backbone.Marionette JavaScript libraries.

This book is a hands-on tutorial, providing step-by-step instructions beginning with setting up the software projects and finishing with a working application that performs create, read, update, and delete operations on a database. As you progress through the book, you will learn to use various popular client-side libraries including: Backbone.js[1], Marionette.js[2], Bootstrap[3], and jQuery[4]. The examples within the book are authored in the JavaScript language, but the full application is available in CoffeeScript[5] as well. As you construct the web services, you will learn to use the core Spring Framework[6], Spring Boot[7], and Spring Data[8] for both JPA and MongoDB database persistence. After completing this book, you will be well on your way to becoming a full stack software engineer.

The book focuses on the development of the user interface and web services; however, the *Production Features* section will help you prepare for deployment to servers. The chapters offer solutions for continuous integration, environment-specific configuration, production databases like MySQL and MongoDB, and monitoring and management of the application.

[1] http://backbonejs.org/
[2] http://marionettejs.com/
[3] http://getbootstrap.com/
[4] http://jquery.com/
[5] http://coffeescript.org/
[6] http://projects.spring.io/spring-framework/
[7] http://projects.spring.io/spring-boot/
[8] http://projects.spring.io/spring-data/

Intended Audience

This book is intended for software engineers, architects, and enthusiasts. The reader is empowered to create modern, single-page user interface applications and the RESTful web services that serve them data. While reading this book, the concepts are illustrated by building a simple issue tracking application.

The Single Page Application, or SPA, is created using JavaScript, CSS, and HTML. It is recommended that readers have a basic understanding of the JavaScript programming language.

The RESTful web services are created using the Spring Framework in the Java programming language. It is recommended that readers have a basic understanding of the Java programming language.

The Complete Book

This book is a full-stack engineering experience. Build both a single-page application (SPA) client-side user interface and a complete set of RESTful web services. Begin by rapidly prototyping using an in-memory database. Then learn to integrate with both MySQL and MongoDB.

Ready for production? The book dedicates chapters to development operations and production operations. Learn to quickly, simply, and reliably build, package, configure, deploy, and monitor the applications in production server environments.

Part I. Introduction

Part II. Server Foundation