

Pro Single Page Application Development

Using Backbone.js and ASP.NET



Gil Fink

Ido Flatow

Apress®

Pro Single Page Application Development: Using Backbone.js and ASP.NET

Copyright © 2014 by Gil Fink and Ido Flatow

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

ISBN-13 (pbk): 978-1-4302-6673-0

ISBN-13 (electronic): 978-1-4302-6674-7

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image, we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors, nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

President and Publisher: Paul Manning

Lead Editor: Gwenan Spearing

Technical Reviewers: Konstantin Tarkus and Fabio Claudio Ferracchiati

Development Editor: Anne Marie Walker

Editorial Board: Steve Anglin, Mark Beckner, Ewan Buckingham, Gary Cornell, Louise Corrigan, Jim DeWolf,

Jonathan Gennick, Jonathan Hassell, Robert Hutchinson, Michelle Lowman, James Markham, Matthew Moodie, Jeff Olson, Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Gwenan Spearing, Matt Wade, Steve Weiss

Coordinating Editor: Christine Ricketts

Copy Editors: Michael G. Laraque and Christine Dahlin

Compositor: SPi Global

Indexer: SPi Global

Artist: SPi Global

Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a Delaware corporation.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this text is available to readers at www.apress.com. For detailed information about how to locate your book's source code, go to www.apress.com/source-code/.

*Dedicated to my lovely wife, Liora, and my wonderful kids,
Oron, Eyal, and Noya*

— Gil Fink

*To Rotem, my baby boy, may your joy know no bounds as you grow up,
just as it now fills your day*

— Ido Flatow

In memory of Noam Sheffer, a polymath, a colleague, and a dear friend

Contents at a Glance

About the Authors	xvii
About the Technical Reviewers	xix
Acknowledgments	xxi
Introduction	xxiii
■ Part I: The Road to Single Page Application Development	1
■ Chapter 1: Introducing Single Page Applications	3
■ Chapter 2: JavaScript for SPAs	15
■ Chapter 3: Modular JavaScript Development	35
■ Chapter 4: SPA Concepts and Architecture	49
■ Part II: Building the Front End	67
■ Chapter 5: Getting Started with Backbone.js	69
■ Chapter 6: Creating a Single Page Application Step by Step	99
■ Part III: Building the Back End	135
■ Chapter 7: Creating a Back-End Service with ASP.NET Web API	137
■ Chapter 8: Implementing HTTP Concepts with ASP.NET Web API	171
■ Chapter 9: Communication Between Front and Back End	187

■ Part IV: Advanced SPA Topics	205
■ Chapter 10: JavaScript Unit Testing	207
■ Chapter 11: SPA Performance Tuning	229
■ Chapter 12: Search Engine Optimization for SPAs	267
■ Chapter 13: SPA Deployment	277
Index	289

Contents

About the Authors	xvii
About the Technical Reviewers	xix
Acknowledgments	xxi
Introduction	xxiii
■ Part I: The Road to Single Page Application Development	1
■ Chapter 1: Introducing Single Page Applications	3
How the Web Has Evolved	3
The Ajax Revolution	4
HTML5 and JavaScript	5
The Mobile Web and JavaScript	5
JavaScript Preprocessors.....	5
Comparing Application Types	6
Traditional Web Applications.....	6
Native Applications	10
So What Is an SPA?	11
Why SPA Development?	13
Summary.....	13
■ Chapter 2: JavaScript for SPAs	15
A Little JavaScript History	15
Functions in JavaScript	16
The arguments Object	17
Anonymous Functions	18

Nested Functions.....	18
JavaScript Scope.....	19
The Global Scope.....	19
Function Scope.....	20
Immediate Function Execution Expressions	20
Working with Object-Oriented JavaScript.....	21
Literal Object Notation.....	21
The new Notation and Constructor Functions	22
The Object Base Type	22
The Object.create Function.....	23
JavaScript Prototype Inheritance	23
The this Keyword.....	24
Creating JavaScript Namespaces	27
Closures and Encapsulation	28
JSON.....	30
ECMAScript and Its Impact on Future Code	31
ECMAScript 5 Highlighted Features.....	31
ECMAScript 6.....	31
Summary.....	34
■ Chapter 3: Modular JavaScript Development.....	35
The Façade Pattern in JavaScript	35
Modules to the Rescue.....	36
Different Options to Create JavaScript Modules	37
Prototype Pattern.....	37
Module Pattern	38
Revealing Module Pattern	39
Revealing Prototype Pattern.....	40
What Is the Asynchronous Module Definition?	42
The async and defer Attributes.....	42
AMD	42

The RequireJS Library	43
The data-main Attribute.....	44
The config Function.....	44
Defining Modules Using RequireJS	46
Using the require Function	48
Summary.....	48
■ Chapter 4: SPA Concepts and Architecture	49
Main SPA Building Blocks	49
JavaScript Libraries	49
MV* in the Front End	51
MVC	51
MVVM.....	52
The Roles of MV* Frameworks in SPAs	52
Routing in an SPA.....	53
The # Sign Role in Routing	54
Leveraging HTML5 JavaScript APIs.....	55
Offline Web Applications and Storages.....	55
Connectivity.....	60
Performance	60
Client-Side Template Engines.....	61
Underscore as a Template Engine Example.....	61
Server Back End API and REST.....	63
SPA Architecture.....	64
Summary.....	66

■ Part II: Building the Front End	67
■ Chapter 5: Getting Started with Backbone.js	69
What Is Backbone.js?	69
Models	71
Defining a Model.....	71
Model APIs	72
Model Validation	73
Model Events	75
Views	76
Defining a View	76
Rendering the View.....	77
Collections	80
Collection APIs	81
Underscore.js Collection Functions	82
Collection Events	83
Events	84
The Events API.....	84
listenTo and stopListening.....	85
View Events	85
Routing	86
Configuring Routes	87
Router APIs	88
Backbone History	89
Routing and Views.....	90
Backbone.js and RequireJS	93
Summary	98

■ Chapter 6: Creating a Single Page Application Step by Step	99
Choosing an IDE	101
Preparing the Solution Folders	101
Adding the JavaScript Libraries	103
Adding the HTML Page, Application Main Script, Styles, and Images.....	104
Crafting the Models	107
Agent Model Implementation	107
Task Model Implementation.....	108
Adding Collections	109
The Agent Collection.....	109
The Task Collection.....	109
Creating a Data Service and Generating Mock Data	110
Adding Front-End Routing	113
The Router	113
The Application Controller	114
Creating Views	116
The AgentTileView	116
The HomeView.....	118
The TaskView and EditModalView	119
The DetailsView	122
The CreateView.....	128
Summary	133
■ Part III: Building the Back End	135
■ Chapter 7: Creating a Back-End Service with ASP.NET Web API	137
Introduction to HTTP-Based Services	137
URIs	138
HTTP Verbs	139
Status Codes.....	141
Content Negotiation.....	142
RESTful Services	143

ASP.NET Web API Basics: Routing, API Controllers, and Actions	145
Creating an ASP.NET Web API Project	146
Routes and Controllers	151
Routing Requests to Controller Actions	153
Responses and Content Negotiation.....	157
Attribute-Based Routing.....	159
The HttpResponseMessage Class	162
Handling Exceptions.....	163
Handling Unhandled Exceptions	165
Validating the Input	166
Data Annotations	166
Validating Input with ASP.NET Web API	167
Validation Filters.....	168
Summary.....	169
■ Chapter 8: Implementing HTTP Concepts with ASP.NET Web API	171
Content Negotiation.....	171
Caching	176
Cache-Related Headers.....	177
Cache Headers and Content Negotiation.....	179
Cache Headers and Actions Methods	180
Resource Versioning with ETags	182
Uploads and Downloads.....	183
Downloading Files from a Service.....	184
Uploading a File to an Action Method	184
Summary.....	186
■ Chapter 9: Communication Between Front and Back End	187
Using Ajax to Communicate with the Server.....	187
The Promises Pattern	188
Backbone.js Ajax Support.....	189
Updating TheAgency.....	190

Working with Other Communication Options	196
Server-Sent Events (SSE)	196
WebSockets	197
Socket.IO	198
ASP.NET SignalR	199
Web Real-Time Communication (WebRTC)	202
Choosing Your Communication Option	203
Summary	204
■ Part IV: Advanced SPA Topics	205
■ Chapter 10: JavaScript Unit Testing	207
Unit Testing in the Front End	207
What Is TDD?	208
What Is BDD?	208
Jasmine Library	209
Setting the Environment	209
Creating Suites and Specs	211
Specs	211
Suites	211
Expectations and Matchers	211
Creating a Custom Matcher	212
Setup and Teardown	213
Nesting Suites	214
Using Spies	215
Testing Asynchronous Code	217
Writing Tests for TheAgency Project Using Jasmine	218
Creating and Setting Up the Environment	218
Models	221
Collections	223
Views	223
Summary	227

■ Chapter 11: SPA Performance Tuning.....	229
Optimizing the Front End.....	229
Browser Rendering Process	229
Profiling the Front End Using Chrome DevTools	230
Static Page Analysis	233
JavaScript Optimization Tips	235
Optimizing the Server (Back End and Network)	238
Tools of the Trade.....	239
Optimization Tips	254
Summary	265
■ Chapter 12: Search Engine Optimization for SPAs	267
The Search Engine Crawler	267
Using a White List: XML Sitemap	267
Using a Blacklist: robots.txt File	268
The Problem with SPAs and SEO.....	269
Options to Optimize Your SPA for SEO	270
Hijax.....	270
Escaped Fragments.....	274
Static Web Site/Application Representation	276
Summary	276
■ Chapter 13: SPA Deployment.....	277
Deployment Techniques	277
Xcopy.....	277
FTP.....	278
Version Control	278
Web Deploy Tool	279
Continuous Deployment	282

Deploying to the Cloud	283
Web Hosting and Platform as a Service	284
Static-File Hosting	284
Summary.....	287
Index.....	289

About the Authors

Gil Fink is a web development expert and Microsoft ASP.NET/IIS MVP. He works as a senior consultant and architect at SELA Group. He is currently consulting for various enterprises and companies, where he helps to develop web and RIA-based solutions. He conducts lectures and workshops for individuals and enterprises that want to specialize in infrastructure and web development. He is also a coauthor of several Microsoft Official Courses (MOCs) and training kits and the founder of Front-End.IL Meetup. You can read his publications on his blog at <http://blogs.microsoft.co.il/gilf>.

Ido Flatow is a senior architect and trainer at SELA Group, a Microsoft ASP.NET/IIS MVP, and an expert on Microsoft Azure and web technologies such as WCF, ASP.NET, and IIS. Ido is a coauthor of Microsoft Official Courses for WCF 4 (10263A) and Building Web Services for Windows Azure (20487B), coauthor of the book *Pro .NET Performance*, and the manager of the Israeli Web Developers User Group. You can follow Ido's work on his blog at <http://blogs.microsoft.co.il/blogs/idof> and on Twitter: @IdoFlatow.

About the Technical Reviewers



Konstantin Tarkus runs a small software consulting company, and over the last 10 years, he has helped multiple companies and tech startups to design their custom web and cloud applications on top of the Microsoft Web Platform. He is a strong proponent of the single page application (SPA) model and thinks that this book will be a valuable reference for web application developers interested in developing SPAs.

Fabio Claudio Ferracchiati is a senior consultant and a senior analyst/developer using Microsoft technologies. He works for Brain Force (www.brainforce.com) in its Italian branch (www.brainforce.it). He is a Microsoft Certified Solution Developer for .NET, a Microsoft Certified Application Developer for .NET, a Microsoft Certified Professional, and a prolific author and technical reviewer. Over the past ten years, he's written articles for Italian and international magazines and coauthored more than ten books on a variety of computer topics.

Acknowledgments

We would like to thank our reviewers, **Konstantin Tarkus** and **Fabio Claudio Ferracchiati**, who provided a lot of valuable feedback for each chapter in the book. They also helped to make the text very accurate, and we thank them for that.

Thanks to our colleagues and coworkers at SELA Group who took the time to review our early drafts and provide notes and corrections. Special thanks to Sasha Goldstein, SELA CTO, who was responsible for a lot of valuable feedback that improved the book's content.

Thanks to our family and friends, who waited patiently for us to finish writing and supported us in the process. Your support kept us going and helped us finish the book on time.

And last, we both wish to give our sincerest thanks to the people at Apress who made this book a reality. We would like to thank Gwenan Spearing for her guidance and help, Christine Ricketts for coordination and support, Christine Dahlin and Michael G. Laraque for copyediting, and Dhaneesh Kumar for helping the book get into production.

Introduction

One of the most important and exciting trends in web development in recent years is the move toward single page applications, or SPAs. Instead of clicking through hyperlinks and waiting for each page to load, the user loads a site once, and all the interactivity is handled fluidly by a rich JavaScript front end.

Most single page application development is done on the front end, as opposed to traditional web applications, which use server-side interactions and server-side render engines to do the same. This transition helps to create a more fluid user experience (UX) and makes web applications resemble desktop applications. Major web applications such as Gmail and Google Docs are implemented as SPAs, demonstrating that SPAs are the way to write your future web applications and not merely a transient trend.

As SPAs become the *de facto* standard of web application development, this book will be your one-stop shop for creating fluid, modern applications on the Web.

About This Book

This book will suit professional web developers familiar with HTML and JavaScript who wish to learn more. Readers will require experience with .NET and C# in order to follow along with Part III, which covers back-end development with ASP.NET Web API, but most of the content covers JavaScript.

What Do You Need to Know Before You Read This Book?

You need to have a good understanding of HTML and JavaScript, ideally from creating rich web apps. You need to understand the DOM API, know how events work, and have a solid grasp of the HTML elements and their DOM object counterparts. We will explain a lot about techniques to create object-oriented JavaScript and modular JavaScript, but we are not covering basic JavaScript.

How This Book Is Structured

This book is divided into four parts. Part I lays the groundwork for SPA development. You'll master some JavaScript techniques that will be useful later on and get to know the building blocks of a single page application, including modules, routing, and MV* frameworks.

In Part II, you'll build the client for your application. This is where the magic happens, as we take you through the process step by step. Backbone.js is the ideal library for demonstrating SPA development in practice, but in your future applications, you can apply the same principles with other frameworks.

Part III takes you through the process of building the server side of your application, using ASP.NET Web API, and hooking up the two parts of your application to create a working whole.

SPA development also comes with its own particular challenges, including tracking history, user interface performance, and how to handle search engine optimization. In Part IV, we guide you through some of these issues and advanced techniques and conclude by showing you how to deploy your application.

Part I: The Road to Single Page Application Development

Chapter 1: Introducing Single Page Applications (SPAs)

The book starts with an introduction to single page applications in which we explain the history and how we got to develop SPAs in the first place. In this chapter, you will learn how it all started and what the milestones were that led to contemporary web development.

Chapter 2: JavaScript for SPAs

This chapter explains in detail numerous critical JavaScript techniques that are essential for building SPAs. You will mainly learn how to create object oriented JavaScript and how to use it professionally. At the end of the chapter, we cover ECMAScript and explore future developments in ECMAScript 6.

Chapter 3: Modular JavaScript Development

Modular JavaScript is crucial for dividing your SPA into smaller parts that are maintainable and reusable. In this chapter, you will get to know a lot of patterns to create modular JavaScript. You will also learn about RequireJS library and how it can help you to create modular JavaScript.

Chapter 4: SPA Concepts and Architecture

In this chapter, you will learn about the building blocks of an SPA. You will be exposed to MV* and its influence on today's JavaScript writing. You will learn about routing, template engines, and crucial HTML5 JavaScript APIs that will help you to create your SPA. At the end of the chapter, we will also explain SPA architecture concepts.

Part II: Building the Front End

Chapter 5: Getting Started with Backbone.js

Backbone.js is the library that we use in this book. It is very small, and it is mostly used to impose structure in your code. In this chapter, you will get to know Backbone.js, and you will learn how to use its API to enforce structure and to create a more modular and reusable front end.

Chapter 6: Creating a Single Page Application Step by Step

Once you have the set of tools to create an SPA, you will create an SPA step by step. In this chapter, you will combine all the knowledge that you learned in the first five chapters and will build TheAgency, an SPA. You will create an SPA that works without a back end, and later in the book, you will see how to refactor it to work against a back end.

Part III: Building the Back End

Chapter 7: Creating a Back-End Service with ASP.NET Web API

Most SPAs are powered by a back-end service, which is usually developed as an HTTP-based service. In this chapter, you will learn how to create an SPA back-end service using ASP.NET Web API, a framework for developing HTTP-based services. This chapter will cover the basics of ASP.NET Web API and will take you through the steps of creating a set of services for the TheAgency SPA you created in Chapter 6.

Chapter 8: Implementing HTTP Concepts with ASP.NET Web API

When creating HTTP-based services, there are many features of HTTP that you can take advantage of, such as content negotiation and caching. In this chapter, you will learn some of the not-so-basic features of HTTP—content negotiation, caching, versioning, and streaming—and how to implement them with ASP.NET Web API. You will then use what you learned to extend the TheAgency SPA, to add more functionality to its back-end services.

Chapter 9: Communication Between Front and Back End

Once you've learned how to create a back-end service using ASP.NET Web API, you will learn how the front end can communicate with the back end. In this chapter, you will be shown both how to use Ajax and how to use it with the Backbone.js library. Later, you will refactor TheAgency to work with the back end you built in Chapters 7 and 8. At the end of the chapter, we discuss other communication options, such as Web Sockets, Server-Sent Events, and more.

Part IV: Advanced SPA Topics

Chapter 10: JavaScript Unit Testing

This chapter explains in details how to unit test JavaScript. We start by explaining what Behavior Driven Development (BDD) is. Later, you will get to know the Jasmine library, which is a BDD JavaScript library that will help you to create tests for your JavaScript code. At the end of the chapter, we create tests for TheAgency application.

Chapter 11: SPA Performance Tuning

Performance tuning is a crucial process when you create a web application. It helps you provide a better experience for your users, who won't need to wait for things to happen. In this chapter, you will have an opportunity to learn a lot of techniques to improve your performance, both in the front end and the back end.

Chapter 12: Search Engine Optimization for SPAs

SPAs are mostly created in the front end, using JavaScript. JavaScript itself is very difficult to optimize for search engines. In this chapter, you will learn what a search engine crawler is and how to use techniques to make an SPA optimal for search engines.

Chapter 13: SPA Deployment

In this, the last chapter of the book, you will learn the last step of SPA—deploying the application and its back-end service. You will also learn some of the familiar techniques used to deploy web applications and services to remote servers, whether they are hosted on-premises, in private, or in the public cloud.