

Table of Contents

About the Author	xiii
About the Technical Reviewer	xv
Introduction	xvii
Chapter 1: About MVC: Model, View, Controller	1
The History of MVC.....	3
MVC in Web Applications.....	4
MVC for Java	6
Finally, Java MVC (JSR-371).....	6
Why MVC	7
Where Is Hello World?	8
Exercises.....	17
Summary.....	18
Chapter 2: Prerequisite: Jakarta EE/Java EE	19
The Nature of Java for Enterprise Applications.....	19
GlassFish, a Free Java Server	25
Getting GlassFish.....	26
GlassFish Shell Administration	28
GlassFish GUI Administration.....	32
GlassFish REST Interface Administration.....	33
Using a Preinstalled Java Server	36
Learning Java for Enterprise Applications	36
RESTful Services.....	37
Exercises.....	41
Summary.....	41

TABLE OF CONTENTS

- Chapter 3: Development Workflow 45**
- Using Gradle as a Build Framework..... 45
- Using Eclipse as an IDE..... 46
 - Installing Eclipse 46
 - Configuring Eclipse..... 48
 - Adding Java Runtimes 49
 - Adding Plugins..... 49
 - Eclipse Everyday Usage..... 50
- More About Gradle 51
 - A Basic Gradle Project 51
 - Gradle Main Concepts 53
 - Standard Gradle Project Layout..... 54
 - The Central Gradle Build File 55
 - Running Gradle Tasks 57
 - Gradle Tasks Explained..... 61
 - Gradle Plugins 63
 - More About Repositories 64
 - More About Dependencies..... 66
 - Changing the Project Structure 69
 - The Gradle Build File Is a Groovy Script 70
 - Script Variables..... 72
 - Custom Tasks 73
 - The Gradle Wrapper 74
 - Multi-Project Builds 75
 - Adding a Deploy Task 79
- Developing Using the Console..... 81
- Installing MVC 84
- Exercises..... 85
- Summary..... 86

Chapter 4: Hello World for Java MVC	95
Starting the Hello World Project.....	95
The Hello World Model	102
The Hello World View	104
The Hello World Controller	107
Using Gradle to Build Hello World	108
Starting a Jakarta EE Server.....	111
Deploying and Testing Hello World.....	111
Exercises.....	114
Summary.....	115
Chapter 5: Start Working with Java MVC	117
Handling User Input from Forms.....	117
Exception Handling in Java MVC.....	120
Non-String POST Parameters.....	124
Handling Query Parameters.....	126
Exercises.....	130
Summary.....	130
Chapter 6: In-Depth Java MVC	133
The Model	133
CDI in Java MVC.....	134
Model Object Scopes.....	137
The Simplified Model Data Container	139
The View: JSPs.....	140
JSP Basics.....	141
Directives.....	141
Static Content.....	143
Java Scriptlets and Java Expressions	144
Implicit Objects.....	145
JavaBeans Components	145
Expression Languages.....	146

TABLE OF CONTENTS

Output.....	148
Variables.....	149
Loops.....	150
Conditional Branching.....	152
Cookies.....	153
The View: Facelets.....	154
Facelets Files.....	155
Facelets Configuration.....	155
Templating via Facelets.....	157
The <ui:decorate> Tag.....	161
An Example Facelets Project.....	164
Mixing Facelets and JSTL.....	177
Unified Expressions.....	178
The Controller.....	179
Controller Basics.....	179
Getting Pages.....	180
Preparing the Model.....	182
Posting Data into Controllers.....	183
Exercises.....	187
Summary.....	187
Chapter 7: In-Depth Java MVC: Part II.....	193
Adding Bean Validation.....	193
Injectable Context.....	203
Persisting State.....	205
Dealing with Page Fragments.....	207
Observers.....	212
Configuration.....	215
Exercises.....	218
Summary.....	218

Chapter 8: Internationalization	223
Language Resources.....	223
Adding Localized Messages to the Session.....	225
Formatting Data in the View.....	229
Using JSF for Formatting.....	234
Localized Data Conversion.....	236
Exercises.....	238
Summary.....	238
Chapter 9: Java MVC and EJBs	241
About Session EJBs.....	241
Defining EJBs.....	242
Accessing EJBs.....	246
EJB Projects.....	248
EJBs with Dependencies.....	250
Asynchronous EJB Invocation.....	252
Timer EJBs.....	253
Exercises.....	256
Summary.....	257
Chapter 10: Connecting Java MVC to a Database	261
Abstracting Away Database Access with JPA.....	261
Setting Up a SQL Database.....	262
Creating a Datasource.....	264
Preparing the Member Registration Application.....	266
Adding EclipseLink as ORM.....	274
Controllers.....	275
Adding Data Access Objects.....	279
Updating the View.....	281
Adding Entities.....	283

TABLE OF CONTENTS

Adding Relations	285
Exercises.....	289
Summary.....	291
Chapter 11: Logging Java MVC Applications	295
System Streams.....	295
JDK Logging in GlassFish.....	296
GlassFish Log Files	297
Adding Logging Output to the Console	297
Using the Standard Logging API for Your Own Projects.....	298
Logging Levels.....	299
The Logger Hierarchy and Thresholds	299
The Logging Configuration	301
The Logging Format	303
Using JDK Standard Logging for Other Servers	303
Adding Log4j Logging to Your Application.....	304
Adding Log4j Server-Wide	305
Changing the Logging Format	308
Adding Log4j to Jakarta EE Web Applications	310
Using Log4j in the Coding.....	312
Exercises.....	313
Summary.....	314
Chapter 12: A Java MVC Example Application.....	321
The BookLubb Database	321
The BookLubb Eclipse Project	323
The BookLubb Infrastructure Classes	326
Configuring BookLubb Database Access	328
The BookLubb Internationalization	328
The BookLubb Entity Classes.....	333
BookLubb Database Access via DAOs	340
The BookLubb Model	347

TABLE OF CONTENTS

The BookLubb Controller	354
The BookLubb View	364
Fragment Files.....	365
Landing Page.....	367
Member-Related View Files.....	368
Book-Related View Files.....	381
Deploying and Testing BookLubb.....	390
Summary.....	391
Appendix:	393
Solutions to the Exercises.....	393
Chapter 1 Exercises.....	393
Chapter 2 Exercises.....	394
Chapter 3 Exercises.....	394
Chapter 4 Exercises.....	396
Chapter 5 Exercises.....	399
Chapter 6 Exercises.....	403
Chapter 7 Exercises.....	405
Chapter 8 Exercises.....	413
Chapter 9 Exercises.....	423
Chapter 10 Exercises.....	428
Chapter 11 Exercises.....	433
Index	437

Introduction

Starting at the very infancy of software creation, developers tried to modularize their applications in order to streamline their projects and increase the maintainability of the software they created. Soon, a very basic segregation scheme was identified: One part of the software must deal with data and persistence, another part must deal with presenting the data to the user, and one last part must handle data input and frontend view propagation.

This segregation scheme showed up in so many projects that it was promoted to a common software design pattern, called Model-View-Controller, or MVC for short. Its power also manifested in its versatility, even with big paradigm changes, like the onset of the Internet age. With database products for the model layer, browsers for the view layer, and some kind of user input processing for the controller layer, the pattern's accuracy and applicability to the majority of software projects became even more apparent with web applications.

Interestingly, even though most web application frameworks under the hood apply some kind of MVC layer demarcation, Java Server products up to JEE 7 did not include a dedicated MVC framework. With JSR-371 (Java Specification Request number 371) only recently and starting with JEE 8/Jakarta EE 8, an MVC specification entered the Java Enterprise application realm, which is one of the reasons this book was born. It does not describe all MVC Frameworks that you can add to Java EE/Jakarta EE as an external library. There are just too many of them and you can learn about them by looking at each library's documentation. Instead, we talk about the genuine Java MVC library as described by JSR-371.

The target version of Java MVC is 1.0, and we use a Jakarta EE version 8.0 compliant server to run Java MVC on it.

The Book's Targeted Audience

The book is for beginning or advanced enterprise software developers with knowledge of Java Standard Edition version 8 or later and some experience in Jakarta EE (or JEE) development. It is also assumed that the reader is able to use the online API references,

INTRODUCTION

as this book is not a reference in the sense that all API classes and methods are listed. Instead, it presents techniques and technologies that help professional Java Enterprise level developers leverage web application programming by including Java MVC in their software.

The book uses the Linux operating system as the development platform, although the code can be run on other platforms (Windows and macOS) without complex adaptations. This book also does not talk about hardware issues (in case you don't use a laptop, a PC, or a server).

The readers will in the end be able to develop and run Java MVC programs of mid- to high-level complexity.

Sources

All sources shown or referred to in this book can be accessed via the Download Source Code button located at www.apress.com/9781484262795.

How to Read This Book

You can read this book sequentially from the beginning to the end, or you can read chapters on an ad hoc basis if your work demands special attention on a certain topic.