

Spring MVC Tutorial – Contact List Application

Overview



SOFTWARE-GUILD

Copyright © 2016 The Learning House, Inc.

All rights reserved. No part of these materials may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of The Learning House. For permission requests, write to The Learning House, addressed "Attention: Permissions Coordinator," at the address below.

The Learning House

427 S 4th Street #300

Louisville KY 40202

Spring MVC Tutorial Overview

This tutorial is an opinionated approach to building database-driven Spring MVC web applications. The application presented here is a simple contact list manager that will allow the user to create, read, update, and delete contacts from the system. The application will be built with the following technologies:

1. Spring MVC
2. REST Web Services
3. Spring Core (for Dependency Injection)
4. Maven
5. NetBeans
6. Tomcat
7. MySQL
8. Twitter Bootstrap (for CSS and styling)
9. jQuery

This will be a hybrid application consisting of both traditional Spring MVC JSP endpoints and REST endpoints accessed via Ajax. The tutorial is intended as a template for general Spring MVC applications, and all concepts presented in this tutorial can be extended to more complicated applications.

We will build the application in these stages, following Agile principles so that we have working, demonstrable software at each step:

1. A simple Hello, World! MVC application built from the SWC Guild Spring MVC Maven archetype
2. A mocked-up version of the application that demonstrates the user interface using canned client-side data and minimal server interaction
3. A fully-functional version of the application that uses an in-memory datastore for contact information (in this version, all contact information is lost when Tomcat is restarted); we will use interfaces and Spring dependency injection (DI) to loosely couple our application to this datastore. In this version, all data can be stored on and retrieved from the server.
4. A fully-functional version of the application that uses a MySQL database for contact information; we will take advantage of the interface-based/DI design of the previous step to easily swap the in-memory datastore for the MySQL datastore