

Spring Framework for Testing

Darshak Mota ¹, Neel Zadafiya ², and Jinan Fiaidhi ²

¹Lakehead University

²Affiliation not available

October 30, 2023

Abstract

Java Spring is an application development framework for enterprise Java. It is an open source platform which is used to develop robust Java application easily. Spring can also be performed using MVC structure. The MVC architecture is based on Model View and Controller techniques, where the project structure or code is divided into three parts or sections which helps to categorize the code files and other files in an organized form. Model, View and Controller code are interrelated and often passes and fetches information from each other without having to put all code in a single file which can make testing the program easy. Testing the application while and after development is an integral part of the Software Development Life Cycle (SDLC). Different techniques have been used to test the web application which is developed using Java Spring MVC architecture. And compares the results among all the three different techniques used to test the web application.

Spring Framework for Testing

Darshak Mota
Computer Science
Lakehead University (1101352)
Thunder Bay, Canada
dmota@lakeheadu.ca

Neel Zadafiya
Computer Science
Lakehead University (1115533)
Thunder Bay, Canada
nzadafiy@lakeheadu.ca

Dr. Jinan Fiaidhi
Computer Science
Lakehead University (Guide)
Thunder Bay, Canada
jinan.fiaidhi@lakeheadu.ca

Abstract—Java Spring is an application development framework for enterprise Java. It is an open source platform which is used to develop robust Java application easily. Spring can also be performed using MVC structure. The MVC architecture is based on Model View and Controller techniques, where the project structure or code is divided into three parts or sections which helps to categorize the code files and other files in an organized form. Model, View and Controller code are interrelated and often passes and fetches information from each other without having to put all code in a single file which can make testing the program easy. Testing the application while and after development is an integral part of the Software Development Life Cycle (SDLC). Different techniques have been used to test the web application which is developed using Java Spring MVC architecture. And compares the results among all the three different techniques used to test the web application.

Index Terms—Java spring, Spring Framework, Software Testing, Software development, Apache MAVEN

I. INTRODUCTION

Java Spring is the most popular application development framework for enterprise Java. It is an open source platform which is used to develop robust and efficient Java applications easily. The Java Spring framework is called as Framework of Frameworks as it is a collection of sub-frameworks [1]. It supports frameworks like Hibernate, Struts, EJB, JSF, etc. [2]. The Java Spring framework contains many different predefined codes (templates) which can be used to perform various functionality in an application. It is considered as a light-weight framework because the business logic code is very short compared to the code written from scratch. Java Spring framework is well known for developing loosely coupled applications and the code can be arranged in modular format which gives security and proper formatting for classes and packages. For testing, it provides some automated test

classes and due to loose-coupling it is easier to test the applications.

- Data Access / Integration layer: These are sub layers of data access layers that also contains data access objects.
- JDBC: Spring provides us with a built-in template for implementing JDBC, hence no need to go through the tedious steps for connecting our application to the database. It removes the need to do JDBC coding.
- ORM: ORM stands for Object Relational Mapping. It is a tool that lets you query and manipulate data from a database using an object paradigm. Here, you don't use

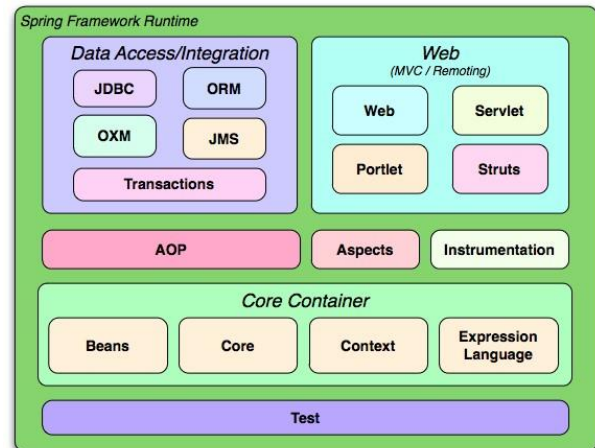


Fig. 1. Architecture of Java Spring Runtime [2]

the SQL anymore, instead directly object of the language i.e. Java in our case.

- OXM: OXM stands for Object XML Mapping. It is the component responsible for linking the object with the XML i.e. Bean class.
- JMS: JMS is Java Messaging Service which is used to create, send and receive messages from one client to another.
- Web layer: These layers contains all the objects that participates in construction of HTTP response output.

- Web: It provides functions such as downloading files, creating web applications, rest web applications or services, etc.
- Servlet: Also referred as Web-MVC, it contains Spring MVC implementation for web application.
- Struts: If you want to use the Struts framework in your Spring web application, then this functionality of Struts MVC can be added using Struts framework in our Spring framework.
- Portlet: It provides the MVC implementation used in Portlet environment. Portlet is like Servlet where both are managed by their own containers i.e. Servlet container for Servlet and Portlet container for Portlet. The main difference between both is that Servlet renders complete web pages, whereas Portlet renders fragments of HTML i.e. it renders a part/component of the whole web page rather displaying the whole web page.

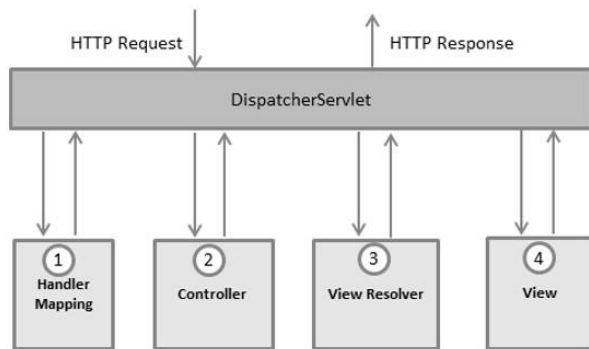


Fig. 2. Architecture of Java Spring MVC [11]

Once the user requests for a web page, an HTTP Request is raised, DispatcherServlet accepts this request and consults HandlerMapping to call the appropriate Controller. Now this Controller takes the request and calls the appropriate service methods based on the used GET or POST method [3]. The service method sets the model and returns view name to the DispatcherServlet. With the help of View Resolver, DispatcherServlet will pickup the defined view for the request. Once the view for the request is finalized, the DispatcherServlet passes the model data to the view which is then sent to the user as a response of HTTP Response. This response is the requested web page.

Then we get started with how to install and use spring. First, we need JDK and an IDE to run the code, so we use JDK8 and Eclipse IDE to write The Spring code. We also need to install and load all the spring libraries which are jar files which consists all the functionality of spring. Then we show how to install Eclipse and load The

Spring libraries and get set to start programming in spring. Creation of required java and xml files are also explained. To explain a simple spring application, we create a basic “hello world” application. There are 5 steps to run a spring application. First, we need a Bean Class, second an XML file, third the main class file (Controller), forth loading all the jar files and fifth executing the code. A Bean in java is an object. This bean object is controlled by the IOC (Inversion of Control) container (spring container). A bean lifecycle consists of instantiating of object, populate the bean properties, then bean is ready, then shutting down the spring container followed by the destruction of the bean. Now all these processes are controlled by the spring container i.e. IOC container (Inversion of Control). IOC says that the objects should not be created by you, objects shall be configured in an XML file. This IOC container is responsible to construct the object by parsing the XML file [4]. Hence, we need to mention the data which the attributes will have in a separate XML file. This XML file leads to IOC. IOC is responsible for creation of objects and managing its lifecycle. There are two types of IOC containers. One if using Bean Factory and second is the

Application Context. The Bean Factory constructs the object only when it is requested, whereas the Application Context constructs the object even when you don’t request for it.

Now let’s talk about the Dependency Injection. Dependency Injection is used to reduce the coupling. Basically, a code is called highly coupled when a class’s object creation and execution is dependent on another class. Spring says that its applications must be loosely coupled which helps to test the application easily. The same XML file is used to achieve this Dependency Injection. There are two types of Dependency Injection: 1. By Constructor 2. By Setter Method. Then a simple Employee and Address example is shown using constructor injection and setter injection.

II. SOFTWARE TESTING

The core topic of the research is JUnit. The project started with introduction to software testing. Software testing is the phase in software development life cycle which is in between development and deployment of the system [5]. In other words, software testing is the process to test performance and correctness of software.

There are two major type of software testing which are functional testing and non-functional testing.

Functional testing involves unit testing, integration testing, system testing, beta testing and other testing which concerns with functionality of software [6]. Non-functional testing involves stress testing, load testing, compatibility and other testing that concerns with deployment of software. The focus of the research was on unit testing. Unit testing involves testing of single individual unit only. It can be done during the development of software. Unit testing can be done manually or using automated test tools. Some of the tools are selenium, cucumber and JUnit. When it is hundreds of methods to test in single unit and hundreds of units to test in the system, it is useful to use automated test tools like JUnit.

JUnit is open source library that provides automated test methods to test single unit of code. The purpose of unit testing is faster development of software with reduced cost and JUnit server better in this area. It also can be run automatically after each pass of the development. The core classes provided by JUnit are Assert, TestCase, TestResult and TestSuite [7]. Assert class provides assert methods which takes result generated by method to be tested and expected output. It compares them both and generates output for test result. TestCase contains test method inside it which runs assert method. Several test cases can be combined into test suit. The final output of test cases can be generated by TestResult.

The question was how to decide order of methods to be tested. The answer was it follows same execution order in code. The one you want to execute first, put that first in test logic. The small example of JUnit was tested later. The example tests two simple methods written main class. The last example was to use JUnit with spring framework [8][9]. All the programs in example were explained step by step using screenshots. JUnit with spring framework can be used in same manner as it is being used with normal program. The only difference is that supplication context of class needs to be created using spring in the test method. After the last example, project was declared finished.

III. METHODOLOGY

Unit Testing is the type of testing which is done during development phase of Software Development Life Cycle (SDLC). It's developer's responsibility to perform Unit Testing, hence Unit Testing is done by the developer itself and not any third-party testing organization or end users. Unit Testing means testing each unit in your code i.e. every single function or textbox needs to be tested

to check if a component works appropriately or not. For example, you can perform Unit Testing on a Login page to check if the username and password credential given by the user matches in the database or in Register Page to check if the entries made by user are successfully inserted in the database table or not. We have used three types of testing frameworks to test our web application namely JUnit, TestNG, and Spring's Test Framework. JUnit is a testing framework which is used for Unit Testing only. Spring is known as Framework or Frameworks because it can incorporate with many different frameworks, hence you can use JUnit and TestNG Frameworks for testing your spring base web applications. TestNG is another framework which is inspired from JUnit test framework but with additional functionalities. NG stands for Next Generation, TestNG provides an automatic testing environment. We have testing our web application using three different frameworks and compared them.

The core topic of the research is development and deployment of spring MVC web application using maven. The project started with introduction to environment setup for maven. The tools and libraries are used to setup an environment includes IntelliJ IDEA Ultimate 2019, Tomcat 8, JDK 8, Apache Maven 3, Spring 5 and MySQL 8. Additionally, MySQL Workbench is also used to view data in the database. The source code of project contains java code and SQL code as well.

IntelliJ IDEA Ultimate 2019, Tomcat 8, JDK 8 and MySQL 8 needs to be downloaded and configured manually. But Spring and Maven will be automatically installed by IntelliJ Idea IDE. Links are provided in the help document to download and configure other components. First step to deploy the existing maven project is to import it using IDE as maven project. It can be installed using Ctrl + E in IDE and select Install. It generates WAR file for project. WAR file is like JAR file, but it is a bundle of classes and web pages. The newly generated WAR file needs to be present at WEB-APP directory of tomcat server and accessed using localhost [10]. But before going to the home page it is necessary to execute SQL queries given in project in order to create a database.

The project is divided into multiple categories of files which includes model, view, controllers, services, DAO, beans, deployment descriptor and test files [11]. The project starts with views which are JSP files and the home page is 'home.jsp'. The other two JSP files are login and register which has associated controllers. These controllers invoke when these JSP files are open

in browser. The controllers convert their http response into model class and sends a bean of that object. In this project, the model classes are login, user and product. These are POJO classes and can be used in throughout the project. User also must insert dependencies in POM.XML and specify welcome pages in web.xml. user-beans.xml contains bean path for JDBC connector and JDBC driver. DAO classes are data access objects which provides methods to access database in the project. All these files together create maven structure and can be deployed into WAR file.

When war file is executed in tomcat server, the first page to see is 'home.jsp'. It has two links for other pages which are login and registration. Successful login or registration leads to product page in which list of available products is presented in the table. The further development is stopped here because this is all the necessary development to test the test cases. Four different type of test cases are tested using three different methods. The test cases include database inset, select, delete and select operations. Three different ways to test those cases are JUnit integrated with spring, JUnit and TestNG. The performance of those test cases is given below.

IV. RESULTS

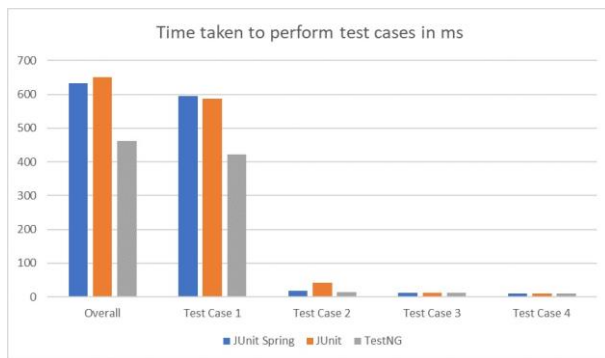


Fig. 3. Result of Test Time

The overall time taken to perform test by TestNG is better than all the other test frameworks. But in the case 4, time taken by TestNG is 1ms high then the other frameworks. So, it is better to analyze the need of testing in the project and select the testing framework according to that.

V. CONCLUSION

Among all the three testing i.e. JUnit, TestNG and Spring Test, for our project TestNG tests all the test cases faster than any other framework. But this won't be same for every web application because it depends on how much data an application has. If a greater number of data is to be fetched from database, then using JUnit framework will test all the test cases faster than TestNG. On the other hand, TestNG performs well when you must update the database i.e. insert, delete or update any values. Overall, for our project, unit testing using TestNG framework provided the best and quick results.

REFERENCES

- [1] Ansari, Amir. (2019). Analysis and Performance Issue of Java and Its Framework and Impacts on Web Application. 10.13140/RG.2.2.11281.48489.
- [2] Jovanovic, Zeljko & Jagodic, Dijana & Vujicic, Dejan & Randic, Sinisa. (2017). Java Spring Boot Rest WEB Service Integration with Java Artificial Intelligence Weka Framework.
- [3] Venkadesh, M. & Sekar, A. (2015). Study on implementing web services using JAVA spring with eclipse and apache Tomcat. 10. 9773-9787.
- [4] Le Borgne, Alexandre & Delahaye, David & Huchard, Marianne & Urtado, Christelle & Vauttier, Sylvain. (2018). Recovering Three-Level Architectures from the Code of Open-Source Java Spring Projects (S). 199-235. 10.18293/SEKE2018-140.
- [5] Gulati, Shekhar & Sharma, Rahul. (2017). Java Unit Testing with JUnit 5: Test Driven Development with JUnit 5. 10.1007/978-1-4842-3015-2.
- [6] Albrecht, M. (2003). Testing Java interfaces with JUnit. Doctor Dobbs Journal. 28. 24-28.
- [7] Fowler, Martin. (1999). A UML testing framework. Software Development. 7. 41-46.
- [8] Jain, Manish & Gopalani, Dinesh. (2017). Automated Java Testing: JUnit versus AspectJ. 10.5281/zenodo.1314478.
- [9] Hunt, Andy & Thomas, Dave. (2003). Pragmatic Unit Testing in Java with JUnit.
- [10] Shivaprasad, Suma & Prasad, Nanditha. (2013). Unit Testing Concurrent Java Programs. International Journal of Computer Applications. 67. 4146. 10.5120/11435-6798.
- [11] Beust, Cedric & Suleiman, Hani. (2020). Next generation java™ testing: testing and advanced concepts.