

REST API Implementation using Spring MVC and Hibernate Framework

¹Apoorva, ²Dr. G S Sharvani

¹Student, ²Associate Professor

^{1,2}Department of Computer Science Engineering,

^{1,2}R.V. college of Engineering, Bengaluru, India

Abstract : The application logic is concealed in the client's User Interface or in the server's database in a two-tier architecture. It's simple to construct and maintain, but it's also slower and less secure. So, everyone is moving to three tier architecture in which application logic is encapsulated in Business layer. The client communicates with Business layer through REST API calls. Various frameworks like Spring and various implementations like Jersey, RESTEasy, Restlet, Apache CFX which uses Java API for RESTful Web Services (JAX-RS) specifications have been used for implementing RESTful web services. In recent past, Java developers are intensively using Spring Model-View-Controller (MVC) to implement RESTful web services. Spring framework is a well-known Web development technique that is built on the J2EE platform architecture. It is built on the MVC design pattern and it decouples the layers by separating the system into the controlling part, business logic part, and view one. The Hibernate framework provides a way to package a lightweight object for JDBC and makes business development easier. In this paper, we discuss about implementing RESTful web service using Spring MVC framework which is most used Java EE framework and Hibernate which is best Java Object – Relational Mapping (ORM) tool.

IndexTerms - REST, API, Spring, Hibernate, HTTP, Web service

I. INTRODUCTION

The application logic is concealed in the client's User Interface or in the server's database in a two-tier architecture. The user interface is generally found on a user's desktop, while database management functions are typically found on a server. It's simple to construct and maintain, but it's also slower and less secure because the client is allowed to communicate directly with the database. It has tight coupling. The performance of application reduces when number of users increases. The application logic is located in the middle-tier in three tier architecture. The data and the user interface are segregated from the business logic or core functionality. The system which incorporate three-tier architecture are more scalable, resilient, and adaptable than single-tier systems. Client Layer, Business Layer, and Data Layer are the three tiers of a three-tier architecture. These solutions are quicker and more secure since the client will not be able to interface directly with the database. The client communicates with the other tiers through application program interface (API) calls.

API is a messenger which accepts requests, informs the system what the user wants to accomplish, and then gives the result to the user. REST is software architectural style or guidelines to implement web services which are used to create interactive applications. In this paper, we are discussing about implementation of RESTful web services in Java. The most famous framework, Spring MVC is used. Spring is most popular Java framework for Java projects. The powerful features of Spring are loose coupling and dependency injection. RESTful web service development is supported by Spring. Spring is lightweight implementation of Java framework. The framework also offers robust security features that you can simply call as functions that have already been created. Processes like authentication, verification, and validation become much easier to incorporate into any project as a result of this. We will explore the annotation style of writing web services in this paper. Spring provides a number of REST annotations, which can be thought of as pre-written functions.

Spring can easily be integrated with other frameworks. We employ Hibernate, a popular Java Object-Relational Mapping tool, in this study. It gives you a way to convert a relational database into an object-oriented domain model. Hibernate's major capability is the mapping from Java classes to database tables, as well as the mapping of Java datatypes to SQL data types. JPA annotation is used by Hibernate to bind Java classes to database tables.

Section I contains Introduction of REST API implementation using Spring MVC and Hibernate framework, Section II contain the related works, technologies used, Section III provides explanation of the methodology, Section IV picturizes the results and discussion of implementation of RESTful web services, Section V concludes the research with future work.

II. RELATED WORK

In [1], Hongjun Li describes the method of using RESTful Web service frameworks in Java to construct RESTful Web services and describes numerous Restful web service frameworks, including Axis2, Restlet, Grails, RestEasy, sqlRest, and Struts 2. The limitation of this paper was it did not compare between the frameworks. The authors of [2] by Urjita Thakar, Amit Tiwari, and Sudarshan Varma demonstrate how to consume SOAP based as well as RESTful services in composition while maintaining the advantage of RESTful services' lightweight nature. The work related to selecting appropriate component services, detecting faulty component services, and replacing them was not discussed in this study. In their study [3,] K Munonye and P Martinek compared and assessed the efficiency of REST API implementations using the Microsoft.Net framework vs the Java SDK. The paper's limitations were that the same workstation served as both the server and the client, that only two metrics were examined in this study: response time and code complexity, and that the dataset was small. The authors of [4] showed how to construct a system with high data interchange efficiency, simple interfaces, simplicity of modification and growth, and strong cross-terminal support based on JSON data format and RESTful web services. The limits described in this study are to improve the system's performance and enhance the efficiency by utilizing some other new technologies. The authors of [5] HaiTao Wang and BaoXian Jia, explain how framework integration can improve system development efficiency and reduce coding workload. The paper's weakness was that it did not explain how to improve system performance, user access speed, or system framework security. In [6], Yuxiang Hou proposes the development of the framework for Spring MVC, MyBatis, and Spring in the creation of Web applications and it simplifies the development process and workload of the system to improve the system's expansion and convenience of deployment. Zhang Zhenyou, Gu Wei, and Cao Zhi demonstrate how Hibernate technology makes the system dynamically linked to multiple heterogeneous database systems in their paper [7]. The limitation of this paper is there was no information of query optimization. In [8], Qinglin Wu, Yanzhong Hu, and Yan Wang described how to build, test, and optimise data persistence layer methods to increase data access efficiency and Web application quality.

III. METHODOLOGY

3.1 Software used

3.1.1. Java 11

Java programming language is used for RESTful web services implementation. After Java 8, Java 11 is the first long-term support (LTS) release.

3.1.2. Eclipse IDE

Eclipse is an integrated development environment for computer programming (IDE). It comes with a pre-configured workspace and a plug-in framework for further customization.

3.1.3. Spring MVC

The Spring Web MVC framework gives MVC architecture with preconfigured components that may be used to develop flexible and loosely coupled web applications. This pattern divides an application's multiple components that is input logic, business logic, and user interface logic while keeping relationship which is not compact, between them.

- In most situations, the Model will be built up of POJO and will hold the application data.
- The View displays the model data and, in most situations, outputs the HTML formatted data that the client's browser understands.
- The Controller is in responsibility of constructing an appropriate model by receiving user requests, which is subsequently provided to the view to be shown.

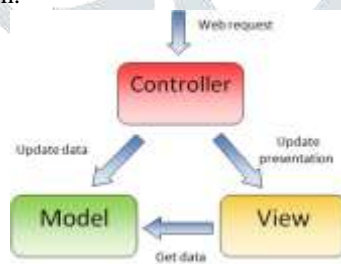


Fig. 1 Spring MVC architecture

3.1.4. Hibernate

Hibernate is a framework for Java that makes it easier to construct database connected Java applications. It's an lightweight Object Relational Mapping (ORM) tool which is open source. Hibernate follows the Java Persistence API (JPA) criteria for data persistence.

3.1.5. MySQL

MySQL is a Relational Database Management System (RDBMS) that is available as an open source project. REST APIs developed fetch and save the data into MySQL database.

3.1.6. JBoss

JBoss is a server that runs Java applications. The JBoss application server from Red Hat is a platform which is open source for producing applications in Java and a variety of other software applications. You can create and deploy Java services that can be scaled to match your company's needs.

3.1.7. JUnit

JUnit is a Java programming language unit testing framework. JUnit is part of the xUnit family of unit testing frameworks, which began with SUnit and has played an essential role in the development of test-driven development. At compilation time, JUnit is linked as a JAR.

3.1.8. Apache JMeter

Apache JMeter is a load testing tool that may be used to analyze and measure the performance of a wide range of services, with an emphasis on web applications.

3.2 Implementation

REST is the acronym for REpresentational State Transfer. REST is a architecture based on web services that employs the HTTP protocol. Web Services developed using the REST Architecture are known as RESTful Web Services. RESTful web services are often used to construct APIs for web-based applications because they are lightweight, scalable, and maintainable. In the REST, everything is a resource. The resource can be image, file, video, etc. URIs are used to identify each resource. Every REST API has a HTTP method, URI, request object and response object.

There is an endpoint for every RESTful webservice. Each endpoint is a location where APIs may have access to the resources, they require to do their tasks. To represent a resource, REST employs multiple representations such as XML, text or JSON. The most often used is JSON.

When we send a request for a resource by specifying the HTTP method, URI and request body, we get the response in the form of JSON along with HTTP status codes.

In a REST-based design, the four HTTP methods listed below are often utilized.

- GET method is used to access a resource in a read-only mode.
- POST method, by which new resources are created.
- DELETE method, by which already existing resources are deleted.
- PUT method is used to update a resource which is already existing.

For coding, we use Java programming language. First, we need to install all maven dependencies for spring and hibernate. We have 4 layers for REST API implementation. The four layers are Controller layer, Data service layer, Business Object layer and Data Access Object layer. Each layer has an interface and implementation file. Different Java classes have been written for request and response object. As hibernate is used for mapping Java classes to database tables, POJO class is written. POJO is normal Java class which has attributes same as the columns in the table of database. Entity of POJO will be represented as columns of a table in database. All of these mappings will be done by Hibernate utilizing annotations. JPA annotations have been used in POJO class to specify that it is an Hibernate entity. Annotations used are:

@Entity: This annotation specifies that this particular class is hibernate entity which will be mapped to a table in database.

@TableName: This specifies the name of the table which hibernate should map.

@Id: This specifies that that this particular attribute is the table's primary which hibernate is mapping to.

@Column: This specifies the name of column that attribute of Java class is mapping to.

There are few more JPA annotation for like @OneToOne, @ManyToOne for specifying foreign key constraints.

The sequence diagram for GET API is given below:

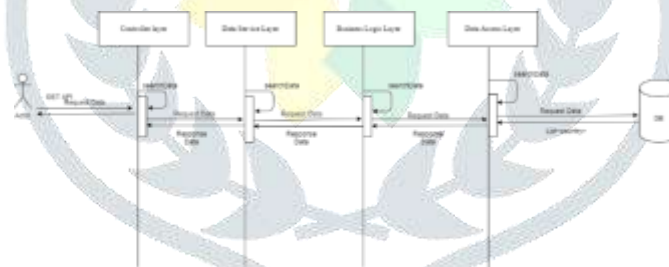


Fig. 2 Sequence diagram of API

In Controller layer, we have specified the type of Webservice by using annotation such as @GetMapping, @PostMapping, @PutMapping and @DeleteMapping. We have also specified the end point of the web service. This layer takes the input as request object and send the request to next layer i.e., Service layer. This layer displays the response data obtained from service layer.

Service layer acts as an interface between Controller layer and Business logic layer. It separates the business logic from the client to guarantee the security of information and encapsulates business logic. In this layer, we use annotations such as @Transactional and @Secured to make sure that only authenticated users can access this web service.

In business logic layer, validation check of the parameters if request object is done. Here we have written logic to set the result data list to our response object which is in the form of JSON. Exceptional handling and error messages to shown when request is not successful is written here.

In data access layer, we create an object of SessionFactory, which is a factory class through which we get sessions and perform database operations. We have used getCurrentSession() method to get the Session object. Then we write hibernate query to execute in database. Hibernate converts the query into SQL query and JDBC connection is made to database and we get the results from database.

After the implementation is done, we deploy our services in Jboss server and we hit the URI of the webservice through Postman and check the response object. Then Junit test cases have been written as part of Unit testing.

IV. RESULTS

The response from Postman is shown here:

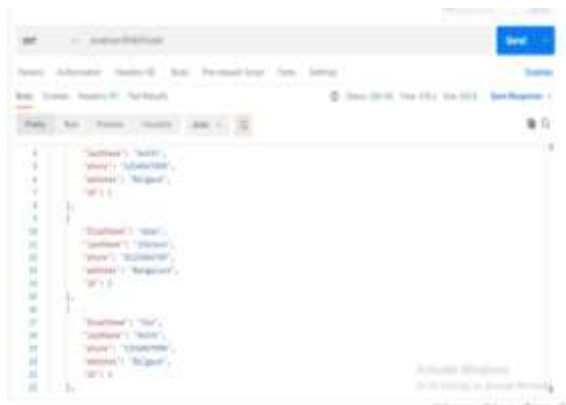


Fig. 3 Results from Postman

EclEmma tool is used to check the code coverage under our test cases. The results of code coverage is shown below:

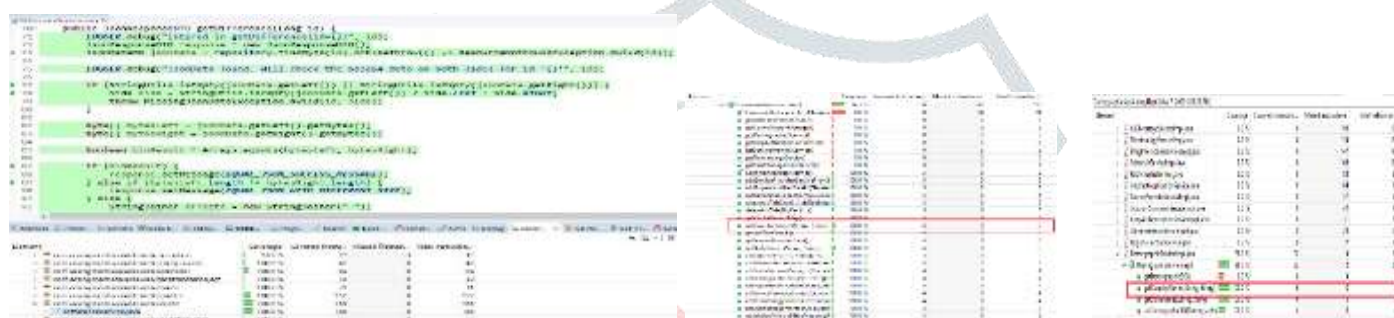


Fig. 4 Code coverage results

Apache JMeter is used for load testing of our web services. We have checked the response time for 10 users. The result is shown below:



Fig. 5 Load testing results

V. CONCLUSION

The paper presents a way to implement Restful web services in Java using Spring and Hibernate framework and it provides a multilayer implementation using four layers. The code coverage is more than 90% for all the methods. The response time for all the web services is less than 200 milli seconds. This method of implementation can be used for enterprise applications development. The actual operating of the system is very steady and trustworthy. This has been demonstrated to be an excellent lightweight J2EE application solution.

REFERENCES

- [1] Hongjun Li, "RESTful web service frameworks in Java," Proc. IEEE Symp. Computational Intelligence in Scheduling (SCIS 07), IEEE Press, Dec. 2007, pp. 57-64, doi:10.1109/SCIS.2007.357670.
- [2] Urjita Thakar, Amit Tiwari, Sudarshan Varma, "On composition of SOAP based and RESTful services", II International Conference on Computer Science and Network Technology, IEEE.
- [3] K Munonye and P Martinek, "Performance analysis of the Microsoft.Net and Java based implementation of REST web services", IEEE.
- [4] Zhixiang Niu, Cheng Yang, Yingya Zhang, "A design of cross-terminal web system based on JSON and REST", IJSCE.
- [5] HaiTao Wang, BaoXian Jia, "Research based on web development of Spring Integration Framework", IEEE.
- [6] Yuxiang Hou, "Design and Implementation of the Framework for Spring+SpringMVC+MyBatis in the Development of Web Applications", International Conference of Information Technology, IEEE 2011.

- [7] Zhang Zhenyou, Gu Wei, Cao Zhi, "Research and implementation the system integration on distributed heterogeneous database based on the Hibernate," Network and Communication, pp. 43-46, Aug. 2009.
- [8] Qinglin Wu, Yanzhong Hu, Yan Wang "Research on Data Persistence Layer based on Hibernate Framework," Computer Engineering and Design, vol. 29, pp. 4245-4248, June 2008.
- [9] Budi Kurniawan, Struts2 design and programming: a tutorial, BrainySoftware.com, 2008.
- [10] Zhang Xi, Yu Shuju, "The research on interoperability of Spring framework," Computer Knowledge and Technology, pp. 1-5, Mar. 2008.
- [11] Song Hanzeng, Shen Lin, "Simplification of Java database access with Hibernate service," Computer Applications, vol. 23, pp. 135-137, Dec. 2003.
- [12] Wang Junhao, Peng yan, "Application of Hibernate+Struts+Spring integrate technology in E-government," Computer Engineering and Design, vol. 29, pp. 1409-1412, Mar. 2008.
- [13] Zhang Haiyang, Zhu Yongzhi, "Research of Hibernate +Struts+Spring integrate technology," Computer Knowledge and Technology, vol. 5, pp. 2443-2446, Sep. 2009.
- [14] Gao Ang, Wei Wenxue, "Application of Java data persistence with Hibernate and Struts framework," Computer Applications, vol. 25, pp. 2817-2819, Dec. 2005.

