



INTERNATIONAL JOURNAL OF ADVANCE RESEARCH AND DEVELOPMENT

(Volume2, Issue5)

Available online at www.ijarnd.com

The Necessity and Best Trending Approach For Porting a Desktop Application to a Restful Web Service

Aishwarya N. Kumar¹, Dr. T. H. Sreenivas²

¹ PG student, Department of Information science and Engineering, National Institute of Engineering,
Karnataka, India

² Professor, Department of Information science and Engineering, National Institute of Engineering, Karnataka,
India

ABSTRACT

A desktop application is one that typically runs as a standalone application in a laptop or on a desktop computer. It does not require internet connection availability or a web browser. They must be installed on your computer to access it. Though this desktop application provides high security, they have various disadvantages such as the same user cannot access his software from a different place, a collaboration of the data between the users is difficult since the data is not centralized etc. To overcome these drawbacks, an approach is explained in the paper which builds a RESTful web service considering the already available desktop application and its basic structure. Thus, eradicating the limitations of the desktop application. REST is an acronym for "REpresentational State Transfer". They are used to build service of the web that is lightweight, maintainable and scalable in their nature. The built service on REST architecture is known as the RESTful service and the underlying protocol for this REST is the HTTP, the basic web protocol. The approach may be followed by the developer to build a RESTful web service from scratch. The best available software are mentioned for the storing of the backend database, testing the developed methodologies, the use of following the models such as waterfall model and agile model, the architecture that the framework follows, the servers used, the hardware and software requirements for the development of the software and brief idea about the development and accessing of the front-end using the most suitable technologies.

Keyword: Desktop Application, RESTful Web Service, HTTP, Standalone Application, Waterfall Model, Agile Model.

1. INTRODUCTION

The generic clients such as the web browsers allow us to access the arbitrary services. Initially, the web browsers were mainly used to view the documents and acted as the standards of the world-wide-web which were document centric. These web browsers invoke an operation at a web server via a standard protocol namely CGI (Common Gateway Interface). The easiest way for developing web service applications is by using technologies such as HTTP, java, and CGI. The novel idea behind the idea in this paper is easy portability and building a bridge between desktop and web applications by introducing a new framework to build a web application on the basis of a developed desktop application. To achieve this, we need not make any changes to the already developed web application. The biggest advantage of this approach is that it is easy to port the legacy desktop applications to that of the web application. The basic Graphical User Interface (GUI) of a desktop application is shown in the figure below.

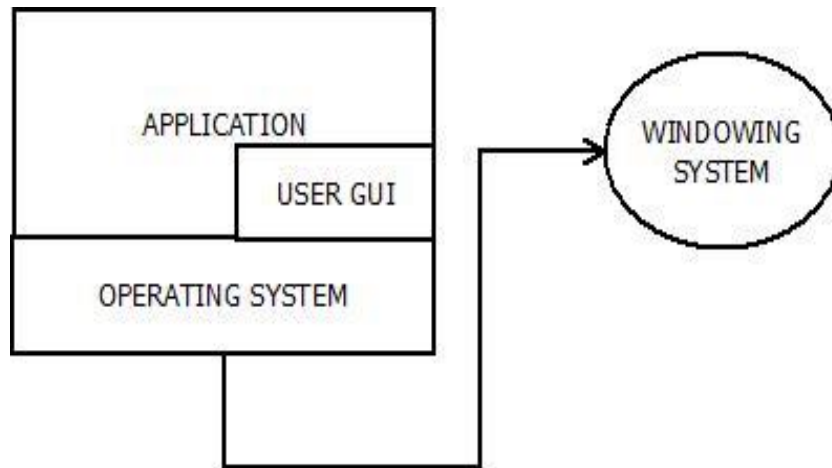


Fig -1: GUI of a Desktop Application

The operating system and the GUI library are the basic factors to be considered while porting the application. The API (Application Programming Interfaces) are the interfaces considered. A specific API to the application is exposed by the GUI library. [3]

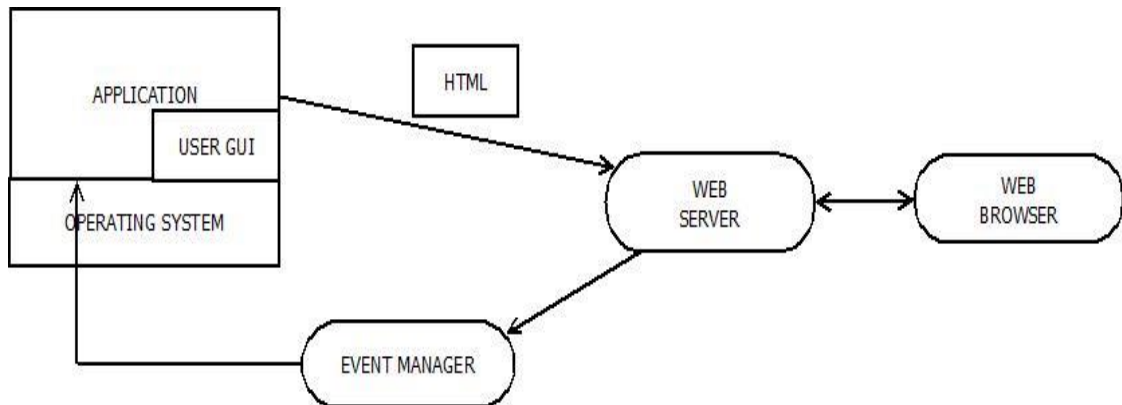


Fig -2: Structure of a GUI Application for Web Application

Investment savings is high along with the easy migration of the existing desktop application to the web application. APIs are the software interfaces whose main applications include the optimization development of an application and service for the data processing. We developers need to follow the best available standards and practices to rightfully develop, publish and consume the APIs. Here we consider developing a java application. They rely upon a basic structure that is dedicated to a programming language, the API must be built for Java, JSON (Javas Script Object Notation). Apart from the usage of the programming language, the API also needs to consider the nature of the service itself i.e certain rules needs to be followed for client-side versus server-side consumption. Here the high performance and secure processing must be our main concern. Regardless of the usage of the APIs in various applications, today’s developers give more importance to the RESTful APIs because of their versatile nature and flexibility. For the integration purpose, the web-service-based software architectures via a JSON follow the SOA (Service Oriented Architecture) model of RESTful APIs. In short, they are consumed by a web server or internet browser. The consumers use these web-based RESTful APIs as a logical solution for the remote consumption of the services like data processing. [3]

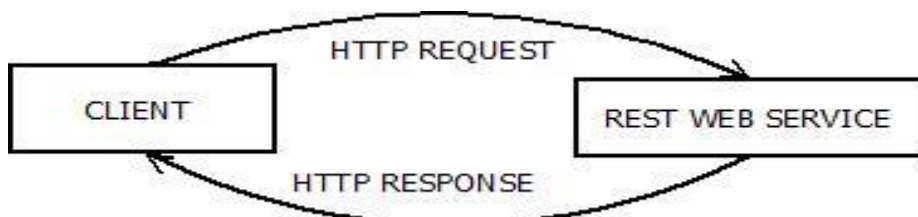


Fig -3: RESTful web service

The most important REST principles for the development of the web application is that the interaction between the client and the server is stateless. The each request sent from the client to that of the server must contain all the necessary information to understand the request sent. Additionally, the stateless requests are free to be answered by any available server, a good example is that of an environment like cloud computing. For improved performance, the client can cache the data. The functionality and application state is divided into resources on the server side. The key element in such an application is the 'Resource', which can be defined as an item of interest, it is exposed to the clients as the conceptual identity. Each of these Resources is uniquely addressable by a URI, which is an acronym for Universal Resource Identifier. The 'Request Verbs' describe what do you want to do with the Resource. The most basic verbs are GET, PUT, POST, and DELETE. The 'Request Headers' are the instructions that are additionally sent with the Request. 'Request Body' is the data sent with the Request. It is usually accompanied with the POST verb. The 'Response Body' is considered as the main body of the Response. They return a document with all the required details. The 'Response Status' codes are the general codes which are returned along with the Response from the web servers. The Verbs used are: 1) POST- For creation 2) PUT- For update 3) GET- Receive data and 4) DELETE – Delete details. [5]

2. LITERATURE SURVEY

Various web application methodologies exist for the development of a good RESTful web service. They are a) HTML Centric: In this method, the HTML page that has been already rendered by the browser is created by a program.

b) Library-Centric: In this method, a library encapsulates the creation of a particular HTML page.

c) Template-Centric: In this method, a template that contains the HTML and the program code, creates the HTML page that has been rendered by the browser. [3]

2.1 Advantages of web services

The following are the advantages of using a web service:

- a) Developing a web application avoids the burden of deploying the software in each client machine.
- b) We do not have to enforce version check in the client machine, as it will be done automatically.
- c) Updates are easier in web based applications.
- d) Fixing the bugs is much easier.
- e) These web-based-applications can be accessed anywhere.
- f) Web applications are platform independent.
- g) Maintenance and providing support for a web application is easier.
- h) It is adaptable in the mobile applications.
- i) It provides an easy collaboration between users if necessary. All data are centralized.
- j) There are fewer chances of finding restrictions that the client computers may have imposed. [2]

3. DRAWBACKS ON THE PRESENT DESKTOP APPLICATIONS

In the building of a web-based-application like a desktop application, there are certain disadvantages. They are:

- a) Access is typically accomplished by running an executable program on the user's computer.
- b) The number of audience in such a desktop application is limited only to the individuals who are sitting in front of a computer/laptop and looking at the screen.
- c) An application belonging to a desktop application designed for offline use relies on its own protocols for user interface and program execution.
- d) In such cases, a developer must spend his time in coding the program elements that are already existing in the web browsing products.
- e) Desktop applications are usually personalized, the more personalized an application, the more programming a developer is expected to do and the toolkits such as GTK (GIMP Toolkit) or QT or any other generic design elements are not best explored.
- f) A crucial disadvantage of such application is that the speed at which the software upgrades are deployed. Each user needs to obtain the upgrade and install it on his computer. [3]

4. PROPOSED METHOD

In this paper, a method is proposed for the development of a web-based-application which overcomes all the drawbacks of a desktop application. Here, the programming is taken to be done in Java language, as Java is one of the most versatile programming languages with many inbuilt applications which can be taken advantage during the development of the application. To build such an application we need to consider few basic hardware and software requirements during the development of such web-based-applications. An application was developed for the below-mentioned requirements. However, the configurations and versions may vary from user to user.

4.1 Hardware requirements

The hardware requirements include:

- a) HARDWARE: RAM-8.00 GB
- b) PROCESSOR: 2.30 GHz

4.2 Software requirements

The software requirements include:

- a) OPERATING SYSTEM: Windows 7
- b) SYSTEM TYPE: 64 bit OS
- c) IDE: Eclipse Luna
- d) Java Development Toolkit (JDK)
- e) SERVER: Apache Tomcat server
- f) OBJECT RELATIONAL MAPPING (ORM): Hibernate
- g) STORING DATABASE: PostgreSQL
- h) BUILDING HTTP CLIENT: Postman Client
- i) SERVER/CLIENT SIDE SCRIPTING: NODE JS/Angular JS
- j) TESTING: PowerMockito

The hardware requirements, operating system, 64bit OS and their configurations above mentioned helps us in developing the desired application with the best performance.

The ECLIPSE integrated development environment is adopted here as it is most widely used Java IDE and it contains a base workspace along with an extensible plug-in system for building and customization of the environment. All the required maven dependencies need to be imported into the developed project. The base classes of this method are the POJO (Plain Old Java Object) classes which are Getters() and Setters(). They are used to describe the same things as that of a 'normal class'. It consists of a Settings.xml which is used to configure maven execution, web.xml provides interaction between servlets and server and text-servlet-context.xml is an object of servlet context created by web container at the time of deploying the project which is used to get configuration information from web.xml. [2]

JDK is a software development environment that is used for developing Java applications and applets. It basically consists of JRE (Java Runtime Environment), interpreter/loader, a compiler, an archive, a document generator and other tools. [2]

Apache Tomcat is best used to deploy Java servlets and JSPs (Java Server Pages). Building the WAR (Web Archive) file is done using Apache Tomcat. It is basically an HTTP server, serving the HTTP.

Hibernate is used for Object-Relational-Mapping (ORM) solution for Java. It has the characteristics such as association, inheritance, and polymorphism which allows us to develop persistent classes following the object-oriented idiom.

PostgreSQL is one of the most powerful database systems. It offers various disadvantages such as much better support than many of proprietary vendors, immunity to over-deployment, extensibility, reliability and stability. It is mainly designed for high volume environments. Data is written in JSON (JavaScript Object Notation) format. JSON is a minimal readable format for structuring data used primarily to transmit data between a server and web application. It is used as an alternative to XML.

The postman client is the best available tool to access the RESTful service endpoints such as GET, DELETE, PUT and POST. In order to make a basic 'GET' request, we have to just select the 'GET' method and add URL (Universal Resource Locator) parameters in postman client. Similarly, we make a basic 'PUT'/POST' request, to modify the already created data or to post a new data. A basic 'DELETE' request is used to delete a parameter.

AngularJS is a fully featured framework which is used for creating a SPAs (Single Page Applications). It promotes all code which is modular, reusable, can be tested easily and also includes SPA routing system. [2]

NodeJS uses a non-blocking input-output, an event-driven model that makes it a very efficient, lightweight, perfect for data-intensive real-time applications that run across distributed devices. NodeJS can be used along with angularJS for the better development of the application. Since Angular is used for client-side application, there are chances of data getting tampered which are not the case when we use nodeJS. Also, a hosted database is not necessary when we are using nodeJS.

PowerMockito is an open source framework for Java. The framework allows us to create and test mock objects in an automated unit test cases for the purpose of Test-Driven-Development (TDD) and Behavior-Driven-Development (BDD). This framework is basically used to check the integrity of the developed code.

4.1 Basic modules for development of RESTFUL web service

The basic modules for the development of RESTful web service can be classified into 3 modules. Namely,

- a) The VIEW module: Consisting of User Interface/Presentation layer.
- b) The CONTROLLER module: Consisting of the Business logic layer.
- c) The MODEL module: Consisting of Data access layer.

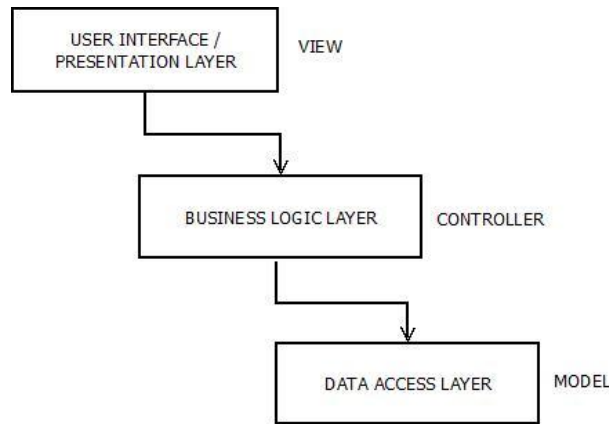


Fig 4: Basic modules for developing a RESTful web service

User Interface is basically the implementation of the intended user experience, which can be considered in terms of a window layout, window transition and window control elements. The Presentation layer is the term that is used in the taxonomy of the code development and its associated resources. They together sum up to the ‘VIEW’ module of the MVC (Model-View-Controller) architecture. [2]

The Business logic layer constitutes to the ‘CONTROLLER’ module in the MVC architecture. It takes care of all the functionalities of the Controller. It co-ordinates between the View and the Model and also handles the request sent by the user. The user’s request is received by the Controller which requests the Model to perform the basic CRUD (Create-Read-Update-Delete) operations, the results are then sent to the user. The Controller class inherits certain properties from the base class. [2] They also provide its own useful properties. They are

PROPERTY	USE
• REQUEST	The current HTTP request is referenced.
• RESPONSE	The current HTTP response is referenced.
• SERVER	Provides useful methods for request handling.
• SESSION	The current session is referenced.
• ACTIONINVOKER	The Action Invoker for the Controller is referenced.

The ‘MODEL’ layer is responsible for managing the backend data of the developed application. Its main functionality is to respond to the incoming request from the View and respond to the instructions from the Controller to update itself. It solves the data that is retrieved via the commands received by the Controller and displayed in View. [2]

4.2 Basic layers for development of RESTFUL web service

The figure below illustrates the basic layers required for the development of a RESTful web service:

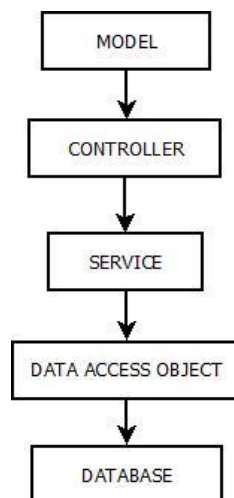


Fig -5: Basic layers for developing a RESTful web service

REST layer provides the interoperability between the computer systems on the web. In a RESTFUL web service, requests are made to a resource's URL, which will return a response that may be in JSON, HTML, XML or any other compatible format. SERVICE layer acts as an interface layer between the User Interface and the backend systems that store the data, it is responsible for managing the business rules of transforming and translating the data between the two layers. The main idea behind introducing such a layer is to have an architecture that can support the multiple Presentation layers in the development of a RESTful web service. DAO (Data Access Object) interface layer basically defines the standards to be performed on the model objects. They help in separation of the high-level business services and low-level data accessing API. It is responsible for getting the data from the database. FUNCTION layer defines the basic functionalities that are used in the operation of the various objects between the DAO and the Database. It provides a separation between the DAO and database. The DATABASE layer is responsible for all the backend functionalities of the component. It manages the entire database and the functionalities mapping to those elements.

4.3 Proposed 5-layer 3-tier architecture for building a RESTFUL web service

A new 5-layer 3-tier architecture is proposed in this paper following the above-mentioned methodologies, techniques, and underlying concepts. The architecture is clearly distinguished into 5-layers and 3-tiers. The CLIENT TIER constitutes to the REST layer. It consists of the automated clients which include the Java web service client and browser client. They run inside the browser and are the consumers of the RESTful web service, they act on behalf of the users. The Request Resource Handler (RRH) can be implemented using JERSEY or RESTEASY which constitutes to the RESTLET that are lightweight and must have the capability to handle the heavy duty of business tier. HTTP requests sent by the automated web service clients to the RRH in the client web tier. The requests are stateless, they contain the method information in the header which are either POST/GET/PUT or DELETE. These incoming requests will be mapped to the corresponding operations of the resources in the RRH. They can be built and tested using the postman client framework. They contain all the necessary information along with the credentials needed for the RRH for processing the request. Once the request has been received from web service client, the RRH requests a service from the underlying business logic layer.

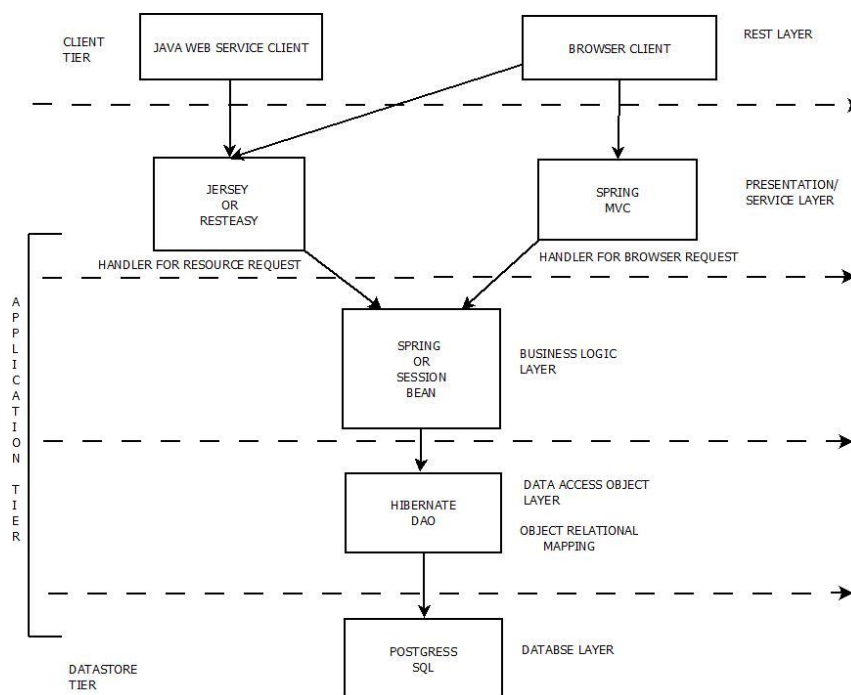


Fig -6: 5-layer 3-tier architecture for building a RESTFUL web service

The Presentation/Service layer uses the concepts of spring MVC and RESTLET, Hibernate DAO layer uses the hibernate framework and Business logic layer uses the concepts of spring/session bean which together constitutes to the APPLICATION tier of the architecture. The DATASTORE tier consists of the Database layer which can be implemented using the POSTGRESQL, it is used for the storing and retrieving the elements into the database.

The web browser client in the CLIENT tier acts as the GUI (Graphical User Interface) for front-end provisioning display function using the HTML generated by the Browser Resource Handler in the Presentation layer, which is implemented using the MVC models. [6] The front end/View module of the Spring MVC and their interactions can be implemented using the concepts of AngularJS and NodeJS together.

4.4 Handling of Request using the Spring MVC

The figure illustrates the basic handling of the request in the Spring MVC framework.

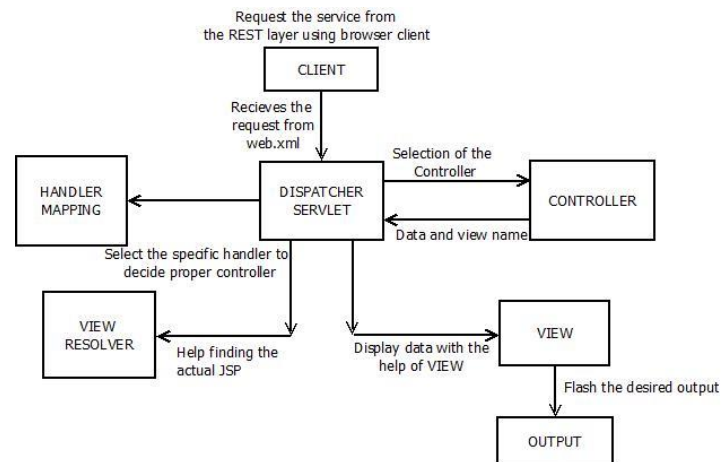


Fig -7: Handling of Request using the Spring MVC in a RESTFUL web service

The steps include:

- The client sends the request for the service from the REST layer using the browser client.
- Dispatcher Servlet received the request from the web.xml.
- The Controller handles the request received which is created by the client.
- The main functionality of the Model and View is that when a request is received, it is the job of Model and View to associate the view to that particular request. It is created by the Controller and when it is executed, it returns data and name of View.
- View Resolver tries to solve the View based on the output given by the Model and View. It selects the output media via AngularJs and NodeJS.
- The main functionality of the Handler Mapping is that whenever the Dispatcher Servlet receives the incoming request, it associates the request to individual Controllers with the help of this component.
- The View flashes the desired output on the client's screen.

CONCLUSION

The paper concludes that the proposed methodology of developing a RESTful web service is the best, safe and secure method for developing an application and it focuses the limelight on the necessity for porting a desktop application as a RESTful web service. All the necessary frameworks along with the hardware and software requirements are mentioned in the paper, using which a highly efficient web application may be developed taking into considerations all the available frameworks. The disadvantages of using a desktop application and the advantages of porting such a desktop application to web-based-application are clearly mentioned. By following the techniques mentioned in the paper, highly effective applications may be built in various fields such as education, medical and banking domains. [1]

A newly proposed 5-layer, 3-tier architecture provides a clear distinction between various layers and how incoming requests are handled in the architecture. Powerbocking framework can be used to check the integrity of the developed code by the concepts of mocking and stubbing. [10] The paper acts as a basis for both the web services and dynamic web applications and the developed applications is known for its simplicity, extensibility, reusability, has a clear separation of its components along with being a sturdy application. Developers looking for most trending and best technologies in developing the desired web application or in porting a desktop application into a RESTful web service may be highly benefited by the proposed method. [1]

5. REFERENCES

- Aishwarya.N.Kumar, Dr.T.H.Sreenivas,"A Survey on Porting a Desktop Application as a RESTful web service", International Journal of Recent Trends in Engineering & Research, Volume 3, Issue 4, April 2017.
- https://en.wikipedia.org/wiki/*.
- Arno Puder, "Extending Desktop Applications to the Web", www.puder.org/publications/dorea04.pdf, 2012.
- Larry Simmons," The Disadvantages of Building a Non-Web-Based Application", <http://smallbusiness.chron.com/disadvantages-building-nonwebbased-application-40347.html>

- [5] RESTful web services, “What is RESTFUL web service”, <http://www.guru99.com/restful-web-services.html>.
- [6] Bruce Sun, “A multi-tier architecture for building RESTful Web services”, <https://www.ibm.com › Learn › Web development, 2009>.
- [7] K.V.Kanmani, P.S.Smitha, “Survey on Restful web services using Open Authorization(Oauth)”, International Journal of Computer Engineering, Vol 15, issue 4, Dec 2013.
- [8] B. Jaya Kaviya, G. SelvaKumar, “A Survey on RESTful Web Services Composition”, International journal of Advanced Research in Computer Science and Software Engineering, vol. 5, Issue 11, November 2015.
- [9] Ritesh Sinha, Manisha Khatkar, Subhash Chand Gupta, “Design & Development of a REST-based Web service Platform for Applications Integration on Cloud, International Journal of Innovative Science, Engineering & Technology, Vol 1, issue 7, September (2014).
- [10] Dave Syer, “Unit Testing with Stubs and Mocks”, <https://spring.io/blog/2007/01/15/unit-testing-with-stubs-and-mocks/>, Jan 15,2007.