**RESTful API development for Ecommerce website using Microservices and Spring Boot.**

**Varun Mehrotra**

School of Computer Science and Engineering, Galgotias University, Greater Noida, India. E-mail:

varunmehrotra2998@gmail.com

**Minal Garg**

School of Computer Science and Engineering, Galgotias University, Greater Noida, India. E-mail:

minal.251298@gmail.com

**C. Vairavel**

School of Computer Science and Engineering, Galgotias University, Greater Noida, India. E-mail:

c.vairavel@galgotiasuniversity.edu.in

**Abstract**

To create the Rest API website using Java, the aim is to reproduce the backend model of the website that is already used by our generation regularly. Examples of one such website are Amazon and the IRCTC website, with a full frontend we will be doing Web Application and Rest API's. As an engineer, we need to think out of the box, plan your development strategy and try to improve the APIs in such a way that they support a large number of features used on the website and maximize results from each API, reduce unemployment and use Microservices Architecture and Spring Cloud to build APIs and -API can be tested with Swagger. The RESTful API uses existing HTTP methods defined by the RFC 2616 protocol. They use GET to get the service; DISCLAIMER or update the app, which may be an object, a file or a block; POST to create that resource; and DELETE to delete it.

The most powerful Web applications in Java, flexible and usable, building integrated Web applications are very important. By introducing the changing style - Representational State Transfer, this paper learned the principles and styles of the REST style, the REST concept and RESTful Web service services, the RESTful Web Service style, the RESTful web services framework in Java and therefore ways to improve RESTful Service of the web. RESTful Web Service frameworks in Java can facilitate internet development in many aspects.

**Keywords:** REST API, Microservices, Spring Boot, Spring Framework, Spring Security, Spring Cloud, Databases, Dev Tools, JSON.

**Introduction**

In today's world with the help of ecommerce web design, you get the opportunity to have your products and services available to customers 24 hours a day. Good exposure to your business. Most people choose to shop online due to lack of time, you can easily make more income and more profit. With the help of an e-commerce website people can choose and buy the products you want anytime anywhere. Payments can be easily made with credit cards or other payment options available on the website. REST (Representational State Transfer), a style of web services design, has become increasingly popular in recent years. Many vendors have opened the doors of their services to developers, giving them relaxed access to a variety of web services. The traditional e-Commerce website is growing in popularity. A large number of product choices, competitive prices and excellent customer care have been driving many customers on the site.

The eCommerce API is a collection of eCommerce applications presented through the API, straight forward. Violation of certain eCommerce functionality includes minimum, cart, opt-out, payment and order - or in other words how to do it on your eCommerce websites. One point of confusion that is sometimes helped is that the payment gateway is a commercial API, where in fact these providers offer a single service, namely a payment API, eCommerce is a comprehensive journey with one or more steps.

REST (Representational State Transfer) is a style for creating designs for non-compliant web services. It is mainly used to develop simple, fast, warm and easy to maintain web services that often use HTTP as communication channels. REST is not connected to any particular platform or technology.  
  
REST defines the Web as a distributed hypermedia program RESTful applications use the HTTP application to send data (create / update), read data (query) and delete data. Therefore, REST uses HTTP for all four CRUDs (Create / Read / Update / Delete).

In a headless commercial architecture, eCommerce APIs act as connections between frontend (head) and backend (body).

APIs allow you to interact with the concept without having to know what's going on under the hood. This is great for eCommerce websites because there is a tone of concept that you need to achieve but at the same time you don’t want to face the difficulties that underpin the whole concept. This creates a natural distinction between your front and back operations and a basic commercial engine. This turns ordinary engineers into highly paid engineers due to good performance and productivity. You no longer need a full stack or an expensive eCommerce platform builder. This allows you to use

new business solutions and get to the market quickly while enjoying access to a wide pool of developer talent.

"We use tons of tools that go with the spring framework and reap the benefits of finding solutions in the box, and we don't need to focus on writing any additional code - to save us time and energy."

Spring Boot is yet one more popular Spring project. There was a feature request to enable container-less applications using Spring Framework. The Spring Developers developed a whole framework round the feature request, and named it as Spring Boot.

To run Spring-Boot you don’t got to deploy the project into a Container. Because, Spring Boot comes with an inbuilt container and every application has one. Similarly, each Spring Boot application features a main method which bootstraps spring boot components, container and other framework components.

Spring Boot is sensible , if you don’t provide certain configurations, Spring Boot uses defaults and starts. as an example , albeit you don’t provide a database driver and respective configurations, Spring Boot will start with an in-memory database. However, once you provide your own configurations, Spring Boot will respect it and remove the auto-configurations for the respective components.

Spring Boot application are bootstrapped from the most method. The way you run a Spring Boot application locally is that the same as you are doing for a Production Box. additionally , having Strong properties management mechanism provides a structured thanks to externalize environment specific properties making it easy to deploy an equivalent jar to any box. It only needs JRE to run a Spring Boot Application.

The Auto Configuration is one among the foremost important features of Spring Boot and looks like some magic to several newbies. Having Auto configurations, saves tons of framework and component initialization.

Spring's flexible and extensive extension set and third-party libraries allow its engineers almost any application you can think of. At its core, Spring Framework’s Inversion of Control (IoC) and Dependency Injection (DI) provides inspiration for a comprehensive set of features and functionality. Whether you’re building secure, efficient microservices, cloud-based internet, or complex business data flow, Spring has some helpful tools.

Spring Boot changes the way you perform Java application tasks, completely simplifying your experience. Spring Boot incorporates requirements such as the context of the application and the web server configured automatically to build a microservice cinch upgrade. for greater speed, you will integrate Spring Boot with a rich Spring Cloud support set of libraries, servers, patterns, and templates, to securely unlock microservices-based buildings in the cloud, in time record.  
  
Spring includes a proven diary to handle safety issues quickly and responsibly. Spring workers are working with security experts to mark and evaluate any reported hazards. Third party reliability has also been extensively tested, and regular updates are released to help keep your data and apps as secure as possible. in addition, Spring Security makes it easy for you to integrate with industry-standard security systems and provide reliable and secure security solutions.

The spring community is vast, global, diverse, and open to people of all ages and abilities, from perfect beginners to timely beauties. Wherever you are on your journey, you will find support and resources that you would like to promote at the next level: instant stars, directions and tutorials, videos, meetups, support, or formal training and certifications.  
  
**Related Work**

Request actions define what needs to be done with the information available from the server. The browser uses the GET application to command the system to retrieve or download data from the server. There are various application actions that can be used outside of GET namely GET GET POST DELETE So in the example http://varar.com/employee/1, the browser actually uses GET Verb because it wants to get the details of the job record.

In the header different additional information is sent along with the data these articles also contain a different format where the data needs to be downloaded. The articles also have a variety of authorization and authentication methods used.

This way, the sender sets the username: password to the application title. Username and password are set by Base64, this cipher encrypted value is converted using Base64 and this encrypted device prevents separate attacks and is secure to transfer. APIs using Basic Auth will also use HTTPS, which means that the content of the message will be encrypted within the HTTP transaction protocol. When the API server receives a message, it deletes that message and checks the subject. After determining the thread and analyzing the username and password, then decide whether to accept or reject the request.

Related activities in the stock market, we identify ways to help calculate the prediction of stock exchange problems. the first phase of the process is connected economic simulation, which includes the Hellenic economics theory of speculation. Common methods are automatic process (AR), MA, ARMA, therefore, autoregressive-integrated-moving-average (ARIMA) (Lathuiliere, Mesejo, Alameda-Pineda, & Horaud, 2020). In other words, this hypothesis captures all untested signals such as the reduced haplotype of previous signals and sound independent sets. However, many people believe in certain strong ideas when it comes to the functions of losing a sound word. within the second class, we have soft computer models. Soft computational can be a computer term that mimics biological processes. These methods adopt the methods of NN, SVM, PSO, and a few others. Several investigators have tried to disrupt the light randomly in price models. Dang Lien Minh, Huy, Min, & Moon (2018) proposed a TGRU stock forecasting model where they need to use the database between October 2006 and November 2013, which they receive from yahoo finance.

Reciprocal Communication usually means more correspondence or cohesive exercise between at least two members of the framework. With regard to REST, we focus on customer communication and the Restful Web API, i.e. multiple resources. Conversations between customer and multiple resources, what we call Restful conversations, can be described as follows.

The NVIDIA kit with Keras interface version 1.2.3 includes Python version 2.7. The hyperparameter used in the model was trained with 35 epochs, cluster size n = 64, print rate = 0.0001, and decay rate t = 0.00001. They obtained a total accuracy of approximately 66.32% of the proposed model.

Yujie Wang suggested a Hybrid Time-Series NN (HTPNN) view of the stock market Conjecture. need to use yahoo financial database. within the HTPNN model, they need to use a 2-convolution layer, a 2-LSTM layer, a reading rate of 0.005, and use a 1000 iteration. Test results showed that the model obtained 69.51 Accuracy (Wang, Liu, Guo, Xie, & Zhang, 1970).

There are two types of people in a relaxed conversation. Clients communicate with the API to achieve a specific target. Resources are the squares of each Soothing Web API structure; they provide a visual interface that allows them to find and adjust their status. Participation between a client and an API can see the creation or removal of its benefits, or in the recovery and renewal of its benefits.

An application in the system must have only one valid URL, and that should provide a way to download related or additional data with the same interface and make it easier to process with different applications and different frameworks.

Idrees, Alam, & Agarwal (2019) have proposed the ARIMA model, which may be a non-existent way to predict the future values ​​of your time series data in the Indian Stock market; they require the use of Indian stock exchange data from Jan 2013 to December 2017. Performance is considered by examining the performance of the index generated by the index so the Indian market has two main indicators called Sensex (contains 30 shares) and Nifty (contains 50 shares). AR (p) represents a “autoregression” model influenced by a variable ‘p’. Subsequently requesting the error name in the AR model to be considered, the "average average" MA (q) model was used, looking at the error of the previous model, but this made the task difficult to match the model with time series data. Finally I used the ARIMA model, which prefers remote statistical timeline data. After further decay of information and further analysis, the LJung Box Test applied Nifty and Sensex p-value results to 0.9099 and 0.8682, respectively.

The body of the respondent is the body or details of the response we receive from the request being sent. So, in our example, if we were to query a web server with http://varar.com / employee / 1, the web server could retrieve the XML document and all employee details to the Responsive Body.

Response status code is those codes that indicate the status of a response from a web server. There are different codes that indicate the status of each response. For example, code 201 indicates that the new application was created on the web server without error.

Restful API is a simple, flexible and secure API built into the REST architecture. Restful Architecture, suggests APIs from the client application in countless ways. The client can perform different tasks using the Restful service.

**Architectural Constraint:**

Different display

Client - server: Client application and server application MUST be able to appear separately without dependence on each other.

Countless: Make all customer-server connections blank. The server will not store any data about the latest HTTP client client that has been created. It will handle each new request. There should be no time, no history.

Data storage and responses are very important. It brings improved customer performance, as well as a better scope for server robustness because the load is reduced.

Systematic: Allows you to apply the layout of a layered system.  
  
**Model View Controller**

MVC (Model View Controller) is straightforward. After creating the website, the following steps are followed by MVC:

1) The client sends the REST keyword request to the server.

2) The server is the same and responds to the requested URL in the administrator action.

3) Returns results to the client for processing in the JSON format. Manufactured JSON

in the fourth step it can be thought of as a ‘vision’.

It is an image of the state, which must cross the boundaries of the process. MVC is a way to create separate server code.

REST is concerned about customer contact.

Model-View-Controller (MVC) is a building pattern that divides the system into three logical elements: model, view, and therefore control. Each of these features is designed to handle certain aspects of the app's development. MVC is one of the most widely used frameworks in the web development industry to create awesome and scalable projects.

It may seem like we're in the middle of an Architecture Astronaut now, but bear with me. The ideas of MVC are a touch of reality, of course, but of a surprisingly common pattern. it is actually around you. In fact, let me bring it back to Earth this way: look at MVC immediately.

Simple and inconsistent testing. Highly tested, expandable and connected framework

It gives you full control of your HTML just like your URLs

Use the available features provided by ASP.NET, JSP, Django, etc.

Clear classification: Model, View, Controller. Separation of application functions namely. business idea, Ul concept, and input concept

Route URL for SEO friendly URLs. Powerful URL map for understandable and searchable URLs

Test drive Development (TDD) Support.

**SYSTEM OVERVIEW**

Renew web services provide a style of architecture to build online services and how to use those client api. API’s, developed using the http protocol may not follow all the remaining issues. The purpose of this paper is to install an api authentication strategy. The method checks that the implementation is

developed as required for a specific document of appropriate api. This paper also explores the challenges in analyzing the remaining api and ensuring the implementation of the api. This method will process the OpenApi specification document for the use of the RESTful web api.

The proposed system will be designed in such a way that the analysis of the remaining APIs is usually managed by processing the application because

REST web api documents. The RESTful API scripts are written in swagger 2.0 format. The springfox framework has been used for the production of text from the application. The next section will explain the structure of the proposed system.

**Design**

There will be two much-needed inputs in our system. The first is that a document document is written and the second is a document created in the framework of springfox. There will be 2 active modules available within the system. The first module will generate a document from

code. The second module will compare the handwritten document and test the available spaces. The rules are explained by comparing the Swagger document or the OpenApi document. This comparison will be used to make an analysis report of

api. policies are usually controlled by developers first.

**System construction**

Figure 1 gives an overview of the various components of the system. The architecture has various modules supported by system-compliant functionality. Analyzer and parser are the most common elements in this strategy development. Parser will work on the two specification files used in this method. One of the Modules is developed using the Springfox framework which is an open-source project. This project has been used to automatically create scripts from api applications in the spring framework.

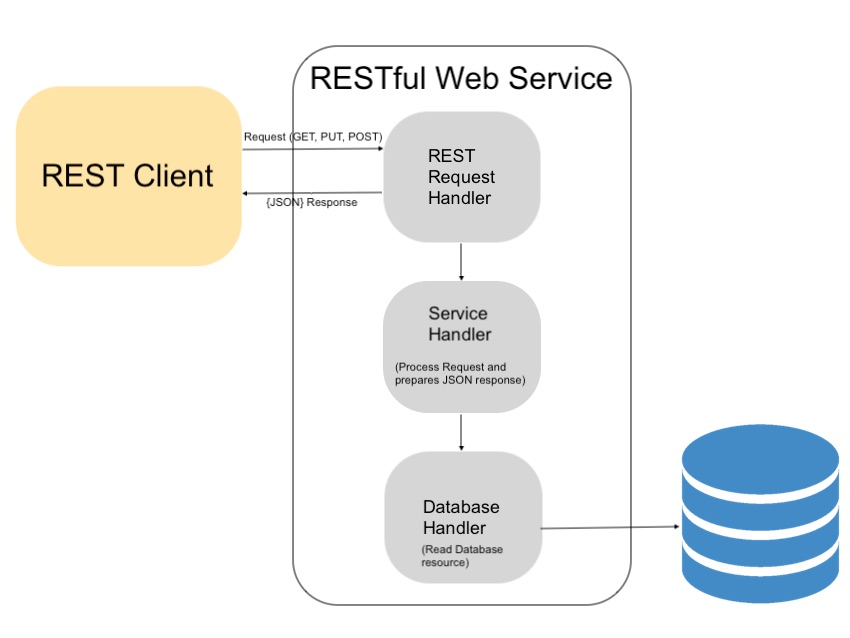


Fig:1.0:How RESTful services Work

This is usually the first module of a project. The program has 2 swagger scripts such as installing One of the open api specification scripts is that scripts that were written before the actual launch of the service. This document provides construction details next to all storage space information and their responses and parameters. Implementation should take place in accordance with this document. The second document namely the swagger document is coded using springfox. This document compares with the original in terms of defined analysis rules to determine whether the application covers all the required information for

app within api or not.

**Software requirements and specifications**

A system designed to monitor consistency between the implementation of a relaxing service and reliance on document documents based on the number of open source technologies. JDK where the project was tested is in JDK 1.8. The reliance on the software is as follows:

* OpenApi 2.0 specification,
* Spring Boot Framework,
* SpringFox Framework,

As the service is developed with the Java language spring boot framework it is the hottest development framework for the REST api.

**SYSTEM ANALYSIS**

The analysis of the REST api is controlled by comparing input to the system. The analysis provides details on the differences within the implementation and therefore the expected documents. Automatically generated document represents api creation. The reports made by the program will look at information about used apis, misuse of api, not using apis and processing but not written apis. there will be 2 types of analysis to be done by the system. The first will give an overview of used or unused apis. And the second analysis report will look at all the end conclusions. In this second analysis the high point has a specific action

will be considered for its uri representation, parameters, side effects and accept mime types. Another major contribution of this paper is the discovery of a gap in literature and practice. If another portion of the written api is not included such a gap is also reported.

**Suggested Method**

We have done the proposed work on windows 10 with Intel Pentium layout. This project is being implemented with the help of the Google Colab server to get better computing power. In the Google Colab server, we used the GPU as the starting point to use the code correctly. We have used libraries such as TensorFlow version 1.x, Pandas, NumPy, Seaborn, Matplot library, Sklearn library, and Cameras.

**A. LIMITATION**

With it can improve the flexibility and system of adopting a growing business and human needs. The built system will withstand failures. Useful for cloud use. Countless items can be freely distributed if something fails. The architectural design considered RESTful or REST style is characterized by:

Status and performance are divided by distributed resources. Each device actually looks special using uniform and small command commands (typically using HTTP commands for GET, POST, PUT, or DELETE over the Internet) Protocol client / server, encrypted, and supports temporary storage. We have various benefits of the REST API: Dragging the function of retrieving files from File fields in (RDOs). Supports creation, read, update, delete, and query functionality. Provides secure access to resources using flexible authentication methods, such as HTTPS. Language stand-alone built-in adherence to RESTful building standards, giving you the option to choose a program language based on application requirements. It involves the implementation of standard HTTP methods and the presentation of the JSON REST API resources following the standard building codes defined by the Representative State State Transfer. It supports all of these key building concepts:

Countless: The REST service does not use login times or store other state information on the server.

Communicating using HTTP: Uses common HTTP methods: a. FIND: reads data b. POST: creating resources c. SET: update resources d. Delete: delete resources

Uses HTTP status codes: Status codes represent the many performance results you can create against the REST service.

Resource Management: The Rest API app represents objects. A unique URL identifies each app. Some businesses are not commonly identified as assets and are also represented as assets.

Providing a hypermedia-driven API: The REST API helps retrieve responses that include links to services available in Relativity. To think about this example: the answer to a workplace question returns a list of similar workplaces. It helps to represent each workspace in a simplified format that contains a link to additional information about the source.  
  
**B. IMPLEMENTATION**

JavaScript is a client writing language used to create web pages and Web applications.

HTML: HTML is widely used in building pages that are displayed across the world wide web.

CSS: CSS is the language to describe the presentation of Web pages, including colors, textures and fonts.

WEBMVC: The Spring Web MVC framework provides Model-View-Controller design and optimized features that can be used to make flexible and flexible web applications. The model enters the application data and will normally have a POJO.

SPRING CLOUD (NETFLIX EUREKA): In this case the client returns a list of all connected peers in the service register and makes all other requests for other services with a load balancing algorithm.

SPRING BOOT: The main purpose of the Spring Boot framework is to reduce Development time, Unit Test and Integration Test and to facilitate the development of convenient web applications.

SPRING SECURITY: Spring Security can be a framework focused on providing both authentication and authentication in Java applications. As with all spring projects, Spring Security’s key strengths lie in how easily they are opened to meet cultural needs.

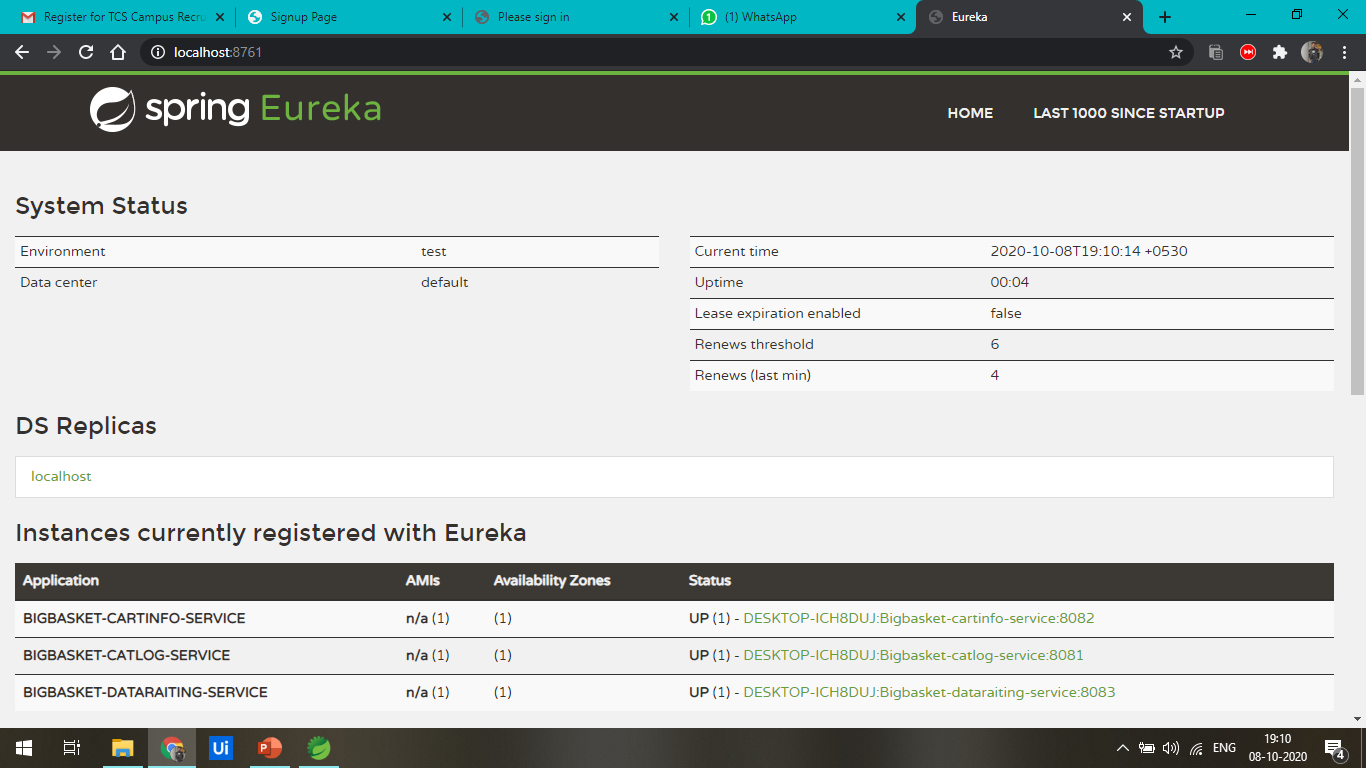
Netflix Eureka: Registration for cloud services using microservices architecture

Fig:2.0:Netflix Eureka Cloud Registration

**CATLOG DIRECTOR:** In the catlog controller we create RESTful services such as create, update, read, and delete. The manufactured JSON contains key words corresponding to the names of the structures of the Catlog category. The controller will manage URI and HTTP methods based on Java (performance) methods. To strengthen communication we use the REST Template**.**

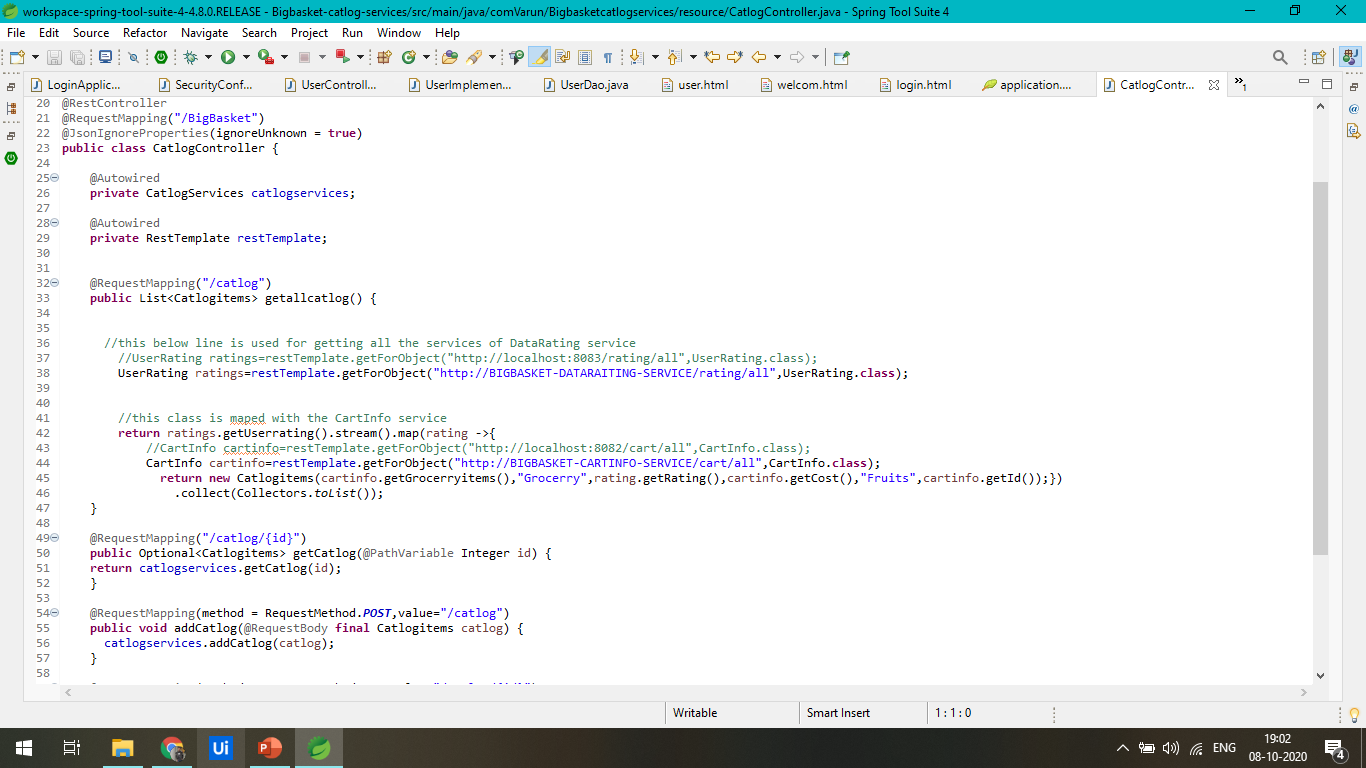


Fig:3.0:Catlog services registration

**H2 DATABASE:** Internal database and Spring Boot server, Fast, open source, JDBC API. Featured methods and server; of memory information. Console based Browser app. Small footprint: approximately 2 MB pot size file.

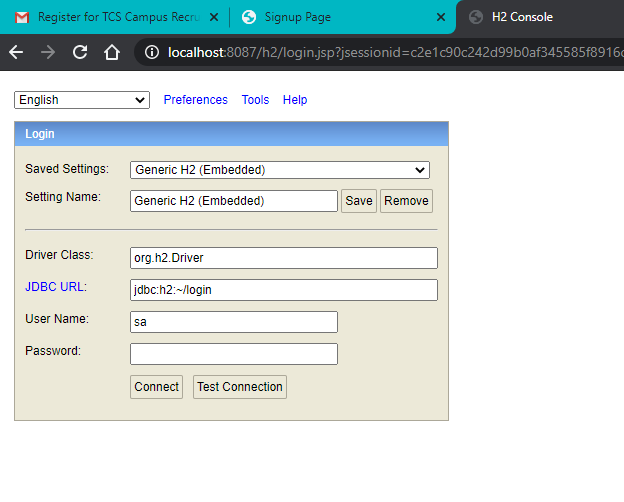


Fig:4.0:H2 Database Registration on server

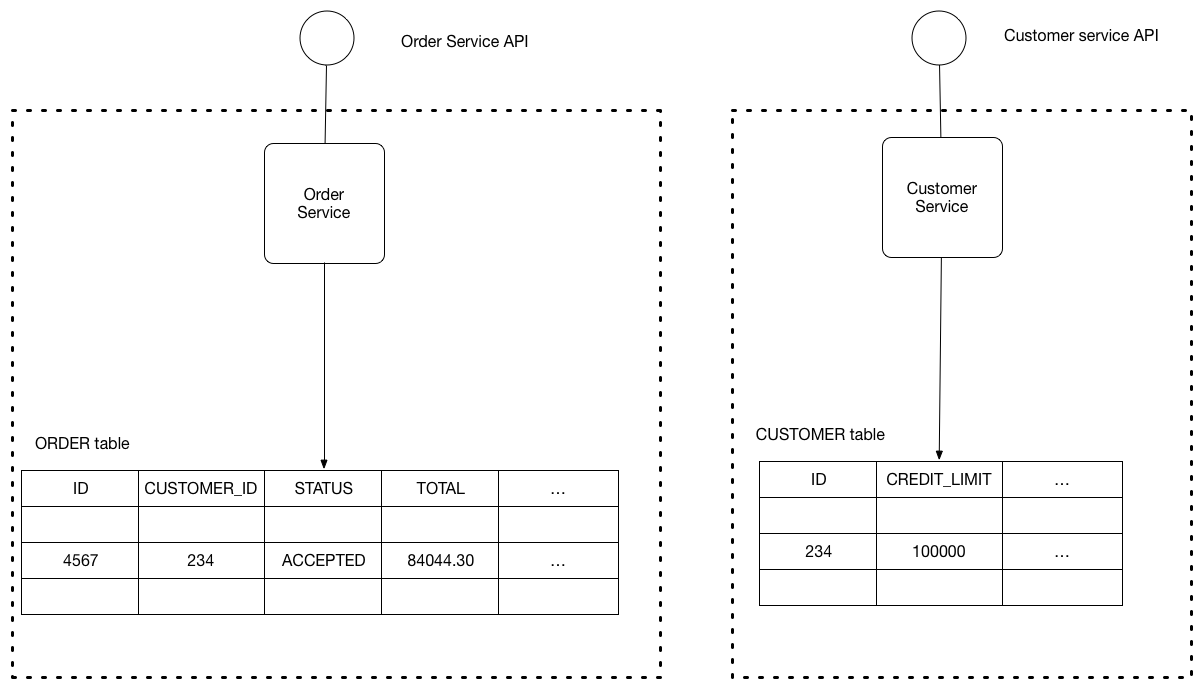


Fig:4.1: Database Structure

* We will keep each microservice data private to that service and accessible only via its API.
* The service’s database is effectively part of the implementation of that service. It cannot be accessed directly by other services.
* For that we can use🡪
  1. Schema-per-service – each service has a database schema that’s private to that service

**PAYMENT SERVICE:** A payment gateway is a technology that captures and transmits payment information from a customer to the acquirer and transfers the payment acceptance or down payment to the customer. The payment gateway verifies the details of the customer card, ensures that funds are available and ultimately enables merchants to make payments. It serves as a link between the merchant's website and its acquirer.

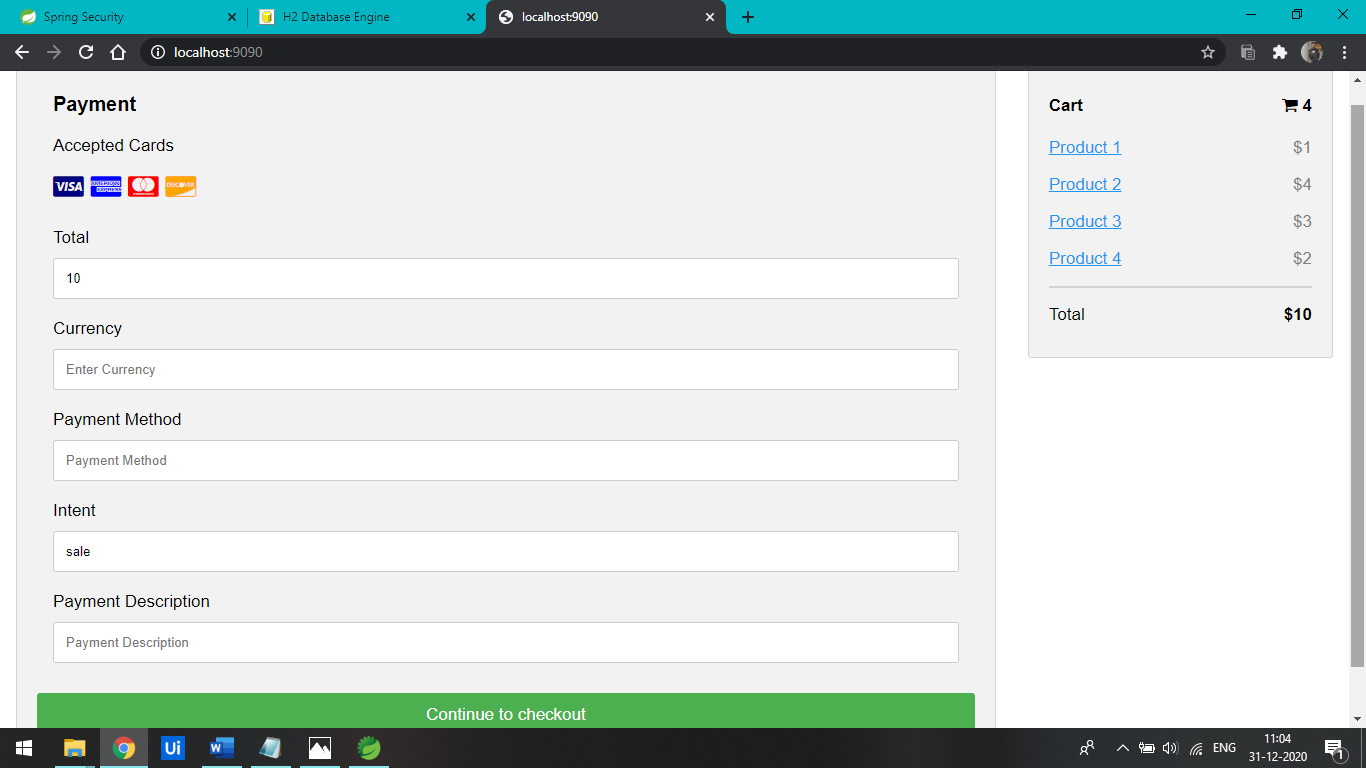


Fig:5.0:Payment Services

**Result Analysis**

This section will discuss two types of data analysis results and the gap identification data generated by this method. The first analysis result can be considered as a summary of the system analysis while the second analysis result is a detailed analysis of each end point. Tracking of results results will be done as the initial analysis.

The second analysis will look at each last point written down and determine whether

the end point used is the same as what is expected of text construction. In this analysis each http action and its details such as header info, of a specific end point will be examined in a document.

No Test parameters

1. Number of Resources

2. Number of Expected Destinations

3. Number of Storage Areas Used

4. Not Expected but Not Performing Final Points

5. Number of Spaces in the System

6. System Percentage completed

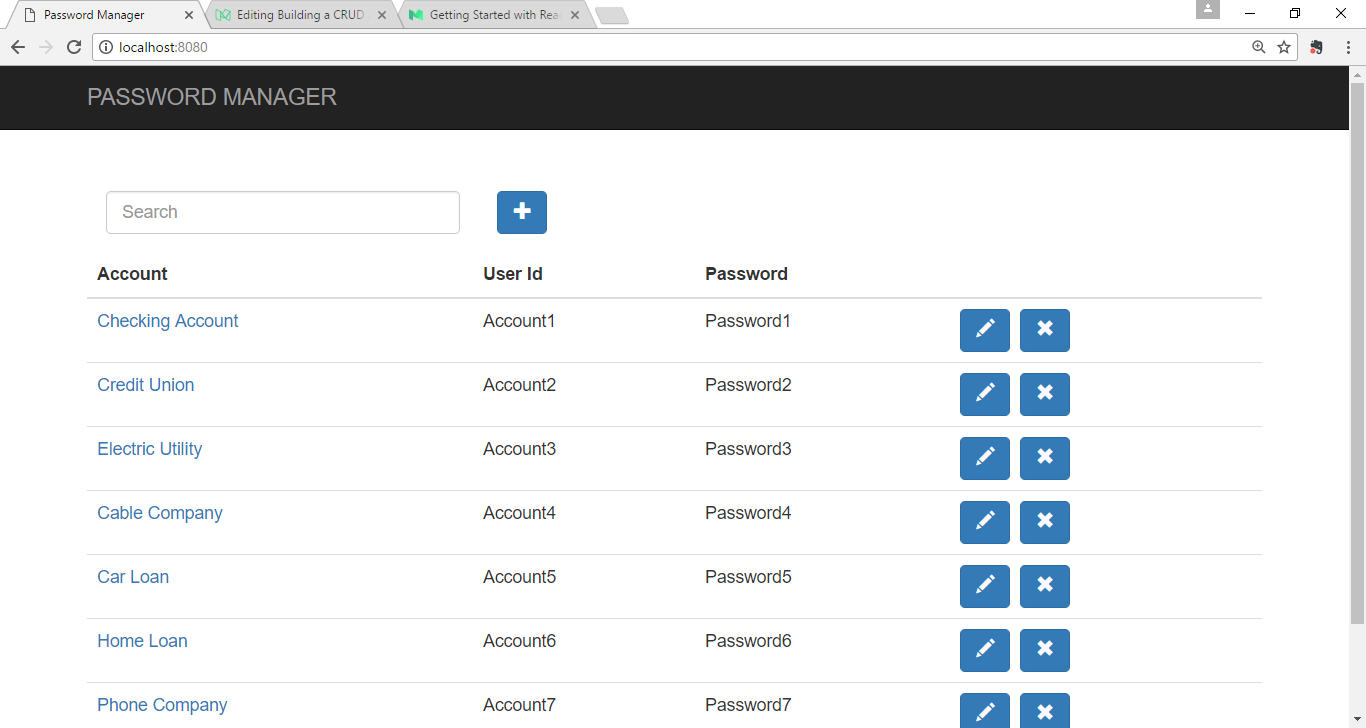


Fig:6.0 Account Register Service

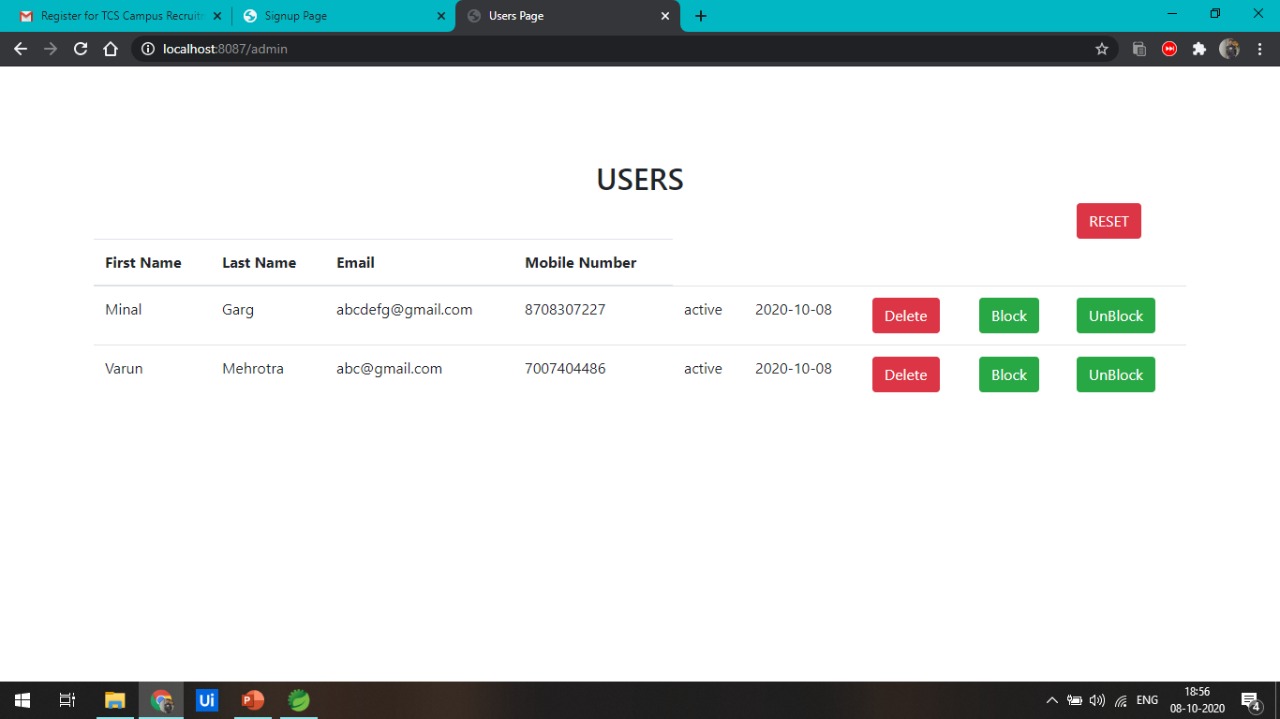


Fig:6.1:Admin Page

This table provides an overview of the application, whether the app is used in accordance with the data documentation requirements or not. Such an analysis is presented with values within the aforementioned results table.

The second analysis will look at each high point written down and see if

the end point used is the same as the type of document expected. During this analysis each http action and its details such as header info, of a specific end point will be examined in a document. The results of this analysis will contain the following properties.

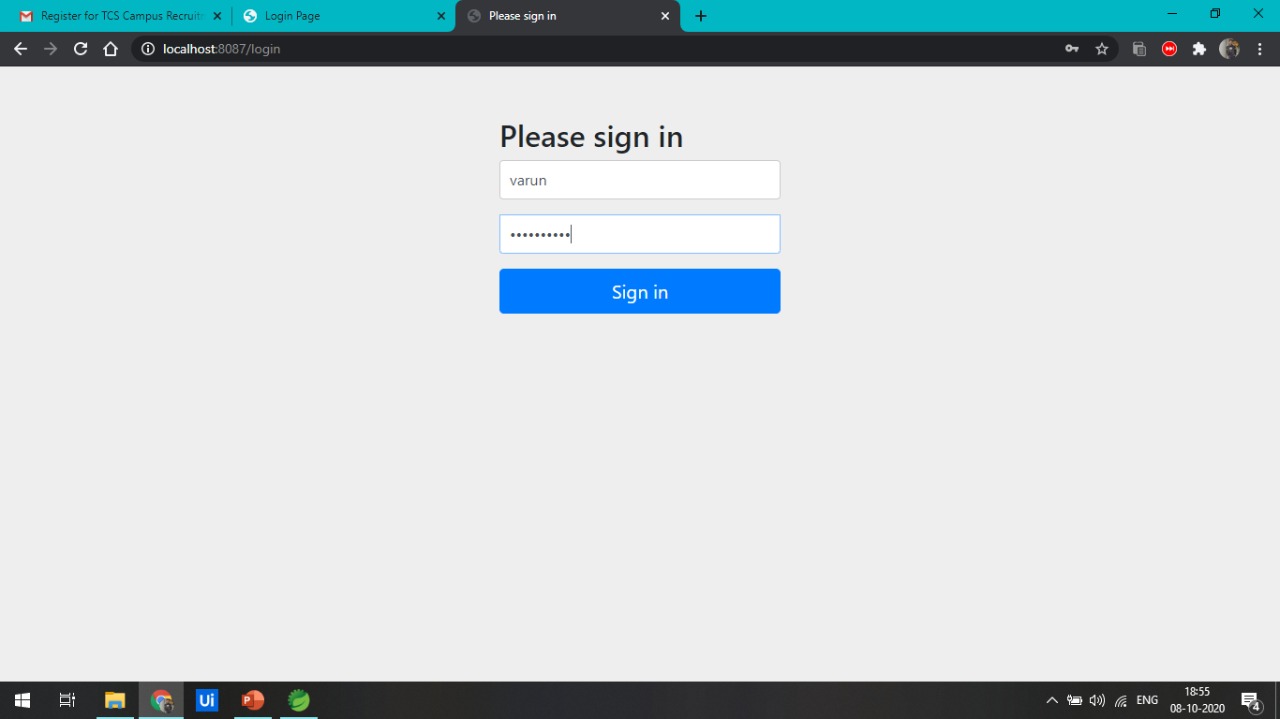


Fig:6.2:Admin Security Verification page

Details. For API Endpoint

1. Expected and implemented parameters

parameters

2. Expected consumes a type of mime too

included eating mime species

3. Expect to produce mime types as well

use produces mime types

4. Expected http Answers and Use of http

answers

**FUTURE WORK**

The current system is built taking into account the version of the fixed 2.0 version. However, the system is often upgraded to get api 3.0 open source data, as this version provides information on multiple servers and much more. The basic design will always be the same. The new update will process the small print that is provided in version 3. Currently, the system is waiting for the activation code, but the function is often extended to run api services.



Fig:7.0: Future use of Microservices Architecture

**Conclusion**

The renewable engine is mainly used to maximize lightweight, fast, usable and easy-to-maintain resources for web services that often use http as a means of communication. We have seen the implementation of RESTful web services by developing the Ecommerce site. The Internet is the most popular business platform, many companies place their businesses online and Ecommerce is the most popular among them. Therefore, the program was built to create a scalable Ecommerce RAL API to serve any previous client.

The program will be very helpful in the process of analyzing the implementation of the remaining api. The proposed system is often used for a complex web service. Once the application is complete, the implementation therefore documents provided prior to actual implementation are both tested by this method and the results are valid. If the application does not have a specific function or function associated with any other application it will be notified to the developer.

**Acknowledgements**

We thank Mr. C.Vairavel and Mr. Praveen Dominic for reviewing this paper and for their helpful feedback on the work.

We take this opportunity to express my deep sense of gratitude towards my project mentor C.Vairavel for their valuable guidance and suggestions for the project.

**References**

To help ourselves with a kick-start for this project, there are certain resources and white papers which we have taken our references from. Those along with their links are mentioned below:

1. Sneha Jadhav1, Anjali Singh2, Apurva Upadhyay3, Smita Avhad4, Prof. Pravin Hole5 (2019) REST (Representational State Transfer) is architectural style for designing loosely coupled web services. Link is https://www.irjet.net/archives/V6/i2/IRJET-V6I2335.pdf
2. [Natalia API2Cart](https://dev.to/natalia27070051)(2021) API2Cart offers a RESTful API and provides a seamless connection to over 40 shopping platforms. Link are https://dev.to/api2cart/restful-api-and-its-usage-in-the-ecommerce-sphere-56cj
3. Ole Lensmar(2010) Swagger core tooling for java adds a number of features to the core annotations used by the Swagger runtime to generate a Swagger definition for your API. Swagger, <http://swagger.io/>
4. Phillip Webb Dave Syer Josh Long Stéphane Nicoll Rob Winch Andy Wilkinson Marcel Overdijk Christian Dupuis Sébastien Deleuze Michael Simons Vedran Pavić Jay Bryant Madhura Bhave Eddú Meléndez Scott Frederick(2012) Referenced as:<https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>
5. Chaitanya Mukund Kulkarni, Prof. M. S. Takalikar (2018) Proposed system can be used for complex web service. Refrenced as:<http://ijsrcseit.com/paper/CSEIT183535.pdf>
6. M. Fowler( 2010.), “Richardson maturity model: steps toward the glory of rest”, Referenced as: http://martinfowler.com/articles/richardsonM  
   aturityModel.html.
7. F. Haupt, D. Karastoyanova, F. Leymann, and B. Schroth,(2014) “A modeldriven approach for  
   REST compliant services”, ICWS.
8. Leonard Richardson and Mike Amundsen(2013), RESTful Web APIs”, O‟Reilly Media.
9. Research paper: Using Mean Stack to implement a RESTful Service for **an online**of things application.
10. R. Fielding and J. Reschke(2014), “Hypertext Transfer Protocol (HTTP/1.1): Semantics and  
    Content”, RFC 7231, Refrenced as: http://www.ietf.org/rfc/rfc7231.txt.
11. [S M Sohan](https://ieeexplore.ieee.org/author/37085425497); [Frank Maurer](https://ieeexplore.ieee.org/author/37275748800); [Craig Anslow](https://ieeexplore.ieee.org/author/37688797300); [Martin P. Robillard](https://ieeexplore.ieee.org/author/37283853000) (2017)  REST API developers can reduce the errors, improve success rate and satisfaction of API client developers. Link are :<https://ieeexplore.ieee.org/document/8103450>
12. Sujan Y M1, Dr. Shashidhara H R2, Dr. Rohini Nagapadma3 (2020) Discussions are a notable idea in administration configuration to depict complex communications between a customer and one or then again various administrations, Refrenced as: <https://www.irjet.net/archives/V7/i6/IRJET-V7I6207.pdf>
13. Li, L., & Chou, W. (2011). Design and Describe REST API without Violating REST: A Petri Net Based Approach. 2011 IEEE International Conference on Web Services.
14. Yang, F., & Manoharan, S. (2013). A security analysis of the OAuth protocol. 2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM).
15. Stoudenmier, S., & Olmsted, A. (2017). Efficient retrieval of information from hierarchical REST requests. 2017 12th International Conference for Internet Technology and Secured Transactions (ICITST).
16. Carlos Rodr´ıguez1, Marcos Baez1, Florian Daniel2, Fabio Casati1, Juan Carlos Trabucco3, Luigi Canali3 and Gianraaele Percannella3(2015), A Large-Scale Analysis of Compliance with Principles and Best Practices Refrenced as: <http://disi.unitn.it/~crodriguez/papers/rest-icwe2016.pdf>
17. M. Fowler. Richardson maturity model: steps toward the glory of rest(2010). Online at http: // martinfowler. com/ articles/ RichardsonMaturityModel.
18. M. Hadley. Web Application Description Language. W3c member submission, Sun Microsystems, (2009). Link as: <http://www.w3.org/Submission/wadl/>.
19. J. Musser. KPIs for APIs. The Business of APIs Conference (2014). Link are: <http://www>. slideshare.net/jmusser/kpis-for-apis.
20. D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F. Nielsen, S. Thatte, and D. Winer(2007). SOAP Version 1.2. W3c recommendation, W3C, Refrenced as: . <http://www.w3.org/TR/soap/>.
21. F. Daniel and M. Matera. Mashups: Concepts, Models and Architectures. Springer, 2014.
22. S. Khirman and P. Henriksen(2002). Relationship between quality-of-service and qualityof-experience for public internet service. In 3rd WS on Passive and Active Measurem.
23. C. Pautasso(2014) . Restful web services: principles, patterns, emerging technologies. In Web Services Foundations, pages 31–51. Springer.
24. Y.Bhole and A.Popescu(2005). Measurement and analysis of http trac. Journal of Network and Systems Management, 13(4):357–371.
25. AleksandraTarkowska(2013),  Web APIs also enhance adherence to FAIR data principles by making data Findable, Accessible, Interoperable, Reusable Link are: <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1006542>
26. Tarkowska A, Carvalho-Silva D, Cook CE, Turner E, Finn RD, Yates AD (2018) Eleven quick tips to build a usable REST API for life sciences. PLoS Comput Biol 14(12): e1006542. https://doi.org/10.1371/journal.pcbi.1006542
27. Solapurkar, P. (2016). Building secure healthcare services using OAuth 2.0 and JSON web token in IOT cloud scenario. Link are : <https://www.google.co.in/books/edition/RESTful_Web_Services/XUaErakHsoAC?hl=en&gbpv=1&dq=building+a+rest+api+for+a+website+using+java&pg=PR7&printsec=frontcover>
28. [Balaji Varanasi](https://www.google.co.in/search?hl=en&sxsrf=ALeKk03ZpUrkjokhoJXPLIDCCxTqRcLqeA:1617684201398&q=inauthor:%22Balaji+Varanasi%22&tbm=bks), [Sudha Belida](https://www.google.co.in/search?hl=en&sxsrf=ALeKk03ZpUrkjokhoJXPLIDCCxTqRcLqeA:1617684201398&q=inauthor:%22Sudha+Belida%22&tbm=bks) (2015) Spring REST is a practical guide for designing and developing RESTful APIs using the Spring Framework Refrenced as: <https://www.google.co.in/books/edition/Spring_REST/2GInCgAAQBAJ?hl=en&gbpv=1&dq=building+a+rest+api+for+a+website+using+spring+boot&pg=PP3&printsec=frontcover>
29. Sanjay Patani (2017) , Design and build Xml, Jars, JSON and Jar-rs, Refrenced as: <https://link.springer.com/content/pdf/10.1007/978-1-4842-2665-0.pdf>
30. Spring Core (2007) Repository are: <https://repo.spring.io/release/org/springframework/spring/>
31. Sophia Bush Spring Framework Notes for Professionals Spring ® Framework Notes for Professionals Link are: <https://www.academia.edu/41568874/Spring_Framework_Notes_for_Professionals_Spring_Framework_Notes_for_Professionals>
32. [K. Siva Prasad Reddy](https://www.researchgate.net/scientific-contributions/K-Siva-Prasad-Reddy-2133353429) (2020)  Spring framework is a very popular and widely used Java framework for building web and enterprise applications, Link are: https://www.researchgate.net/publication/320070654\_Introduction\_to\_Spring\_Boot
33. [Zeljko Jovanovic](https://www.researchgate.net/profile/Zeljko-Jovanovic) (2017) Java Spring Boot Rest WEB Service Integration with Java Artificial Intellgence Weka Framework Refrenced as: <https://www.researchgate.net/publication/321757987_Java_Spring_Boot_Rest_WEB_Service_Integration_with_Java_Artificial_Intellgence_Weka_Framework>
34. [S M Sohan](https://ieeexplore.ieee.org/author/37085425497); [Frank Maurer](https://ieeexplore.ieee.org/author/37275748800); [Craig Anslow](https://ieeexplore.ieee.org/author/37688797300); [Martin P. Robillard](https://ieeexplore.ieee.org/author/37283853000) (2017)  REST API developers can reduce the errors, improve success rate and satisfaction of API client developers. Link are :<https://ieeexplore.ieee.org/document/8103450>