iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** II **Month of publication:** February 2021

DOI: https://doi.org/10.22214/ijraset.2021.33013

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue II Feb 2021- Available at www.ijraset.com

Application of Restful APIs in IOT: A Review

Richa Maurya[1], Keerthi Anil Nambiar[2], Poornima Babbe[3], Jyoti Popat Kalokhe[4], Y. S. Ingle[5], N. F. Shaikh[6]
Modern Education Society's, College of Engineering, Pune, India

Abstract: This paper briefly examines the interpretation of RestFul APIs, its methods, and responses. It explains how REST (Representational State Transfer) is preferred over SOAP (Simple Object Access Protocol). Majority of the IOT devices are not straightaway related to each other. They are connected via services that provide interfaces to users. In this paper, the author addresses how the Internet of Things (IoT) can get profited from APIs. This paper also introduces a guide for Application Programming Interface (API) documentation.

Index Terms: REST, API, SOAP, JSON, IOT, HTTP, XML

I. INTRODUCTION

Currently, the data is often made available to the users on the servers via specialized RESTful APIs. APIs connect modern world applications. Majority of the applications uses APIs to establish and transmit data. API consist of request and response which differ from each other. Originally, REST was known for the connection of the Web Services but, nowadays it is becoming a common method for the development of applications. RESTFul Web Service is implemented using the Web standards that includes HTTP, XML, URI and REST principles. The paper is designed as follows:
Section II contains the definition of RESTFul API is answered in-depth.
Section III justification for the need of REST. Section IV states all the principles of REST API, and its methods are mentioned in Section V . Section VI discusses on JSON and includes the key differences between XML and JSON.
Section VII explains why REST is preferred over SOAP. Section VIII shows how API can be used to ease IOT. Section IX guides how to document APIs correctly. It is followed by Conclusion, Future Scope and References.

II. WHAT IS RESTFUL API?

A. What Does API Mean?

API stands for Application Programming Interface. It is defined as a set of routines and instructions that are used for developing software applications which allows two applications to communicate. It defines how software elements should interact. APIs unite important "things" like medical equipment, temperature sensors, and others to your ecosystem. It is crucial that the deployed API is secure, adaptable and scalable.APIs saves developer's time by taking the help or support of APIs that are already implemented by different developers which helps in reducing the number of code a particular developers need to write and it also helps to generate consistency across all the applications for the same purpose. Access to hardware and software resources can also be controlled by APIs.

B. What Does Rest Mean?

REST is the acronym for Representational State Transfer. It defines the set of protocols used to develop Services for Web. It is a client-server communication, in which the user can access the data as and when required from the server. It also defines the rules for using web standards that includes URLs, HTTP and REST principles.The architectural approach that REST has limits the usage of bandwidth and hence makes it more suitable for internet. It's the most commonly used web services. It allows multiple clients like browser applications to interact with a server using the REST APIs.In REST framework style data and functionalities are observed as resources which are retrieved using URI's which stand for Uniform Resources Identifier and typically known as links on websites.
REST framework is basically client-server framework, in which client and server exchanges representation of resources having interface and protocols. This makes REST apps simpler, weightless and increases its performance. REST services are the applications which are modeled on top of REST framework. REST exposes data through web URI's and uses HTTP functions to create, update, retrieve and delete data. The key elements of Restful APIs are follows:

©IJRASET: All Rights are Reserved

1) *Resources:* The basic idea in any REST API is the resource which is an object having types of associated data, a relationship to other resources, and a set of functions that works on it. A resource is said to be any kind of information or data such as a text document, a picture, a temporal service or a set of another resource, etc.

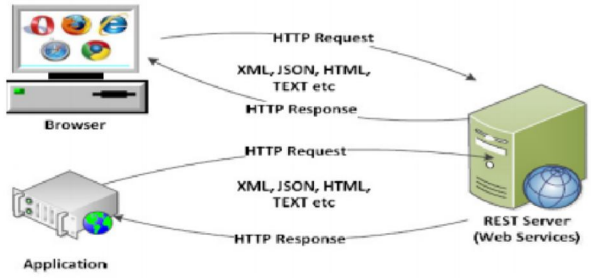


Fig. 1. REST web service architecture

2) *Request Verbs:* Request Verbs are generally HTTP methods that describe what needs to be done with the data available on the server. Browser uses GET verb for getting data from server. Like that other request verbs such as PUT, POST, DELETE are used by browser for performing operation on server.
3) *Request Headers:* Headers are different additional information that is being send along with data. It also contains the different formats in which data need to be fetched. They contain the data for the request and the response body. It also contains the information for request authorization, response caching and response cookies.
4) *Request Body:* Data is posted with a request. Data is sent to server when POST request is made to the web service. As POST request adds data to server which is not added previously the request body contains the details of where the new data needs to be added in web server.
5) *Response Body:* Response body consist of the details of response that we get when from the request that is being sent. The response can be get in XML format or JSON format.
6) *Response Status Code:* Response status code are the codes that indicates status of response from web server. There are different codes which indicates different status of response. For example code 201 indicates that new resources added to server without any error.

Considering the drastic change over the technologies in the last few years a huge amount of data is being generated. Users can access large amount of data from the server by using RESTFul APIs.

III. NEED FOR RESTFUL API

Data is received from the server as a response when the client requests some information from the server. This entire process executes via API. REST APIs can supervise different types of calls, it can return multiple data formats and can even adjust structurally having the right approach of hypermedia as information is not tied to data or functions. Coding of REST service and its execution is easier as compared to SOAP. The REST web service decrease the time of implementing and gives a fast solution. Rest API are highly flexible and easy to build (no tool-kits are required).

IV. PRINCIPLES OF REST API

A. Stateless

The client transmits request to the server containing the required information to access the particular data stored in the server. The request may include the headers, parameters, authorization details, query-string parameters, body, URL.
The URL is useful in distinctly identifying resources and its body contains the state of the requesting resource. When the server processes a request then a response is transmitted to a client via status, body, or headers.

B. Client Server

It makes the interface steady and isolates the clients from servers. Clients are not concerned with information storage, so the flexibility of client side code is enhanced. UI does not rely on servers, so they can be made simpler and more scalable. It is possible to replace and code Servers and clients independently.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue II Feb 2021- Available at www.ijraset.com

C. Cacheable

Responses are labelled as cacheable or non-cacheable from the server either implicitly or explicitly. If its responses are marked as cacheable, then the client cache can be reused later.

D. Uniform Interface

Uniformity in the application can be obtained by using REST. Following are the interface constraints:

1) Identification of the resource
2) Manipulation of the resource using representations
3) Messages that are Self-descriptive

E. Layered System

With the help of a layered system, appliances become stable by limiting component behavior, which assists in increasing the app's security as the components in each layer cannot interact to any other layer if it is not the immediate layer.

F. Code On Demand

It is an elective constraint and is rarely used. It is used to simplify client's work by creating a smart app that does not depend on its own code structure.

V. METHODS OF REST APIS

API calls can be categorized into various types that supported their functionalities. Web applications are often created using REST APIs having all possible CRUD (create, retrieve, update, delete) methods. REST guidelines advise using an appropriate HTTP method on a selected kind of call made to the server.

A. HTTP Get

Get request are wont to retrieve resource information only and to not modify it that's it don't change the state of the resource which is why it's called to be safe method also get also GET API should be idempotent ,which means that it should give the identical result for multiple identical request.

B. HTTP Post

It is used for creating the new sub ordinate resources .POST is neither safe not idempotent. The request body consists if the data send to the server. If invoked two identical post request then it will generate two different response.

C. HTTP Put

It is primarily used to update existing resources .If the resource does not exists then the API decides whether to create a new resource or not .If a new resource has been created by PUT API then the server should be informed about it.Calling the same PUT request multiple times will generate the same results.

D. HTTP Delete

It is used to delete resources. Delete operations are idempotent which means that if you delete the resource it is removed from collection of resources and calling delete API multiple times on the same resource will not change the result but will you the error[NOT FOUND] as it is already removed.

E. HTTP Patch

It is used to modify capabilities. It only contain the changes to the resource, not the complete resource .It is neither safe nor idempotent.

F. HTTP Head

The HEAD method is similar to GET, except without the response body. If GET /users return a list of users, then HEAD /users will make the same request but the list of users is not returned back. With help of HEAD requests, we can check what a GET request will return before actually making a GET request.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue II Feb 2021- Available at www.ijraset.com

G. HTTP Options

The OPTIONS method changes the different communication options and can be used to both open up more channels of communication and lock them down.

H. HTTP Trace

This function is used to execute a message loop-back analysis along the path to the destination resource.

VI. APPLICATIONS OF API

A. Travel Booking

The travel booking services is maintained through the use of APIs. This enables quick and autonomous exchange of both the data and request of the clients.

B. Twitter API

Twitter have twitter bots. These bots can tweet, retweet and can even message an account directly. .

C. Educational Institutions

Educational Institutions such schools, coaching classes use RESTFul APIs to track and manage the students information.

VII. JSON RESPONSES

JSON (JavaScript Object Notation) is a standard text format which is easy for humans as well as for machines. Human have the capability to read and write while the machines will analyse and generate. It is a bunch of key and value pair.

REST APIs being most popular it is used by website to communicate to the server. Therefore, they should be designed properly so that there is no problem at client's end. JSON is used by the REST APIs as the request payload and it will send responses to the JSON.

Transferring data is mostly preferred through JSON and it is being used by almost every technology. Technologies to decode JSON are available at the server side for easy work. JSON gets automatically adapted in the form of an array or objects making the interaction with APIs easier. It makes API more feasible to use it at out work.

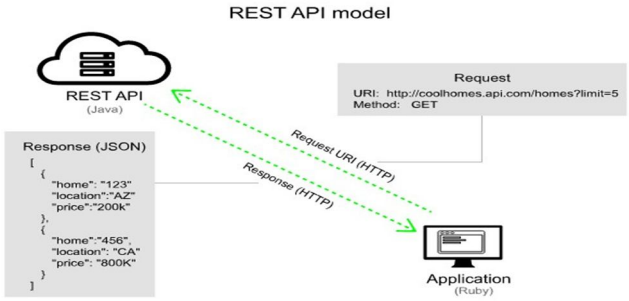


Fig. 2. Model of REST API

Extensible Markup Language (XML) used is mainly used for exchange of data and developments in these fields have increased the number of options. XML was widespread method for API integration but,JSON having the maximum advantages it is mostly attracting large number of users. JSON is lightweight compared to XML as it has got a compact structure consisting of key value pair and it is considered more legible. While working with systems that are complex , JSON can be used for advancements in RESTful APIs.

JSON objects can be readily accessed while XML data needs to be parsed.

XML representation is by default heavy because of the use of closing and opening tags that adds too much useless weight to the transmission data unit and hence it becomes difficult to parse.

JSON is more compact as compared to XML and can be easily loaded in JavaScript.

JSON's data structure is a map and it is easier to interpret and is predictable whereas XML's data structure is a tree. RESTful APIs requires simple, reliable and quick data transfers and, JSON satisfies the requirement for these attributes while it is difficult for XML to meet its requirements.

©IJRASET: All Rights are Reserved

VIII. WHY IS REST PREFERRED OVER SOAP

Web services are categorized into two categories one of them is REST and other one is SOAP(Simple Object Access Protocol). These two categories are formed based on their architectural style .SOAP was standard approach for web service communication but in recent years it is dominated by REST approach.

SOAP messages comprises of a body and a header, using a specific XML schema and namespace. It uses a standard protocol and it requires message in the XML format for responses and requests. XML messages are generally to heavy and lengthy. So this format proves to be problematic in the applications where there is a low bandwidth and because of the increased overhead it slows down the processing.

REST APIs can use any of message format including XML, JSON, RSS ,Atom, CSV, HTML. Most of the REST APIs use JSON as the default message format due to its added advantage over others. Therefore in REST the work can be done in variety of data formats but in case of SOAP it can only be done with XML.
REST operates through consistent interface to access named resources It is mostly used by developer for unmasking public API.
SOAP on other hand unmask the components of application logic as service rather than data. In simple way, REST access data while SOAP perform operation through standardized set of messaging pattern

A. Benefits of Rest Over Soap

1) The superior performance is provided by REST by catching data for non volatile type of data, while SOAP gives lower performance compared to REST .
2) Comparatively REST is faster and by using less bandwidth enables it to integrate with preexisting website without changing infrastructure.
3) REST based mobile services uses less resources when it is compared to SOAP based mobile services and also achieves higher performance.
4) Some work is performed to analyze performance of REST and SOAP , Comparisons showed that REST takes lower time to respond by comparing tested results.
5) The experimental results showed that JSON perform better as compared to XML which is used by SOAP in terms of being parsed and being serialized.

IX. IOT AND APIS

A. The concept of IoT or Internet of Things helps us to understand the association of hardware devices, and the connection to network. Actually, when we look into a real life case, many electronic devices are not straightly connected with one another- but it is connected to some different assistance that helps incorporating with other devices and other users. IoT devices are becoming very popular in every walk of our life even in large scale. We use IoT in our daily appliances at home, workplace and also in large institutions for establishment of smart city, traffic management and utility billings, etc.
B. IoT helps us to build digital applications that can be integrated with our daily appliances using REST.
C. REST API are directly connected with the IoT devices as they allow the safe and authorized exposure of devices to clients, and various appliances in the IT framework. To get benefitted completely by IoT, REST API is required for every IoT device as REST ensures the smooth data transfer over network regulations and controls the authorized and secured transactions.
D. APIs use the HTTP protocols for performing various operations such as create,read and modify and these could be linked to any type of IOT devices.
E. A RESTFul API portrays a lot of classes of IoT applications very well. Consider an example, an IOT appliance which does not have an in-built UI can act as a special purpose posting-client, by accessing the API it can push the data acquired from the sensor to a central server using a POST (Create) method. The data could be avail;able as a response by using GET API calls. With the help of get and post method the data can be posted and easily be retained using the interface
F. REST API provides the most simple and efficient way for appliances and network to connect in a standardized manner. The information/data received through REST API calls requires to be analysed and managed in different databases.
G. REST APIs makes it possible to get the information available and get a generalized method to read,create and modify the data stored. It is possible for the developers to create a REST model for easy access of data. Consider a light bulb, if its current state id ”off”, then a request can be made to turn it on. Or, in the case of a heater, its current temperature can be received and can be changed.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue II Feb 2021- Available at www.ijraset.com

X. GUIDE FOR API DOCUMENTATION

A. Why API Documentation is Necessary?

API keeps evolving day by day,so the process of maintaining their documentation with examples is time consuming and expensive. If the API document does not contain proper date formats, authorization details, header details, request body then it is difficult for API client developers .If proper guidance documentation is followed by the developers then they can improve their accomplishment rate and user satisfaction rate.So a proper documentation includes the problem faced by most of the API client developer, due to lack of proper knowledge of usage which is only given by considering various examples and different types of APIs. Also the documentation includes some evidences so that developers can reduce that mistake and can improve API usability.

B. A Proper API Documentation Must Include The Following Points

1) *Representative API:* Select an API that is used by most of the API client developers for the study. Selection of mature API for study plays a vital role as it reduces the probability of hurdles in the implementation. Choosing a recognizable arena will help participants to correlate to the API features without requiring any direct training.
2) *Open Source:* Documentation should be an open source to be able to conclude the result of usage examples, for this there is need to select an open-source API where we can add examples.
3) *Time Bound:* Large amount of time constraint should be applied to each developer to calculate probability of success within required time bound. So that the study needs to be setup such that participants are able to focus on performing the tasks minimizing any overhead
4) *Participant Selection:* For the recruitment of the members to study and to do the study within an specific amount of time and in a sensible setup, Developers with former experience on REST API needs to be selected.

C. Metrics Measures Used From API Consumers' Point of View

1) *Response Time:* 1) Response Time: Time that is measured in milliseconds is the time that web service needs to react to data. The time starts by sending a request to a service and stops when a response is delivered.
2) *Response Length:* It is the length of the responses as the number of characters that match the complexity of the service. It becomes difficult for the metric to differentiate between a big response having numerous key-value pairs and with a single large element. We can also assume more complicated services should return higher responses.
3) *Number Of Keys In Response:* The number of keys in the response that a service provides to the client. This metric distinguishes between a long response with several keys and that with a single long element.

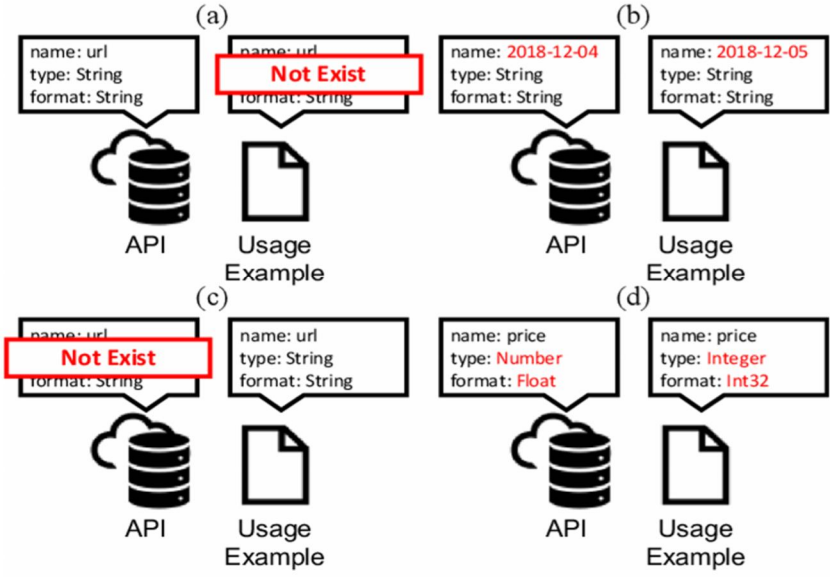


Fig. 3. Types of mismatches: (a) Undocumented Keys Pattern, (b) Dynamic Keys Pattern, (c) Un-returned Keys Pattern, and (d) Type Mismatched Pattern

©IJRASET: All Rights are Reserved

D. Some Recommendation For API Documentation

1) *Recommendation 1:* If Patterns of data types such as Integer, String, and array are added in the REST API documentation for the API field it will lead to the relief of the API client developer about the knowledge required.
2) *Recommendation 2:* Sample valid data formats need to be included for the API elements in the documentation by the REST API developers.
3) *Recommendation 3:* In extension to the request method and the body should also the REST API developers should also introduce patterns of HTTP headers if API requests demand to use of HTTP Request headers.

XI. LIMITATIONS OF RESTFUL APIs

1) *It is Stateless:* The web applications mostly require stateful mechanisms. The responsibility of preserving the state lies on the client, which makes the client tough and difficult to maintain its application
2) *Lack of Security:* REST inflicts comparatively a poor security than SOAP. REST is suitable for public URLs, but it is not good for private data passage between client and server because there are chances for data leakage.
3) As HTTP is request/response protocol, the protocol does not inherently support peer to peer notifications which are required.
4) In case of robust security, data privacy and integrity rest API get lowered as compared to SOAP.
5) In case of event based parsing SOAP services adds scalability by using HTTP processing along with XML parsing while REST uses HTTP processing method alone which turns drawback.

XII. CONCLUSION

In this paper, the author explains in depth about RESTFul APIs and the importance of it in data communication over client and server. This paper also highlights why REST is preferred over SOAP along with information about SOAP. The author discusses why REST request and response is mostly preferred in JSON format and why one should not use XML. The paper also described how IOT can be benefited by the use of APIs and how they work with each other. This paper explains the importance of documentation of APIs, mentions the important points in the documentation and provides recommendations for API documentation. This paper also lists the limitations of RESTFul APIs.

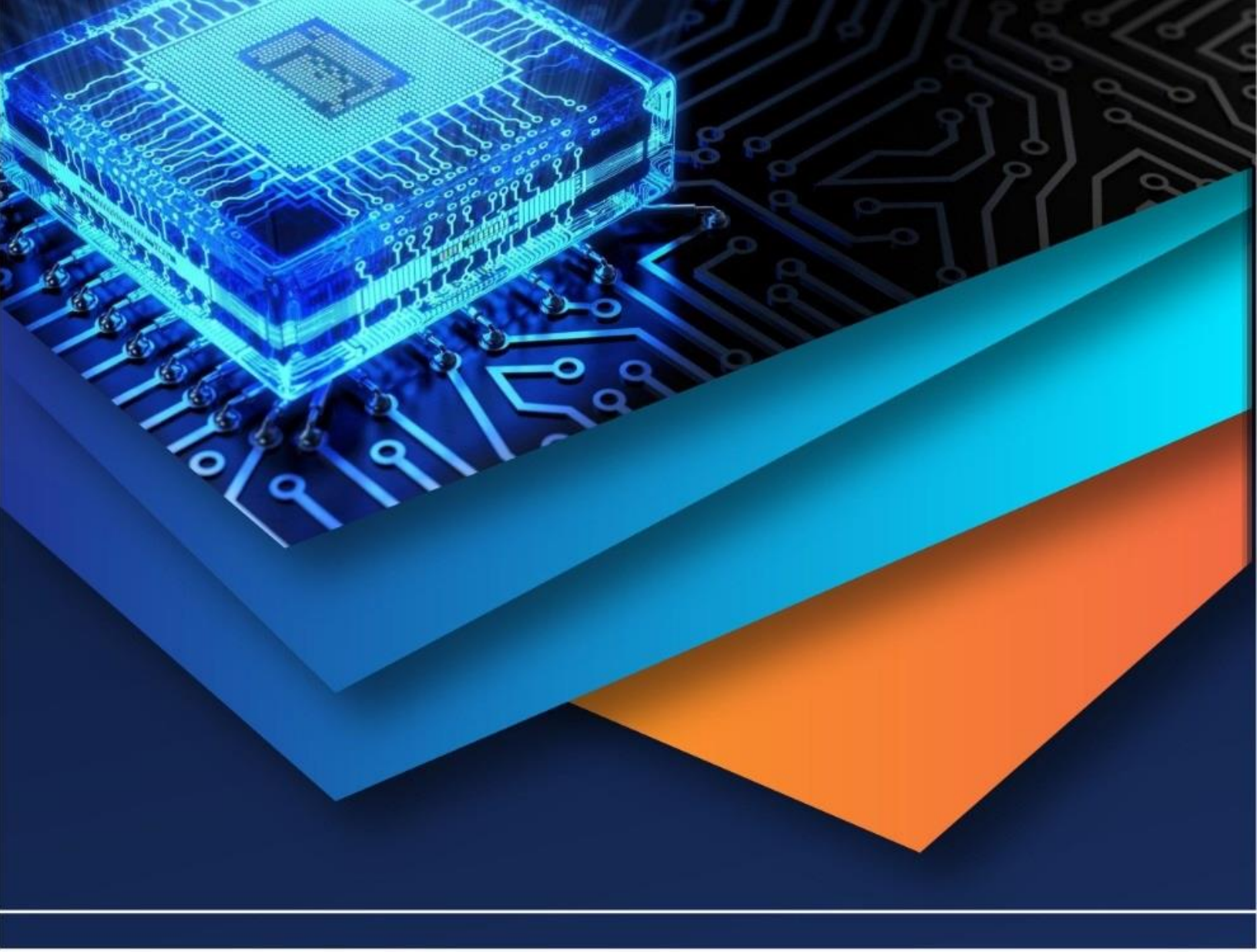
XIII. FUTURE SCOPE

REST API design had performed a necessary role as it affects both clients as well as user agents and origin servers. REST is best for public URLs but has some problems when it comes to sensitive/confidential communication between clients and servers as it is lagging behind from a security perspective and may have the probability of data being leaked. So we can improve the security feature of REST to make it appropriate for private data transfer. In the future, we can work on making the REST capable to work on different protocols other than HTTP.

REFERENCES

[1] M Sohan, Craig Anslow, Martin, Frank Maurer, P.Robillard."A Study of the Effectiveness of Usage Examples in REST API Documentation". Symposium on Visual Languages and Human-Centric Computing (VL/HCC) 2017 IEEE.
[2] Edna Dias Canedo, Rafael Timoteo´ de Sousa Junior, Hiro Gabriel Cerqueira Ferreira. "IOT Architecture to Enable Intercommunication Through REST API and UPnP Using IP, ZigBee, and Arduino." IEEE 9th International Conference on Wireless andMobile Computing, Networking and Communications (WiMob) 2013
[3] Li Li, Wu Chou "Design and Describe REST API without Violating REST: A Petri Net Based Approach", IEEE International Conference on Web Services 2011.
[4] Ei Ei Thu, Than Nwe Aung "Developing Mobile Application Framework by Using RESTFul Web Service with JSON Parser", International Conference on Genetic and Evolutionary Computing,GEC2015,Advances in Intelligent Systems and Computing, vol 388. Springer, Cham.
[5] Poulter, A. J., Johnston, S. J., and Cox, S. J. "Using the MEAN stack to implement a RESTful service for an Internet of Things application". IEEE 2nd World Forum on Internet of Things (2015).
[6] Beniamino Di Martino,Antonio Esposito,Salvatore Augusto Maisto,Stefania Nacchia,"A semantic IOT framework to support RESTful devices' API interoperability",IEEE 14th International Conference on Networking, Sensing and Control (ICNSC) 2017.
[7] Florian Haupt, Frank Leymann, Anton Scherer, Karolina Vukojevic-Haupt."A Framework for the Structural Analysis of REST APIs".2017 IEEE International Conference on Software Architecture.
[8] Li, L., Chou, W., Zhou, W., and Luo, M." Design Patterns and Extensibility of REST API for Networking Applications". ,IEEE Transactions on Network and Service Management, 12 January 2016.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)