INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 5.928

Travel Advisor using REST API and Google API

Tushar .V Kafare

Ph. D Scholar, Dept. of Electronics and Communication Engineering, Nirwan University, Jaipur, India

ABSTRACT: "Travel Advisor" is an application that helps to search travel details, restaurant details, hotels details. This application gives real-time location details or shows the nearby best restaurant and hotels. There are many other applications in the market but they are not so user-responsive or user friendly our application has a google map it helps tourists to show the location of restaurants, hotels, and interactive places. To save time for the tourist and provide quick and fast search results, provides information about the nearby best restaurant. To provide information about the best hotels to stay Our project gives 24x7 updates to users. Abstract—Businesses are increasingly deploying their services on the web, in theform of web applications, SOAP services, message-based services, and, more recently, REST services. Although the movement towards REST is widely recognized, there is not much concrete information regarding the technical features being used in the field, such as typical data formats, how HTTP verbs are being used, or typical URI structures, just to name a few. In this paper, we go through the Alexa.com top 4000 most popular sites to identify precisely 500 websites claiming to provide a REST web service API. We analyze these 500 APIs for key technical features, degree of compliance with REST architectural principles (e.g., resource addressability), and adherence to best practices (e.g., API versioning). We observed several trends (e.g., widespread JSON support, software-generated documentation), but, at the same time, high diversity in services, including differences in adherence to best practices, with only0.8% of services strictly complying with all REST principles. Our results can help practitioners evolve guidelines and standards for designing higher quality services and also understand deficiencies in currently deployed services. Researchers may also benefit from the identification of key research areas, contributing to the deployment of more reliable services.

KEYWORDS:-API, Simple Object Access Protocol (SOAP), Remote Procedure Call (RPC), Google API.

I. INTRODUCTION

 Application Programming Interfaces (APIs) have been growing rapidly. Several studies point out that developers have moved from Simple Object Access Protocol (SOAP) or Remote Procedure Call (RPC) to deploying Representational State Transfer (REST) web services, as the means for consumers to use their services. This is corroborated by major websites like Google, Facebook, and Twitter, which are now deploying REST services to provide easy access to their valuable data resources while promoting their businesses. An API is a set of programming codes that enables data transmission between one software product and another. It also contains the terms of this data exchange [1].

TYPES OF API:-

Private APIs:-
These application software interfaces are designed for improving solutions and services within an organization. In-house developers or contractors may use these APIs to integrate a company's IT systems or applications, build new systems, or customer-facing apps leveraging existing systems. Even if apps are publicly available, the interface itself remains available only for those working directly with the API publisher. The private strategy allows a company to

fully control API usage.

Public APIs:-
Also known as developer-facing or external, these APIs are available for any third-party developers. A public API program allows for increasing brand awareness and receiving an additional source of income when properly executed. There are two types of public APIs – open (free of charge) and commercial ones. The open definition suggests that all features of such an API are public and can be used without restrictive terms and conditions. Commercial API users pay subscription fees or use APIs on a pay-as-you-go basis. A popular approach among publishers is to offer free trials, so users can evaluate APIs before purchasing subscriptions. In our project, We are using an application programming interface (API). Basically, an application programming interface (API) is a connection between a computer or between computer programs. It is a type of software interface, that offers a service to other pieces of software. In other words, an API is a messenger that delivers your request to the provider that you're requesting it from and then delivers the response back to you, Imagine a waiter in a restaurant. You, the customer, are sitting at the table with a menu of choices to order from, and the kitchen is the provider who will fulfill your order. You need a link to communicate your order to the kitchen and then to deliver your food back to your table. It can't be the chef because she's cooking in the kitchen [2].

II. PROBLEM DEFINITION

To book hotels, the customer has to visit different hotels' websites and the customer has traveled to a city to find out his affordable hotels to stay in.To visit an attraction spot customer has no idea of the condition in that respective city.For customers finding travel details at an instance, time is difficult, like hotels to stay in, the best foods in the city, the best attraction in the city [3].

III. METHODOLOGY

The empirical analysis carried out in this paper was performed from February to July 2017. For simplicity, from this point onwards we use the term REST (accompanied by web service API, service API, or web API) to designate a service that is claimed to be RESTful by some provider. We use the term truly RESTful if it adheres to all mandatory REST principles [4]-[5].
The study comprised the following steps, described in further detail in the next paragraphs: 1) Identification of best practices for REST API development 2) Find a suitable URI source to use as a basis for the analysis; 3) Filter duplicate sites (e.g., regional site variants);4) Identify the site category (e.g., games, shopping); 5) Examine the site for available service APIs: a. Identify service APIs announced as REST by the site and select a subset for analysis; b. Identify other types of service offered by the Site.6) Analyze technical features (identified in previous works [and gathered during our observations) of the REST service APIs, going through a. Compliance with REST principles [5]; b. Service design decisions; c. Adherence to best practices [6]. 7) Verification of the results.
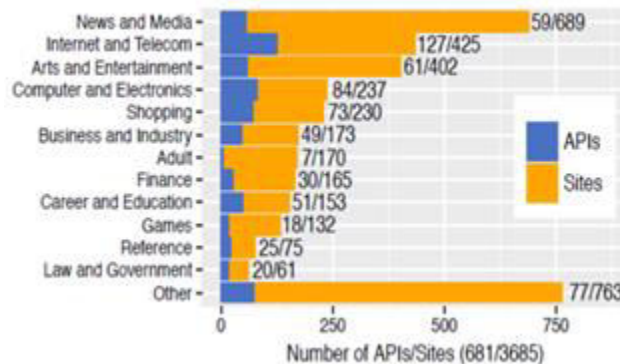
Fig.1 Number of Slides and APIs Category

REST Architecture – This dimension mostly covers core REST features of the API [7]: i) a number of operations; ii) design schema used in the URIs (i.e., does the URI only contain resource information or does it also contain information about the operation); iii) the technique of the output format selection; iv) the scoping information (i.e., how the server is informed about on which data it should operate, for instance using data specified in the URI path or in an HTTP header field); v) API versioning support (i.e., is it possible and how to select among different API versions, how many major versions were deployed); (e.g., sending a success response using a 2xx status code) I/O and Software Development Support –This covers the input and output formats supported by the services[8]. It also addresses the support of Software Development Kits (SDKs) for the API consumer.• Security Mechanisms – This refers to the presence of user authentication mechanisms (is user authentication required and if yes, which mechanism is supported) and if communication channel encryption is mandatory, optional, or simply not supported. Usage Policies – This dimension covers the type of registration that is necessary for using the API, whether payment plans for using the API are available if call limits are enforced, and in this latter case, how many daily API calls are allowed. Documentation and Application Use – This dimension covers general documentation aspects, namely if the APIs provide: i) documentation updated in the current calendar year (at the time of data collection), i.e., between January to July 2017; ii) generated API documentation (i.e., produced by a tool, such as the Swagger UI [9]); iii) interactive developer consoles (i.e., for testing requests and observing responses); iv) explanations of error messages; v) example requests and responses. In addition, we also examine the reported number of applications that use the analyzed APIs[10]-[11].

IV. CONCLUSION

Travel Advisor (REST API and Google API) meets the objectives of the requesting data by remote API and data which it has been developed. The application has reached a steady-state where all bugs have been eliminated. The webpage is operated at a high level of efficiency. The Travel Advisor websites render the travel details. It was intended to provide information and to visit attraction spots as requirement specifications. The web application can respond efficiently and showcase real-time weather predictions at the current location on a map. Our Project "Travel Advisor (API)" is mainly intended and further modified into more like we can fetch Real-time Flight details. We can also book a flight to travel or search for trains. Also, we are going to link a DBMS to our project. Where we can store past traveled data. So users can get information about their past visits to a different location.

REFERENCES

[1] B. Mulloy, "RESTful API Design: complement with an SDK |Apigee."[Online]. Available:https://apigee.com/about/blog/developer/restful-api-designcomplement-sdk. [Accessed: 18-Jan-2018].

[2] M. B. Juric, I. Rozman, B. Brumen, M. Colnaric, and M. Hericko, "Comparison of performance of Web services, WS-Security, RMI, and RMI–SSL," J. Syst. Softw., vol. 79, no. 5, pp. 689–700, May 2006.

[3] D. Akhawe, A. Barth, P. E. Lam, J. Mitchell, and D. Song, "Towards a Formal Foundation of Web Security," in 2010 23rd IEEE Computer Security Foundations Symposium, 2010, pp. 290–304.

[4] R. Rodríguez, R. Espinosa, D. Bianchini, I. Garrigós, J.-N. Mazón, and J. J. Zubcoff, "Extracting Models from Web API Documentation,"in Proc. of the 12th Int. Conference on Current Trends in Web Engineering, Berlin, Heidelberg, 2012, pp. 134–145.

[5] M. Vaziri, L. Mandel, A. Shinnar, J. Siméon, and M. Hirzel, "Generating Chat Bots from Web API Specifications," in Proc. ACM SIGPLAN Int. Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software, 2017, pp. 44–57.

[6] Rosado, D. G., Fern´andez-Medina, E., L´opez, J., and Piattini, M. (2010b). Developing a Secure Mobile Grid System through a UML Extension. J. UCS, 16(17):2333–2352.

[7] Ross, S. (1983). Stochastic Processes. Wiley series in mathematical statistics. Probability and mathematical statistics. Wiley.

[8] Rugina, A.-E., Kanoun, K., and Kaˆaniche, M. (2007). A System Dependability Modeling Framework Using AADL and GSPNs. In Lemos, R., Gacek, C., and Romanovsky, A., editors, Architecting Dependable Systems IV, volume 4615 of Lecture Notes in Computer Science, pages 14–38. Springer Berlin Heidelberg.

[9] Sanders, W. H. and Meyer, J. F. (1991). A Unified Approach for Specifying Measures of Performance, Dependability, and Performability. Dependable Computing and Fault-Tolerant Systems: Dependable Computing for Critical Applications, 4:215–237.

[10] Sandhu, R. S., Coyne, E. J., Feinstein, H. L., and Youman, C. E. (1996). Role-Based Access Control Models. Computer, 29:38–47.

[11] Scarfone, K. and Mell, P. (2007). Guide to Intrusion Detection and Prevention Systems (IDPS). Technical report, The National Institute of Standards and Technology (NIST). Special Publication 800-94.

INTERNATIONAL JOURNAL
OF MULTIDISCIPLINARY RESEARCH
IN SCIENCE, ENGINEERING AND TECHNOLOGY

9710 583 466   9710 583 466   ijmrset@gmail.com

www.ijmrset.com