

Application to Classify the Status of a Given Environment Service

Rahul Subramaniam and Rashmi R

Department of Information Science and Engineering, RV College of Engineering, Bangalore, Karnataka, India
Assistant Professor, Department of Information Science and Engineering, RV College of Engineering, Bangalore, Karnataka, India

Abstract - With increasing complexity of software systems, the manifold debugging techniques have increased anomalies detection, assess impact, and schedule software patches or full updates to a system. In most Multi National Conglomerates, the servers deal with a phenomenal amount of data. Every individual working on an organization product works in the capacity of a certain environment. Most systems contain Developer and Testing Environments wherein each environment has a certain set of functionalities. A status of an environment gives us information about the Operationability i.e how many of the services offered by the environment are operational at the given juncture of time. It is of a great deal of importance that a given environment user knows the services that are being provided.

Hence, the proposed project aims at resolving the issue at hand by taking care to resolve the environment status issue. A given environment gateway/service may either be in a 'good' state, 'undefined' or an 'error' state. With appropriate access to the servers, the project deals with analysing whether the given service of an environment contains an error or not. The details are displayed on a centralised web application that is available to the organization. The project provides this is done by the use of relevant scripting languages such as Windows Powershell to obtain the results, and feeding them to an appropriate Javascript runtime. The Javascript runtime, after obtaining the ideal of the environments and its services, analyses whether the level to which the service is functioning, and determines whether it is good or not. If the number of services that possess an error state in a given environment exceed the maximum threshold, then with the aid of Simple Mail Transfer protocol, a notification will be sent to alert the team that the given environment requires dedicated attention.

Key Words: Simple Mail Transfer Protocol, Error, Javascript Runtime.

1. INTRODUCTION

The objective of proposed project is to help simplify the bug fixation and the monitoring process. The proposed system helps in solving the problem of Efficiency by alerting the entire community in case an error is encountered at any given time. The status tracking is done automatically thus, reducing human effort in this process. Since, the process is done automatically the human effort required is greatly mitigated. The project also attempts to provide solution for the situation when few errors are encountered that have a

predefined course of action. This is done by using predefined subroutines available as a part of error procedure.

The scope of the project "Dynamic Environment status tracking web based application to support system bug fixation" mainly lies in such a way as to keep a check on the status of the environment services, gateways and the overall environment using Scripting, Server access, Javascript runtime.

The automated system alerts the team with respect to which environment is under attack, thus aiding in minimising the overall scrutiny.

2. LITERATURE SURVEY

Aaron Sterling in "NodeJS and Angular tools for JSONLD" discusses about the three open-source, ISC-licensed tools that improve the experience of web programmers who use jsonld.js, the official JSON-LD JavaScript library. Two of them are for NodeJS, with which one can manage servers using JavaScript. The NodeJS tools are: jldc, a NodeJS command line tool; and node-json ld, a module that can be plugged into other programs. The two NodeJS tools are written in Typescript 3 and compiled to JavaScript ES2015. The final tool is written in and for Angular, which is used in frontend development. The Angular tool is ngx-jsonld-provider. All three tools extend jsonld.js with better error trapping, more verbose help messages, and greater type safety. All three tools are registered with the Node Package Manager (npm). At the time of this writing, web developers have downloaded them from npm more than 500 times combined. [1].

Anjali Goyal and Neetu Sardana in "Characterization Study of Developers in Non-Reproducible bugs" discusses about how bug handling is a cumbersome and time consuming task. They gauged developer similarity (in terms of developer marking bug report as NR and Fix), tossing trends, presence of IsBack and expertise level of developers who marked NRF bug reports as NR and Fix. They studied the change history of 24, 995 NR bug reports of Mozilla Firefox project. Their results show that 87.34% NRF bug reports are fixed by a developer who had not marked the bug report initially as NR. Also, they found that average tossing path length in NRF bug reports is three times higher than tossing path length in NR bug reports. This higher rate of bug tossing results in higher fixation probability of NR bug reports. It has also been

observed that developers who fix NR bugs possess higher expertise than developers who marks bug reports as NR. [2].

R. Durga, S Nagadarshini in "Newsone-An aggregation System for news using web scraping method" discusses Newsone, a dedicated platform that aggregates all the latest news updates from multiple national and international resources and summarizes them to present in a short and crisp words. This online platform provides a service oriented interaction among the users from across web. The main motto of this application is to access the news fast. It will bring news directly without wasting any time for searching and news loading. NewsOne uses crawling method to extract the content from websites. The basic concept deals with news feeds, the admin or sub admin will add all the news URL into a database. The crawler fetches the content from the RSS feeds of the stored URL. A bot is employed that will dynamically extract the contents at certain intervals. This paper describes a model to perform categorization which extracts useful information for classifying a document into category by referring to URL. Users will get a flexible experience on this application. It allows the readers to read the news based on interest. This can be possible by enabling them to choose the categories of news such as Just-In, Technology, Health, Science, Sports, Business and Economics and Entertainment. It provides the reader to read the news for free of cost and in the fastest way possible[3].

Tomas Cerny, Jiri Sebek and T.Subha in "On metadata extension to derive data presentations with angular2" discusses the viewpoint of how the World Wide Web and web application development evolves. The Robust Server application (backend) handles DBMS, as well as a broad range of tasks, and the Client application (frontend) provides data for presentation to users and mediating the connection with the server. Contemporary frameworks, such as Angular 2 or ReactJS, extend the frontend code volumes. Such development approach has on large with information restatement and this exacerbates maintenance cost.[4].

Erich Stark, Pavol Bistak in "Virtual Laboratory based on Node.js Technology" demonstrates remote control of test experiment in the virtual laboratory. This is a common problem, but other way can always be used to solve it. The paper compares several existing virtual laboratories and their possible issues at present. To develop such a new solution JavaScript technology was used on both client and server side using Node.js runtime[5].

One of the more interesting developments recently gaining popularity in the server-side JavaScript space is Node.js. It's a framework for developing high-performance, concurrent programs that don't rely on the mainstream multithreading approach but use asynchronous I/O with an event-driven programming model. This is the primary topic of discussion in 'Node.js: Using Javascript to build high performance network programs' by Stefan tilkov, Steve Vinoski [6].

3. METHODOLOGY

The Methodology is divided into eight steps as follows:

- Raising permission requests to an organizations permission platform in order to be able to view the various gateways and services active in each environment of the product[7].
- Appropriate analysis and study of various services in good or bad state among the active pertaining environments.
- Ensure the servers in the backend are up and running, so that a connection from an individual client occurs smoothly.
- Connect to the remote database on Community servers using a suitable Javascript runtime preinstalled on the local system[8].
- Obtain the ideal state value of each service present in the system by performing web scraping and parsing the html content, with the aid of available Application Program Interfaces.
- Club the scraped data and data from backend to obtain a suitable Javascript Object Notation response, that is passed to the front end modules.
- Invoke chart.js, an available chart design framework to graphically display the detail of services and gateways in environments[9].
- Alert the community via Simple mail transfer protocol when the a greater percentage of the services in a given environment are failing.

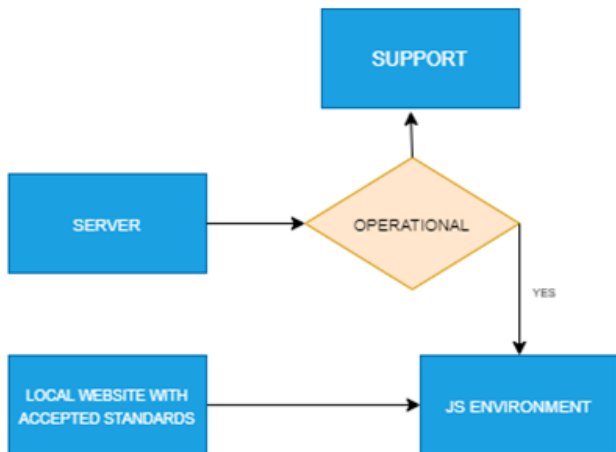


Fig 1. Block Diagram of obtaining environment information

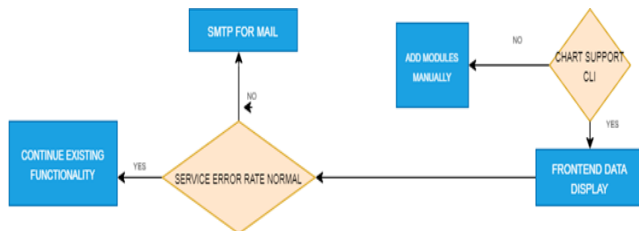


Fig 2. Block Diagram of Mail Sending process

4. RESULTS AND DISCUSSIONS

The objective of proposed project is to determine status of services. The proposed system helps in solving the problem of Efficiency by alerting the community incase an error is encountered at any given time. The status tracking is done automatically thus, reducing human effort in this process. Since, the process is done automatically the human effort required is greatly mitigated. The project also attempts to provide solution for the situation when few errors are encountered that have a predefined course of action. This is done by using predefined subroutines available as a part of a error procedure. The objectives and their means of fulfillment listed are as follows[12-14]:

- Getting the Angular8 framework work. This step was successfully completed within 1 week of the date of project initiation. The NodeJS command prompt was installed that allows us to create angular projects using the ‘ng’ command.
- To allow multiple programs written in languages such as typescript, python etc. to run. An IDE, such as Visual Studio Code, that supports multiple languages was installed.
- To obtain backend access. Access was obtained via Jira and Powershell script. PermSoft was required to obtain

the backend permission, while Powershell was required to fetch the data on to your system

- To scrape the guidelines using axios and cheerio to obtain the community guidelines for acceptable services. Objective was achieved using NodeJS express.
- Services to arranged or good, error or undefined state. Appropriate data was analysed using acceptable guidelines and services were arranged or good, error or undefined state. Typescript was used to achieve this given objective.
- To achieve the environment status optimization. Once the services were classified, if the number of services in a bad state (‘error’) exceeded a particular number, an alert via outlook is sent to the systems team. Python with flask was used to achieve this objective.
- To perform the Service Health monitoring a part of environment status tracking. Service health monitoring involved classifying the channel testcases into success or failure. Based on the number of testcases, an appropriate channel report was prepared. This was again achieved with the help of typescript and Javascript, while simulation was performed using Postman.
- To enable service searches in the web application portal. Was done using javascript with availing the functionality delivered by ChartJS.
- To host the given project in the servers, using services such as Google Cloud Hosting. Etc. There is a possibility that Kubernetes akin services may be used for this given purpose, however these options are tentative and no concrete software was thus decided for this given purpose.

5. CONCLUSIONS

This project was initiated because of the overheads faced by the existing way the environment status determining was performed. These limitations were translated into new requirements and a new project “Environment Service status classification application” was designed and implemented, so as to overcome the limitations faced by the existing situation.

The project will be an improvised version of existing way to perform environment status determining with automation techniques implemented in it. There are two ways of implementing this idea. The first approach is to run the project on every individual system. This approach would require lesser efforts as the extensions are already implemented in the browser. However, this approach was not successful as the process of loading the shared library was not as expected. There was a need to import all of the code which was not intended[10].

The second approach is to integrate the Docker and Kubernetes functionality and to implement the extensions which are already implemented in the existing way of things functioning by using the which helps in retrieval of the category of the URL.

The limitations available as a part of the projects are as follows:

- The Visualizations in each segment of the web application may look a little cramped. Appropriate measures must be taken to improve the aesthetic appeal.
- Time Overheads from GET/POST calls using REST API.
- The front end and backend are running on two different server ports[11].

REFERENCES

- 1) Beimborn, Daniel, and Maximilian Palitza. "Enterprise app stores for mobile applications-development of a benefits framework." (2013)
- 2) Leiba, Barry. "Oauth web authorization protocol." *IEEE Internet Computing* 16, no. 1 (2012): 74-77.
- 3) Recordon, David, Dick Hardt, and E. Hammer-Lahav. "The oauth 2.0 authorization protocol." *Network Working Group 5849* (2011): 1-47.
- 4) Xylogiannopoulos, Konstantinos, Panagiotis Karampelas, and Reda Alhaji. "Clickstream Analytics: An Experimental Analysis of the Amazon Users' Simulated Monthly Traffic." In 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 841-848. IEEE, 2018.
- 5) Chitra, Lakshmi Prasanna, and Ravikanth Satapathy. "Performance comparison and evaluation of Node.js and traditional web server (IIS)." In 2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET), pp. 1-4. IEEE, 2017.
- 6) Jiang, Wei, Bin Zhou, and Meng Zhang. "Architecture Analysis and Implementation of 3D Theatre Display System Based on Node.js." In 2015 International Conference on Network and Information Systems for Computers, pp. 496-499. IEEE, 2015.
- 7) Tilkov, Stefan, and Steve Vinoski. "Node.js: Using JavaScript to build high-performance network programs." *IEEE Internet Computing* 14, no. 6 (2010): 80-83.
- 8) Scharl, Arno. *Evolutionary web development*. Springer Science & Business Media, 2012.
- 9) Gu, Xiao-Feng, Le Yang, and Shaoquan Wu. "A

real-time stream system based on node.js." In 2014 11th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), pp. 479-482. IEEE, 2014.

- 10) Faitelson, David, Robert Heinrich, and Shmuel Tyszberowicz. "Functional Decomposition for Software Architecture Evolution." In *International Conference on Model-Driven Engineering and Software Development*, pp. 377-400. Springer, Cham, 2017.
- 11) Moroney, Laurence. "Firebase Cloud Messaging." In *The Definitive Guide to Firebase*, pp. 163-188. Apress, Berkeley, CA, 2017.
- 12) Neumann, Andy, Nuno Laranjeiro, and Jorge Bernardino. "An analysis of public REST web service APIs." *IEEE Transactions on Services Computing* (2018).
- 13) Yang, Tao, Xiaoyi Man, Jiao Feng, Jie Chen, and Ran Tao. "Research on color expansion method in clothing recommendation system." In 2018 2nd International Conference on Data Science and Business Analytics (ICDSBA), pp. 482-487. IEEE, 2018.