INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 7, Issue 3 - V7I3-1645)
Available online at: https://www.ijariit.com

Web Based Network Element Resource and Information Tool

Tejamallik K. S.
tejamallik97@gmail.com
RV College of Engineering, Bengaluru, Karnataka

Suma B.
sumab_rao@rvce.edu.in
RV College of Engineering, Bengaluru, Karnataka

ABSTRACT

Web based network element resource and information tool (NERIT) which is the new technical way to maintain and keep track the network elements (NE) information It also keeps on updating NE information like software version, allocation period, NE connection details and NE health status. Users can access all the NE information details in a single window interface. NERIT is helpful for network service providers which reduces the network operation costs. While traditionally different solutions were available in the market for manual maintenance of NE details, but it is a tedious process and time consuming. In our proposed system, user can view all the required NE information. Proposed methodology, created a graphical user interface (GUI) for the NERIT with the help of web technologies and tools and it allows user to interact directly with the application. To maintain and monitor the NE's MySQL database management system is used. To connect the GUI and database management system Nodejs Express framework is used. The data will be stored in the remote server. The Admin or Superuser should be a register user. The admin or superuser can login into their account through the application itself and update the network elements information like NE IPs, credentials, their allocation period, availability, reachability, hardware details, software details, location etc. In NERIT application users have easy access for viewing and searching all the NE information. The application will check user authentications. Users are not permitted to manipulate any data only admin and superuser can manipulate the data. Other than this the advanced features are: Bulk addition of NE's, automating the allocation process and software version update. In case of any NE allocation is about to expire means it will send notification to the user for whom NE is allocated, so they can take required actions. The users can download the NE information file which includes all the NE details NERIT also has the feature for user feedback, which will enable to further enhance the functionality of the platform and add many more features to it. Performance of the application can be improved and speed up by using HTTP/2, optimizing the MySQL query and clustering the NodeJS.

***Keywords*:** *Network Elements, Web application, MySQL, NodeJS, feedback and performance*

1. INTRODUCTION

Web based network element resource and information tool which keep tracks the network elements (NE) information and it maintain the network elements details. A user can fetch the NE IPs, credentials, their allocation period, availability, reachability, hardware details, software details, location etc. Manual maintenance of details of Network Elements is tedious process and NERIT provides a solution by automating the process of information gathering and information processing related to NE's which will help to reduce the operational costs and human intervention with the rapid growth of the internet today, there is a rapid rise in the development of applications and websites. Storing data is important to maintain availability and to improve the performance. So application data has to be stored in such manner ,it can be access anywhere and anytime. Currently, a developer can use one of the following methods to link his or her application to a backend database: a.Third party cloud server. b. Implementing the back-end server. The data will be stored in the remote server. The Admin or Superuser should be a register user. The admin or superuser can login into their account through the application itself and update the network elements information like NE IPs, credentials, their allocation period, availability, reachability, hardware details, software details, location etc. In NERIT application users have easy access for viewing and searching all the NE information. The application will check user authentications. Users are not permitted to manipulate any data only admin and superuser can manipulate the data. Other than this the advanced features are: Bulk addition of NE's, automating the allocation process and software version update. In case of any NE allocation is about to expire means it will send notification to the user for whom NE is allocated, so they can take required actions. The users can download the NE information file which includes all the NE details. It also includes the feedback where users can mail their ideas for improvement of the application and if they find any bugs.

2.LITERATURE REVIEW

Main factors need to consider in web application development are:

- Security: The most important factor that need to be ensured in web application development is securing the data, because

it contains some sensitive data. It should ensure that only authorized and authenticated users gets access to the data.

- Availability: The database could crash at any time due to a virus or a failure of the operating system, at that time data may lost and unavailable. Availability is important factor that need to be ensured in web application development.
- Performance: Application should accept more users request per second and there should be less response time to the users or else it will create problem so users cannot access the pages.

Florian Haupt et al., in [1] have given a strong statement that REST APIs are highly important to apprehend distributed systems. They have more value in the context of cloud computing, IoT and microservices. The major issue with REST API is that there are no specific rules to design them unlike SOAP, which has a very standardized documentation. The quality attributes that are important with respect to REST may not be present at all. The authors have discussed a framework to provide a structural analysis of the REST API. The authors of [1] have discussed a canonical meta-model for REST APIs, which involves the integration of several meta-models. The meta-models are grouped into atomic model and URI model respectively. The general approach adopted by the authors is that the existing REST API description documents are utilized effectively.

Krisada Sangsanit et al., in [2] say that web services permit creation of new work processes and accelerate data exchange through a network. This is a dreary process when done manually. Therefore, automation of tasks of collecting components of a web service, development and planning of system's drive, minimizes the manual errors during web service composition. 18 such methods for web service composition have been surveyed. The pros, cons and the objections faced for each such methods is described precisely. The survey yielded the outcomes that 44% of the methods are algorithm updates or model driven, 28% are descriptor and planning methods, and the rest 28% are ontology and genetic algorithms. Based on the levels of automation, 0% techniques were manual, 22.2% were semi-auto and the remaining 77.7% were automated.

Neha Singhal et al., in [3] throw light on how services is the most important unit of deployment and is the most suitable building block. The authors have leveraged different repositories for different services. A thorough analysis of the distinct service performance measurement is done.

Alexandru Archip et al., in [4] discuss the impact of REST architectural style on the distributed application development. A model for the user data and session handling tasks is presented to demonstrate the most suitable approach to RESTful web service design procedures. The main goal is to make detailed analysis in order to give a standardized implementation of the RESTful web services.

Andy Neumann et al., in [5] provide an analysis of public REST web service APIs. The authors have selected Alexa.com's top 4000 most popular sites and recognized 500 websites that appeal that they provide the REST API. The paper [5] provides an empirical analysis of the 500 APIs based on the criteria of the key practices, adherence to the best practices and degree of compliance with REST architectural principles. The analysis is done on 26 API features. In the paper [5], 4000 sites were scrutinized, out of which 315 were duplicate sites. 18% of the remaining that is 681 provide some kind of API, and 13.1% which is equivalent to 681 do not provide any documentation at all. Finally, only 522 APIs are analyzed. Among these 522 APIs, 500 are REST. From the rest of 22 APIs, 2.5% belong to SOAP and 1.7% are XML-RPC APIs.

Roy Fielding et al., in [6] reflect how REST is a stepping stone to understand the working of WWW. The inadequacies, progressions and the history of REST, all are beautifully projected in consort with the architectural features that were taken as an inspiration. The usage of the application of the architectural styles, may have an effect on the understanding of the enterprise software architecture, keeping in par with its development.

Aditya Singh et al., in paper [7], have fabricated a lighter web framework that has its roots in the MVC architecture. This is advantageous in constructing small to medium-sized web applications. The authors have also emphasized on the benefits that the MVC architecture holds.

Carlos Rodriguez et al., in [8] have shown and analyzed how REST APIs are implemented in practice with respect to HTTP traffic collected by Italy's most famous mobile internet supplier. In paper [9] authors discussed about the software testing technology and J2EE architecture and in paper [10] authors compared about SOAP and REST web services .

3. PROPOSED SYSTEM

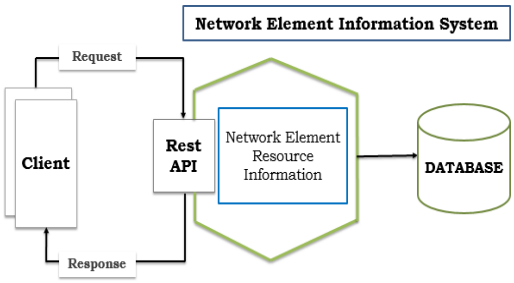

Fig 1: Network Element Resource Information Architecture

The proposed system architecture is shown in the fig 1, purpose is to present the main components and the basic relationships between them. According to the fig 1, there are four main processing areas in Network Element Resource Information Architecture:

- Client Management area: contains all the interfaces and processes to receive the events coming from the Rest API. Client makes a request to access the network elements resource information and all the information are stored in database.
- Rest API: It is used to fetch the request from client and respond in return with appropriate information.
- Network Element Resource Information: this is the "active" part of the system, the area containing the processes and mechanisms to execute the actions decided by admin. the most important elements in this area are the agents, which connect to the network element to execute the specific commands that change the behavior of the elements.
- Database Management area: this is the most important area to store all the information related to the resources of network element in most effective way. Sensitive data stored in a protected manner.

4. WORK FLOW OF NERIT

The workflow of the application is divided in to three groups follow as:

Normal User: User can view all the network elements details without login. In home page user can view all the network elements information and data table plugins is used for multiple order search and global search, by this search options users can search any network elements quickly without any time delay if more than 10000 elements were present. If user clicks on each network elements it redirects to the network elements details page. In that page user can view complete information about that network elements, information like NE location, ip, allocation period ,identity, owner ,hardware type and software type etc... .User doesn't have an access to modify any data. If user want to modify any data means they should login for admin or super user account.

Admin User: User can view all the network elements details with login. In admin home page user can view all the network elements information and data table plugins is used for multiple order search and global search, by this search options users can search any network elements quickly without any time delay. User can download the network elements information data file and if user need to insert more network elements means they can add all the information in the excel file and then they can upload that file. Before saving in the database, it will check whether any duplicated network elements is present or not. If it is present it won't save in the database it displays the message as "Element's information is already present" or else it will save in the database. For admin user, if user clicks on each network elements it redirects to the network elements details page. In that page user can view complete information about that network elements, information like NE location, ip, allocation period ,identity, owner ,hardware type and software type etc.., along with this they can see the credential details of the NE. User have an access to modify any data but user cannot delete any data and cannot modify allocation information, only superuser have an access to delete and modify the allocation details.

Super User: User can view all the network elements details with login. In admin home page user can view all the network elements information and they have an access to delete the information and modify the allocation details. Rest all it is same as an admin user. To secure the application REST full API's is used. For each user's session will be maintained so application is full secured .

Particular operations based on the data can be represented by a flowchart, in fig 2, flow chart of the NERIT is shown , A flow chart diagram is a method of representing a flow of data of a system or a process.

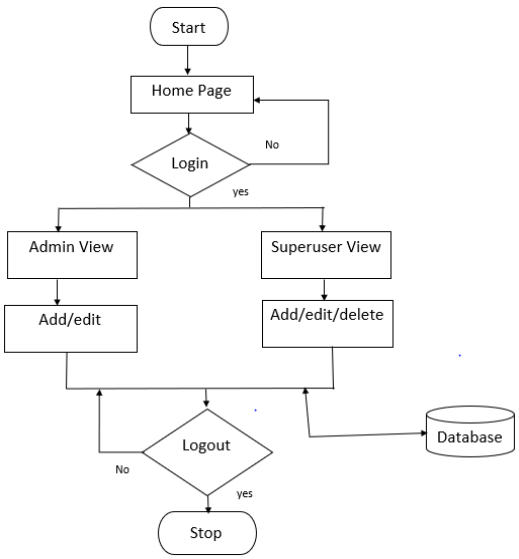

Fig 2. Flowchart of the NERIT

5. IMPLEMENTATION DETAILS OF NERIT

Mainly focused on the selection of the programming language platform selection which is dealt in detail.

Programming language selection: Node.js is a server environment that is free to use. Node.js is capable of creating dynamic page content. On the server, Node.js will build, open, read, write, delete, and close files. HTML provides the foundation for a website's structure, which is then expanded and updated by CSS and JavaScript. Presentation, formatting, and layout are all controlled by CSS. JavaScript is used to manage how various elements behave.

Development Environment Selection: OS selected is CentOS 8, which is an open-source Linux distribution. WinSCP is a free SFTP, FTP, WebDAV, S3 client, and SCP client for Windows that is open source. Its primary purpose is to move files between a local and remote device. NetSarang's Xshell is a remote access programme that also includes Xmanager and Xftp. SSH, Telnet, and other protocols are used by Xshell to handle remote servers. Stable connectivity solutions for managing Linux servers on Windows platforms are provided by Xshell and other programmes.

Data redundancy: Duplicate network element details may be present in the application, it creates data redundancy problem which cause data inconsistent, increases the size of the database unnecessarily and decrease efficiency of database and may cause data corruption. To avoid this problem algorithm 1 is designed which traces out the duplicate network element.

Algorithm 1: To traces out duplicate Network Element

```
Step 1: start
Step 2: Add network elements details in the GUI
Step 3: if network element name and ip > 0 :
            Display message "Information is already existing" in the GUI
        Else:
            save the network elements details in the GUI
Step 4: stop
```

Checking network element health status: Each time ping command has to be executed manually for each network element. It is a mundane activity and updating the status of each network element in the application is complex. To reduce the complexity of work algorithm 2 is designed which will execute the ping command for each network element for every 12 hours.

Algorithm 2: To execute ping command for every 12 hours

```
Step1: Start
Step 2: Import modules ping, async, cron
Step 3: Execute mysql query to fetch the NE id, IP from the database
Step 4: for i=0 to results.length
Step 5: Declare variable neid and neip
Step 6: Initialize variables
            neid ← results[i].id
            neip ← results[i].ip
Step 7: for each element in the neid,neip
            execute ping command
            declare variable ping
            ping ← results
Step 8: Execute mysql query to update the results in the database
Step 9: Repeat steps from 6 until i >=results.length
Step 10: Stop
```

6. RESULT ANALYSIS

Performance can be improved by optimizing the query and by clustering the nodejs .

Table 1:Performance analysis for without optimizing the query and with optimizing the query

Actions Performed	Time taken to perform in sec (without optimizing the query)	Time taken to perform in sec (with optimizing the query)
Redirect to next page	8 sec	1 sec
Load the page	6 sec	.5 sec
Inserting data	12 sec	2 sec
Deleting data	4 sec	.5 sec
Update data	10 sec	2 sec

Table.2: Performance analysis for without clustering the nodejs and with clustering the nodejs

Actions	Without clustering nodejs	With clustering nodejs
Accepting user requests	1-100 users per sec	1-500 users per sec
Response time	0.75 - 2 sec	0.05 – 0.5 sec

From above two tables we can conclude that application can be speed up and improve the performance by optimizing the query and by clustering the nodejs. By default, Node.js runs on a single thread on a single-core of the processor. and It does not utilize several cores that may be available in a system. with cluster on Node.js, it allows easily create child processes that all share server ports. it's meant that cluster can handle a large volume of requests with multi-core systems. And automatically, it will increase the performance of the server.

7. CONCLUSION

The proposed system's main goal is to simplify the use of network elements information, which will help minimise operating costs and enable users to retrieve information in a single window interface. Owners of network element would benefit from the platform's functionality, which will save them a significant amount of time by eliminating all manual intervention activities. It always provides correct details. The information collected over the years can be saved and accessed at any time. The information contained in the repository enables management in making informed decisions. As a result, a web-based network element information management system is preferable. Both users and management will get the details they need right away.

8. REFERENCES

[1] F. Haupt, F. Leymann, A. Scherer and K. Vukojevic-Haupt, "A Framework for the Structural Analysis of REST APIs," 2017 IEEE International Conference on Software Architecture (ICSA), Gothenburg, 2017, pp. 55-58.

[2] K. Sangsanit, W. Kurutach and S. Phoomvuthisarn, "REST web servicecomposition: A survey of automation and techniques," 2018 International Conference on Information Networking (ICOIN), Chiang Mai, 2018, pp. 116-121.

[3] U. Sakthivel, N. Singhal and P. Raj, "RESTful web services composition & performance evaluation with different databases," 2017 International Conference on Electrical,Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), Mysuru, 2017, pp. 1- 4.

[4] A. Archip, C. Amarandei, P. Herghelegiu, C. Mironeanu and E. şerban, "RESTful Web Services – A Question of Standards," 2018 22nd International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, 2018, pp. 677-682.

[5] A. Neumann, N. Laranjeiro and J. Bernardino, "An Analysis of Public REST Web Service APIs," in IEEE Transactions on Services Computing, doi: 10.1109/TSC.2018.2847344.pp.1-1

[6] T. Fielding Roy & N. Taylor, Richard & R. Erenkrantz, Justin & M. Gorlick, Michael & Whitehead, Jim & Khare, Rohit & Oreizy, Peyman. (2017). Reflections on the REST architectural style and "principled design of the modern web architecture", pp. 4-14.

[7] A. Singh, P. Chawla, K. Singh and A. Kumar Singh, "Formulating an MVC Framework for Web Development in JAVA," 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, 2018, pp. 926-929.

[8] Rodríguez, Carlos, Marcos Báez, Florian Daniel, Fabio Casati, Juan Carlos Trabucco, Luigi Canali and Gianraffaele Percannella. "REST APIs: A Large-Scale Analysis of Compliance with Principles and Best Practices." ICWE (2016), pp. 21 – 39.

[9] S. Lijuan, W. Shujuan and W. Yanli, "Research and Application Based on J2EE Software Architecture of Web Application System Testing Methods," 2016 International Conference on Robots & Intelligent System (ICRIS), Zhangjiajie, 2016, pp. 289-291.

[10] S. Kumari and S. K. Rath, "Performance comparison of SOAP and REST based Web Services for Enterprise Application Integration," 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Kochi, 2015, pp. 1656-1660.

