

Social Media Posting Platform using Node.Js

Hradsh Kumar (Assistant Professor), Devanshi Gupta, Kalpit Kulshrestha
SCSE Galgotias University Noida,India

Abstract:- Social media includes social networking websites that help build relationships between users from different domains, leading to a richer social structure. User-generated content encourages asking and making decisions. Considering the relevance of the communication platform to the various stakeholders, it has received significant attention from researchers in a variety of fields, including information systems. Node.js provided the opportunity for Full Stack Engineers to manage the server and the client themselves. Node.js is fast and reliable for heavy files and heavy-duty network applications due to its event-driven, unrestricted, and inconsistent systems, where developers can also keep complete projects on a single page (SPA) and can use IoT. The result of the study concludes with a study and review of the literature on the areas of use and challenges of Node.js. Finally we will offer a suggestion on how to improve to overcome the challenges.

I. INTRODUCTION

Web-based applications increase its popularity as they become easier to upgrade, keep them safe and secure. They are also easily accessible to customers and do not require additional installation in most cases and are customized quickly [1]. A web application is based on a web-based system, which has the added function of using the organization's business mind. These applications are completely web-based instead of needing to install a separate application on the operating system [2]. Google Docs, web-based retail stores, Google Maps, and web-based email apps are Web applications [1] [3]. The Web development industry will receive two types of developers. namely Front Engineers and Back-end developers [3]. Previous engineers need knowledge of HTML, CSS, and language editing to add results and much more to the front - ultimately JavaScript. They create a display of web sites and results that are displayed to customers by modifying the design of the designer [3]. Developers develop a business concept behind any web application. Actions such as adding and retrieving great news to or from web applications, or sending email from web-based forms, or verifying the authenticity of visitors or clients are all part of the background developers. Background developers need to know languages such as PHP, .NET, Java, and other [3] [4]. Back developers must also have knowledgeable information such as My SQL, Oracle, and SQL Server, or they must hire or trust an administrator. website to work with flow. The database administrator will take care of the database server and ensure its smooth operation [5] [4]. The full stack developers are the jack of all trades, and they are the ones who do it all. In particular rear engineers are required to have advanced and vice versa engineers skills and have more responsibility for

them to learn the additional skills set [4]. and Java script and HTML and CSS, Server Side Languages like PHP, .NET, Java, Ruby, etc., and database information like SQL Server, MySQL Server, and Oracle. As shown in Figure 1. Figure 1: Full Stack Developer Node.js overcomes the problem of learning multiple languages at once to become a Full Stack developer where a leading JavaScript developer who just needs to learn more. Server Side Functions also becomes a Background developer [6] [7]. Apart from this where Node.js is free, it is also used by thousands of developers around the world [8]. While the web application requires functionality and scaling, history has already adopted configurations and events W © 2017 Global Journals Inc. (US) Author α σ: Department of Computer Science, SZABIST Dubai Campus, UAE. Emails: Email@hezbollahshah.pk, tariq@szabist.ac.ae
Keywords: node.js, javascript, SPA,

IoT. Global Journal of Computer Science and Technology Volume XVII Issue II Version I 73Year 2017
() E Client Side Language ie JavaScript Server Side Language ie PHP, .NET, Java, etc. Database Side Language ie SQL Server, MySQL Server, etc. separately or its combination. But the increase in its daily internet traffic requires better and newer solutions to improve the type of service at the same time. One solution is Node.js (sometimes also called Node [9]) which is lightweight, and meets the requirements of an event-enabled and unrestricted I / O model and JavaScript near the server [10] [11]. The Chrome V8 JavaScript Engine is the core of Node.js as the Node.js JavaScript operating system is built on it [11]. Written in C ++, V8 is an open source source from Google as the most efficient JavaScript engine. Google has used V8 in the Google Chrome browser, and it can work independently and can be embedded in C ++ applications. It is able to use different versions of Windows, Mac OS, and Linux [12]. Since Node.js is event-based and not network-based, it can scale up to millions of connections at once, while using the event loop between a single series and not over multiple threads. Node.js evaluates its I / O performance appropriately and is not well understood by AJAX. Although Node.js and AJAX are very different [9]. a) Benefits of Node.js above others Node.js is built from the ground up for the purpose of managing incompatible I / O as it is built with JavaScript and JavaScript is built as an event loop. Like a button-click event on the client side JavaScript is an event loop. Although some sites have this feature, they use it for third-party libraries or are not built for the same purpose as Node.js and that is why they are often slow, or slow and not a common feature in them. . Other examples are an event machine - designed by Ruby, Twisted - licensed under the open source MIT License, introduced via Python and available from Python 2 onwards, as well as the Apache MINA network framework library also called

“Network. Socket Library ”is another example of providing event and functionality that is limited to APIs only. Similarly Apache AsyncWeb was built using Apache MINA and Perl's Any Event. Similarly the edge of Node.js more than any other will be that it will be able to manage multiple applications while it will act as a client in third-party services by extracting only one series. Some languages in this regard will block processing until the remote server first responds to its initial request and will therefore need additional processing to perform. Compared to Node, all you will use is asynchronous as it will be difficult if you write inconsistent code. And Node.js will never force data entry into the bar before downloading while others, like Event Machine, force the final volume in most cases to store data [7] [6] [13] [14] [15] [16] [17]. As part of the JavaScript server, one of Node.js' most admirable benefits over others

III. LITERATURE SURVEY

The most important thing about a web server is its ability to manage multiple users effectively. This is closely related to the editorial language used to write its text. Thus the performance of third-party writing scripts such as PHP, Python was considered compared to Node.js [8]. P. S. Bangare et al proposed a secure writing approach for a novel combination of a mixed transport map and RC4 stream cipher [10]. Node.js is an excellent tool if you are looking for some kind of live interaction, real-time results. Able to deliver data very quickly / from web server. Traditionally, there has been a major problem with computers where the CPU is able to do one thing at a time. It was solved long ago by loading multiple threads, allowing us to have multiple 'threads' in one CPU. It changes between them all the time, and while it is very fast, the change has a high tone. To avoid this extra node.js solves this problem using a single thread, which is run by the event. Instead of creating a new series for each application, there is one series for every single application. When a new one arrives, it explodes an event that uses a specific code. When you make a call to a website, for example, instead of blocking it, it simply performs a call again after the call is completed. Any number of call-backs can respond to any event, but only one activity will ever be done at any time. Everything else your system can do — such as waiting for data in a file or an incoming HTTP request — is hosted by the Node, similarly, behind the scenes. Your application code will never be used at the same time as the most important thing about a web server is its ability to manage multiple users effectively. This is closely related to the editorial language used to write its text. Thus the performance of third-party writing scripts such as PHP, Python was considered compared to Node.js [8]. Node.js is an excellent tool if you are looking for some kind of live interaction, real-time results. Able to deliver data very quickly / from web server. Traditionally, there has been a major problem with computers where the CPU is able to do one thing at a time. It was solved long ago by loading multiple threads, allowing us to have multiple 'threads' in one CPU. It changes between them all the time, and while it is very fast, the change has a high tone. To avoid this extra node.js solves this problem using a single thread, which is run by the event. Instead of creating a new series on each application, there are

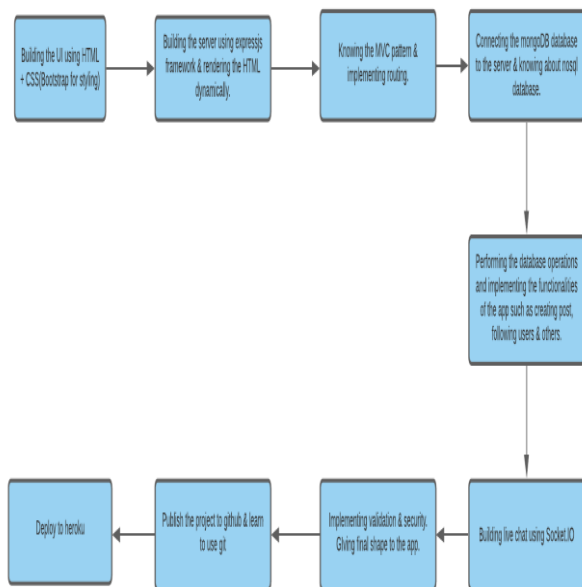


Fig. 1: Dataflow diagram

II. BACKGROUND

A. APPRISING

Software development is usually done by different teams working in different areas and so the need for high collaboration between teams is essential for the project to be successful. Common ways of documenting needs, creating test cases, tracking errors have not been changed to improve the improved communication the current generation has. There is a need to create a new way of performing software engineering tasks in a more efficient way. This will help to implement current social media strategies to implement effective processes from existing software development methods.

A. Limitations

- a) Lack of private involvement:

AI tools will work with minor errors that will not affect the diagnostic or performance process significantly. not like computers, doctors will break certain rules to try their best to avoid wasting human life. The private sector is a key stakeholder in both urban and economic development, being a major contributor to national income and the principal job creator and employer. The private sector provides around 90% of employment in the developing world (including formal and informal jobs), delivers critical goods and services and

contributes to tax revenues and the efficient flow of capital.

b) Increased levels of state among aid workers:

As AI is forced into the entire full-fledged aid system, a number of previously man-made tasks will be automated today. Chatbots and robots will provide a mental state easier, analyze the patient's health status, and anticipate other issues such as fainting, sepsis, illness, etc.

c) Opportunity for Misdiagnosis:

Accurate diagnosis of a definite illness depends on the amount of information gathered from a few people with similar symptoms and conditions. To encourage proper comparisons, AI information should contain respectable information about patients of the actual collection. Therefore, if there is a shortage of data for a few people from a specific domain, AI will provide AN inaccurate diagnosis.

d) Social Discrimination:

Social discrimination is defined as the continuing inequality between people due to illness, disability, religion, sexual orientation, or any other measure of diversity. AI-based devices cannot fully identify the identity and background that make them biased towards diagnostic patients.

B. Future work

The study found findings regarding the implementation of Node.js. Below is a discussion of the use of positive results as a result of book reviews and surveys. Node.js has made Full Stack Engineer a dream come true. In the absence of Node.js it was difficult for the developer to learn a few different languages and situations in order to manage the complete server side and client side. Organizations and developers are now able to launch Node.js to build more tolerant and faster apps and with Single Page Applications (SPA) now server calls are being reduced and apps are much simpler and faster. Node.js has made it easy to perform high-speed tasks such as image processing and GIS internet at high speeds, and can be used reliably across file sizes or network bandwidth. Node.js will make such tasks faster and have less bandwidth demand. Phakathi The public prefers its feature that the same language is used on the server side while JavaScript has been used on the client side for many years. Node.js has some challenges in the context of its public use and its acceptance by developers and organizations with existing planning technologies. There is no doubt that Node.js has some great benefits, and it has some community challenges. One such challenge is the ability to misuse a language commonly used by engineers. Another enthusiast has created background software that uses Node.js in Raspberry Zero. It can create backdoors on a targeted computer and their network even if the computer is locked with a password. While there are solutions from such sites but some do not seem to work as a complete blocking of USB ports, and shutting down the web browser

every time a user leaves the computer. And some options are not made by many and may not be known about it such as using a secure layer on single websites (https), as well as enabling secure flags cookies that ordinary users may not be aware of [45]. There is a compelling point but as understood in a study that made the community find it difficult to read JavaScript for Node.js Also developers with knowledge of other programming languages have difficulty accepting Node.js. Even setting up a server with their editing function is not an obstacle. This is the result of research. Other integration points include Event-Driven Programming, Non-Blocking I / O, and compliant feature. But according to research findings factors such as event-driven programs, Non Blocking I / O, and Asynchronous processing are a hindrance. As a result of the research, the challenge arises that many organizations are not yet ready to use new technologies such as Node.js in addition to existing ones such as PHP, .Net, etc. And there is a lack of market awareness that creates a barrier to using Node.js to use. At the developer level, there is a challenge arising from the research results that they do not find it easy to read a website that works with JavaScript. And there seems to be a lack of information among developers as the survey reveals those who answer questions about key factors such as the I-O event, unrestricted and inconsistent processing makes a decision about this. The public can be taken to Node.js for small training efforts, as well as conducting workshops to introduce Node.js to new and existing programmers and explain the benefits of Node.js features such as event-driven programs, I / O prevention, and asynchronous. The community needs to be updated about the signs of how quickly Node.js performs network-related tasks especially when it comes to the challenge of managing large network files and managing multiple callbacks in other languages © 2017 Global Journals Inc. US) Global Journal of Computer Science and Technology Volume XVII Issue II Version I 81Year 2017 () At Node.js . and the speed of applications built on Node.js.

IV. DATA ANALYSIS

In the real world when we want to send short tweets, shayari and any image to the world, so that one can share his thoughts with the world. like "OurApp".To improve that forum we used JavaScript, HTML, CSS. You also use the socket.io feature to chat with other people who follow you. You can use it among your friends or family and use it as a real small world community media application. This is a Node.js application developed using the ExpressJS framework to build a web server and project according to the MVC pattern. Install the required dependency packages using npm. Download all the useful tools.

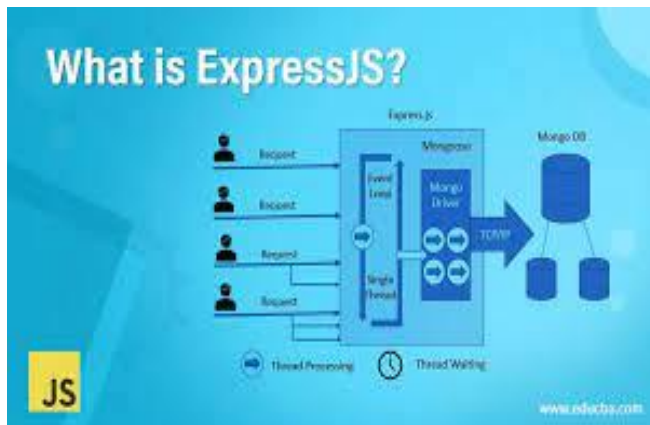


Fig. 2: ExpressJS Diagram

First, we design the basic UI of the visitor page and the login page to show the original effect of the product we are working on. We design the layout using HTML, CSS, splitting the front area into: home page, dashboard page, send page, profile page, single page screen, outgoing chat page. At the end of this task, html pages should be created. Creating a single page of 404 error code can be a plus of the app.

start building the server using express. Express.js, which is a simple framework for creating web servers, makes it easy to configure your application with middleware and router. After building the server, your project folder should contain server.js file and package.json, package-lock.json, and node_modules. Now create a folder with a view name inside your main folder and insert all HTML files inside it and create a public folder and insert CSS files in it. Use EJS; allows you to embed javascript into HTML templates i.e., create powerful content in your templates. It's faster and easier than using a full frame like Angular or React, though a slightly richer feature.

Use nodemon to restart your server Nodejs application automatically every time a change is saved in your code.

Build your templates in a way that can be reused using sections. Like not using the header and footer repeatedly on all pages, you are using situations to provide divisions (div).

At the end of this activity, your server should listen to requests in the port. Also, your view folder should contain sections and pages using EJS. The terminal should look like this after using the nodemon when starting the server.

- Use local variable to maintain the connection cable. Create an .env file and keep private keys and attributes there and use them wherever needed. This ensures the security and privacy of your website collection.

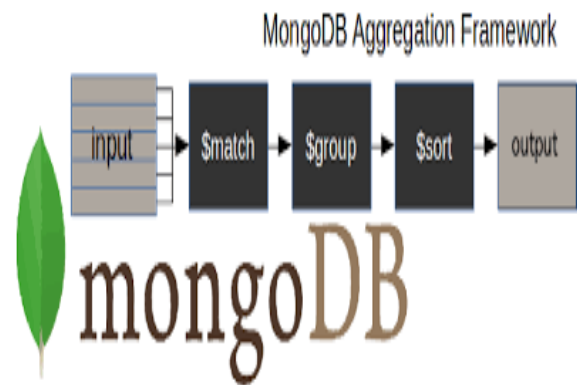


Fig. 3: MongoDB Framework

- Use the dotenv package to access natural variables.
- You can use MongoDB Compass - An easy way to test and manipulate your MongoDB data. It provides a GUI and is easier to use than the MongoDB or Mongo Shell atlas.
- Apply routes to all routes in a separate file and call manager functions, respectively.
- Separate controllers and models w.r.t their functionality. For example, perform user-related tasks on userController.js and models on userModel.js. Confirm attributes before saving it to a website. As such, divide others based on performance.
- Use hashing before storing passwords on a website. In hashing bcryptjs and md5 packages can be used.
- You can use MongoDB without Mongoose too. But using a mongoose makes it easier to use the functions.
- Create an instant message template in case of error.
- Build a website for the post by incorporating all the features needed to save the job.
- Write your controller function to save and edit user posts.
- Build a model in such a way that followers and followers can be accessed by the user. (Hint: Use the collection within the collection, there is no external key on the SQL website)
- Download posts for users to follow in their dashboard.
- NodeJs CRUD Operations Using Mongoose and MongoDB Atlas
- CRUD performance with Mongoose and MongoDB Atlas
- CRUD Operations in Express, Nodejs, and MongoDB
- Data Modeling
- One-plural Relationship Model With Embedded Documents
- Node.js: create a simple social network with ExpressJS
- Learn the models and functions of the mongoose ## Bring it!
- Tasks should be done properly ,.
- Make all routes guaranteed / protected.

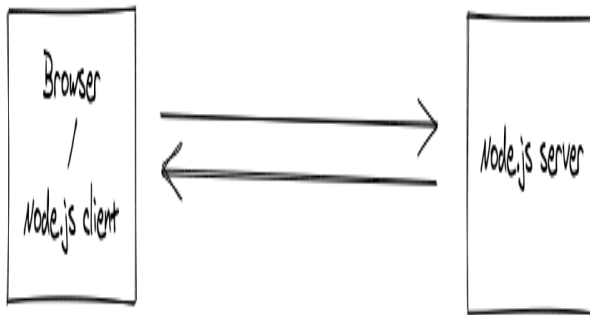


Fig. 4: client & server interchange

- Try to create a private conversation with users again. See
- Try to use live search features by users and even if you can create your own search way.
- Tip: use `sanitize-html` and `dompurify`
- Create one separate `frontend-js` folder and install `webpack`, `webpack-cli`,
- `@babel / core`, and `@babel-loader`.
- Customize the `webpack.config.js` file and change the `clock` option accordingly to
- `package.json` file.
- Now use the `npm clock` and check the `public` folder, it will have `main-bundled.js`
- file in it.
- Start creating previous files to search with chat panels.
- Create task controller for search results by post.
- Use `socket.io` to build chat performance.
- Use live authentication as a separate `frontend-js` pre-made for live search.
- Disable Cross-Site Request Forgery (CSRF) in your application using the `csrf` package.
- Launch the project as a github repository using `git init`.
- You will see an `.git` file inside your repository and random files.
- Ignore `node_modules` and `.env` file (if available) before uploading it to GitHub using
- `.gitignore` file.
- Make your changes and push them to github.
- Continue to Heroku and create your free account. Dashboard is amazing
- it's simple and easy to use so check it out.
- If you have a pair of private key values in your `.env` file, use the `config vars` in the edit menu of heroku to store those values before feeding.
- Now use the command line to run your node program.

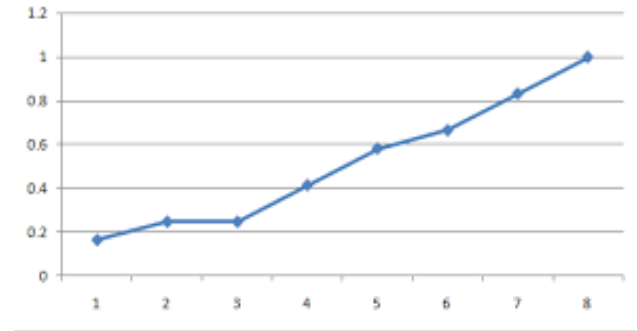


Fig. 5: chart

V. CONCLUSION

This paper introduces a framework using `node.js` to build a very dangerous and high-speed backend database server for web developers and app developers. It also demonstrates the use of a NoSQL website such as `Monodb` in the proposed project work in addition to another traditional website such as `MySQL`.

REFERENCES

- [1.] "Benefits of web-based applications," [Online]. Available at: <http://www.magicwebsolutions.co.uk/blog/the-benefits-of-web-basedapplications.htm>. [Accessed 25 November 2016].
- [2.] Web Application Basics, Pearson Higher Education.
- [3.] F. Bridge, "What Types of Engineers Are There ?," tree house, 24 June 2016. [Online]. Available: <http://blog.teamtreehouse.com/what-types-of-developer-are-there>. [Accessed 25 November 2016].
- [4.] M. Wales, "Front-End vs Back-End vs Full Stack Developers," Udacity, 08 December 2014. [Online]. Available: <http://blog.udacity.com/2014/12/front-end-vs-back-end-vs-full-stack-web-developers.html>. [Accessed 25 November 2016].
- [5.] J. Long, "I Don't Speak Your Language: Frontend vs. Backend," tree house, 25 September 2012. [Online]. Available: <http://blog.teamtreehouse.com/dont-speak-your-language-frontend-vs-backend>. [Accessed 25 November 2016].
- [6.] Mardan, "PHP vs. Node.js," Programming Weblog, [Online]. Available: <http://webapplog.com/php-vs-node-js/>. [Accessed 28 January 2016].
- [7.] J. Kaplan-Moss, "Quora," [Online]. Available: <https://www.quora.com/What-are-the-benefits-of-developed-in-Node-js-versus-Python>. [Accessed 29 June 2016].
- [8.] "Node.js Tutorial," learning point, [Online]. Available at: <https://www.tutorialspoint.com/nodejs/index.htm>. [Accessed 25 November 2016].
- [9.] N. Chhetri, "Comparative Analysis of Node.js (Server-Side JavaScript)," Climax Projects in Computer Science and Information Technology., P. 5, 2016.
- [10.] RR McCune, "Node.js Paradigms and Bench marks," 2011.

- [11.] Node.js, "Node.js Home Page," Joyent, 2016. [Online]. Available: <https://nodejs.org/zu/>. [Accessed May 01, 2016].
- [12.] G. Developers, "Chrome V8 | Google Developers," Google, [Online]. Available: <https://developers.google.com/v8/>. [Accessed 27 May 2016].
- [13.] event machine, "GitHub, Inc.," [Online]. Available: <https://github.com/eventmachine/eventmachine>. at Accessed 30 June 2016].
- [14.] Twisted Matrix Labs, "Twisted Matrix Labs," [Online]. Available: <http://twistedmatrix.com/trac/>. [Accessed 30 June 2016].
- [15.] Apache Software Foundation, "Apache MINA," [Online]. Available: <http://mina.apache.org/>. [Accessed June 30, 2016].
- [16.] Apache Software Foundation, "Apache MINA," [Online]. Available: <http://mina.apache.org/asynch-web-project/index.html>. [Accessed 30 June 2016].
- [17.] Ryan Dahl: Original Node.js presentation. [Film]. Youtube, 2012.
- [18.] "How Uploading Time Affects Your Bottom Line," Kissmetrics Blog, [Online]. Available: <https://blog.kissmetrics.com/loading-time/>. [Accessed 25 November 2016].
- [19.] Buckler, "Site Point Smack Down: PHP vs Node.js," Site Point, [Online]. Available: <http://www.sitepoint.com/sitepoint-smackdown-php-vs-node-js/>. [Accessed 28 January 2016].
- [20.] Firehose, "Firehose," Firehose Project, [Online]. Available: <http://blog.thefirehoseproject.com/posts/nodejs-vs-rails/>. [Accessed 30 June 2016].
- [21.] "Quora," [Online]. Available: <https://www.quora.com/What-evil-use-Node-js>. [Accessed 30 June 2016].
- [22.] "Hashnode," Hashnode, [Online]. Available: <https://hashnode.com/post/what-are-the-actual-dis-benefits-of-using-nodejs-ciibz8fd3017yj3xtxqz1r9hs>. [Accessed 30 June 2016]
- [23.] Flickr, "Flickr, Yahoo Company," Yahoo, [Online]. Available: <https://www.flickr.com>. [Accessed 18 November 2016].
- [24.] A Harris, "The Birth of a Node: Where Did It Come From? Creator Ryan Dahl Share History," Silicon ANGLE, 01 April 2016. [Online]. Available: <http://siliconangle.com/blog/2013/04/01/the-birth-of-node-where-did-it-come-from-creator-ryan-dahl-shares-the-history/>. [Accessed 18 November 2016].
- [25.] L. ORSINI, "What You Should Know About Node.js," read the article, 07 November 2013. [Online]. Available: <http://readwrite.com/2013/11/07/what-you-need-to-know-about-nodejs/>. [Accessed 2016 November 2016].
- [26.] G. Nemeth, "Rising Stack Engineering," [Online]. Available: <https://blog.risingstack.com/history-of-node-js/>. [Accessed 27 October 2016].
- [27.] R. Dahl, "Porting Node to Windows With Microsoft's Help," Node.js, 23 June 2011. [Online]. Available: [https://nodejs.org/en/blog/uncategorized/porting](https://nodejs.org/en/blog/uncategorized/porting-node-to-windows-with-microsofts-help/)
- node-to-windows-with-microsofts-help /. [Accessed 18 November 2016].
- [28.] P. Krill, "Q&A: Why io.js decided to launch Node.js," JAVAWORLD, 04 December 2014. [Online]. Available at: <http://www.javaworld.com/article/2855639/open-source-tools/egg-why-io-js-decided-to-fork-node-js.html>. [Accessed 18 November 2016].
- [29.] "Node.js Foundation Advances Community Collaboration, Announcing New Members and Certified Technology Management," Linux Foundation, 15 June 2015. [Online]. Available.