
Team 20 Assignment 3

Simon Caven 11966939

Locky

IoT Analysis of Alternatives

21st April 2017

OVERVIEW

This report seeks to analyse the best alternatives for the development of an IoT Application with server-client communication. The application is required to exchange sensor data between the IoT device and the server. This data is then communicated to the client. The development of the application will be in the Node.js framework.

ALTERNATIVES

IoT - Server

For the IoT-Server data exchange there are two alternatives we will consider: Johnny-Five and Serialport. These are both available as Node.js packages.

Johnny Five

Serialport

Serialport is a Node.js package that enables access to the serial ports of an Arduino Uno IoT device for reading and writing. Data can be exchanged between a Node.js application and the Arduino board using serial communication. The Arduino board is accessed and set through an uploaded Arduino sketch which then sends its data to a Node.js application.

Server - Client

For server-client communication we will consider the following two alternatives: Firebase and Socket.IO.

Firebase

Firebase is a Google owned mobile and web application platform which provides the developer with the tools and infrastructure needed for their projects. Firebase consists of complementary features that can be mixed and matched. It is a full application suite that includes: Analytics, Cloud Messaging, Authentication, a Real-time Database, Storage and Hosting [1]. We will use only the real-time database in our spikes and consider the possible use of these other features in the analysis.

Socket.io

Socket.IO is a Javascript library for realtime web applications. It provides an event driven API similar to Node.js that enables web clients and servers bi-directional communication in real-time. The library consists of two parts: a client-side library that runs in the browser and a server-side library for Node.js (implementations exist for other languages too). Both of these components share an almost identical interface [2]. Socket.IO has the ability to provide real-time analytics, binary streaming, document collaboration and instant messaging. It has found widespread commercial use with many notable users including Trello and Microsoft Office [3].

CRITERIA

The analysis of alternatives will focus on the following criteria:

Learning Curve

- How easy is the framework to learn?
- How much time does the framework take to learn?

Ease of Programming

- How easy is programming and developing in the framework once learnt?

Support

- How much support is available in the form of online forums, manuals, documentation etc?

Communication Time

- What is the average response time between the server and the client for communication?

Integration (may omit)

-
- How well does the framework integrate within the application?

Extensibility and Flexibility

- How easy is it to extend or change the application when using the framework?

RESULTS

The four alternatives were explored through four small spike codes designed to receive data from the Arduino board and send it to the client. These spikes can be found at

https://github.com/sliggady/FIT3140_A3_Team20.git

Johnny Five

Learning Curve

Ease of Programming

Support

Integration

Extensibility and Flexibility

Serialport

Learning Curve

Ease of Programming

Support

Integration

Extensibility and Flexibility

Firebase

Learning Curve

The learning curve for the Firebase framework is the longest of all the frameworks. The API is extensive and novice developers may find it difficult to pick up. Experienced developers with no prior Firebase knowledge will need time to learn to and master the new framework. For use with Node.js, a firebase administration library is required and for ease of development it is recommended also installing the firebase command line tools Node.js package. Setting up Firebase in the Node.js application on the server requires first establishing a Firebase account. This account can then be linked to in the Node.js application. Linking to Firebase from the client then requires only some minimal scripting code.

Ease of Programming

Writing data to the real-time database is easily coded with Firebase allowing for data to be stored in any format via their No SQL database. The API for retrieving data from the database is more difficult with listener events being the most commonly used method. Queries can also be constructed allowing for ordering and filtering although the interface to do this is tricky.

Support

The Firebase website provides an excellent reference for all of its API's. All methods and classes are documented and examples provided. A search of 'firebase stackoverflow' reveals a hit of 213,000 for the popular coding support site and Firebase applications and code are appearing on Github in an ever increasing number. As a relatively new framework it is expected support channels and sites for Firebase will increase in the future.

Communication Time

Server-client communication using Firebase is achieved through the server writing to the database and the client then listens for these new additions. Due to this, server-client communication suffers when compared with Socket.IO. The average response time over 20 messages was 260.95 ms when integrated with Serialport in Spike 1 (fig1).

Integration

Extensibility and Flexibility

This is where Firebase excels with its rich infrastructure and vast application suite. The real-time database could be utilised for storage and allows for greater programming options and the flexibility to provide additional application features. The Firebase hosting service could be employed if needed as well as the messaging and storage facilities. Firebase provides the ability to extend any application to its full potential and reach.

Socket.io

Learning Curve

Ease of Programming

Support

Communication Time

Integration

Extensibility and Flexibility

CONCLUSION

REFERENCES

[1] Firebase. (2017, April 10). In *Wikipedia, The Free Encyclopedia*. Retrieved 12:23, April 14, 2017, from <https://en.wikipedia.org/w/index.php?title=Firebase&oldid=774835066>

[2] Socket.IO. (2017, March 17). In *Wikipedia, The Free Encyclopedia*. Retrieved 11:36, April 14, 2017, from <https://en.wikipedia.org/w/index.php?title=Socket.IO&oldid=770805388>

[3] From <https://socket.io/>, Retrieved 21:38, April 14, 2017

FIGURES

SPIKE2 - FIREBASE RESPONSE TIMES

LAST RESPONSE TIME

281

Count

20

Running Average

260.95

SERVER

1492154197506

CLIENT

1492154197787

*****Milliseconds since 1/1/1970*****

Figure 1. Average response times for Firebase with Serialport.