GLOBUS

Globus An International Journal of Management & IT
A Refereed Research Journal
Vol 12 / No 2 / Jan-Jun 2021 ISSN: 0975-721X

AN OFFLINE PUSH NOTIFICATION FOR MOBILE DEVICE

*Jose Naldrix D. Rivera

Paper Received: 17.12.2020 / **Paper Accepted**: 15.01.2021 / **Paper Published**: 16.01.2021
Corresponding Author: Jose Naldrix D. Rivera; Email: josenaldrix.rivera@chmsc.edu.ph; doi:10.46360/globus.mgt.120211001

Abstract

Every business has the objective of minimizing cost and maximizing profit while adapting to the new technology in order to compete with their competitor. One of the technologies that every business must adopt is using mobile devices and its application to channel their business information and view it with convenience inside the business organizational grounds. Sadly, every available push notification service implemented to communicate between the business server computer and client's mobile devices is internet-dependent and known to be very expensive. Offline push notification for mobile devices is developed to address the underlying issue that would eliminate the internet-cost in operating the system.

Keywords: Mobile Push Notification, Offline Push Notification, Zero Cost Push Notification, Mobile Device, Mobile Offline Notification.

Introduction

In this fast-paced, changing world brought by technology, every business has to adapt to what is trending and latest for the sake of survival in the fierce competition against its competitors [8]. It is evident that every person in this mobile age had already owned a mobile device and became the business world's business target. The mobile device becomes a game-changer and a trendsetter to business as to how the transaction between the business firms and clients should be done in a more innovative and efficient way. Aside from that, it also impacts the business processes and employees in the business grounds since it may improve the data communication drastically. A mobile device like smartphones has become a primary gadget in the business as it channels all the data about business status using a mobile phone application. Data may disseminate from the computer to the mobile device of an employee through a messaging service like Short Message Service (SMS), and the other one is the so-called push notification.

However, messaging is indeed essential to the business, but it is not a free service. We all know that every SMS costs an amount in which if it is accumulated, it will become a huge amount [1]. And just like SMS, push notification is also expensive since it requires internet [2], which demands to pay internet fee both to the server and client of the system. A push notification service is a messaging service in which data is delivered from a server computer through a server application to the mobile device over a network [7]. The common push notification is known to be server push notification wherein the request of data is initiated by the server computer rather than the mobile device. Famous examples of this are Firebase Cloud Messaging (FCM), and Twilio requires the internet [3] as a network between them and the client's mobile device to channel the data. Moreover, since it is an internet-based service, and it may cause a delay in delivering the message due to internet failure or instability [4]. Thus, the development of the offline push notification which is a zero cost and not an internet-dependent service is necessary.

The primary purpose of every business is minimizing the expenses to the lowest extent to increase profitability without compromising the quality [5]. The offline push notification will

*Carlos Hilado Memorial State College – Alijis Campus, Philippines.

become an alternative messaging service to the traditional internet-dependent push notification service that may eliminate the supposed cost for the internet in operating the computerized system within the business organization. Consequently, the delivery of the message to mobile devices will become much smoother and faster since it uses Local Area Network (LAN) to channel the data [6].

The offline push notification is comprised of a pull and push method. The pull method is where the mobile client device connects to the server computer by accessing it through the server IP address and sending a request of new messages if there is any. The push method is where the server sends the new messages as a response to the request. Hence, there is no intervention of any third-party services to push the messages.

System Architecture

The system architecture of the offline push notification is shown in Figure 1 which comprises of a mobile device and a server computer both connected in a local area network. The mobile device is a smart phone and has a custom mobile application which configured to execute various functions such as connecting to the custom server application of server computer by accessing the server IP address, requesting of message data and receiving of server response over a local area network (LAN). The server computer has custom server application which is configured to received request, authenticate client mobile device request and sending message data to the client mobile device.

The mobile device and server computer have to be connected in one Local Area Network. The custom mobile application of mobile device must have a feature that allows to enter the IP address and connect to the server in case the IP address changes. Once mobile device is connected to the server over a network, it can now send request to the server of the latest message data. Consequently, the server will receive the request and authenticate it. Authentication of request is needed to verify client's request if request belongs to the recommended custom mobile application to keep the server resources secured and protected. After the request is authenticated and found out that it is a legit request, the server computer will respond to the request of getting latest message data. The response is a JSON formatted string that contains message ID and message Content. Then, the custom mobile application of mobile device will receive the JSON response from the server over a LAN. The JSON response will be parse by the custom mobile application and convert into string data. Finally, the push notification feature of the mobile device will be activated through operating system push notification service (OSPNS) with the string data as content of the push notification.

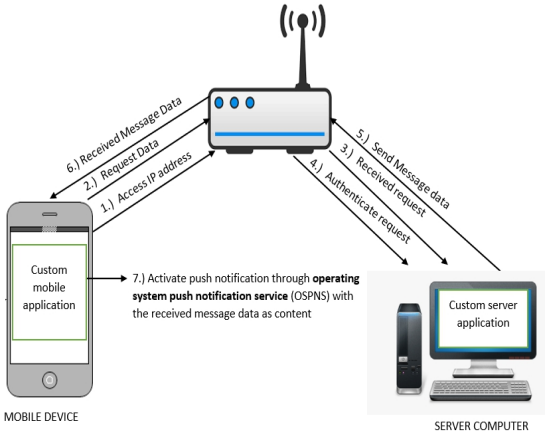

Figure 1: An Offline Push Notification Operational Framework

System Development

In developing the offline push notification, both the hardware and software must be established as described in the system architecture.

For the hardware component, the mobile device for client side, server computer for server side and router for network between the client and the server. The idea of offline push notification is applicable in any mobile device and any operating system such as Android OS, iPhone OS, BlackBerry OS, Symbian OS, etc. The server computer specification must at least handle receiving multiple request and process multiple request responses from or to smart phone. The router is needed for establishing the network between the mobile device and server computer. Router doesn't need to connect to the internet since there is no third-party internet-based service needed.

For the software component, custom mobile application for mobile device and custom server application for the server computer must be developed. The custom mobile application can be an android application, iOS application, or any mobile-based application.

In developing the mobile application following feature must be developed:

- Server access dashboard for dynamically connecting to the server application of the server by inputting the IP address and security code issued by the server application as shown in Figure 2;
- Background task execution of requesting new messages;
- Receiving of the JSON formatted data response of the server;

- Parsing JSON formatted data into string data;
- Activating push notification through operating system push notification service (OSPNS) with the string data as content of the push notification;
- Inserting the new message data in mobile device database.

Figure 2: Server Access Dashboard of Custom Mobile Application

The mobile application should be connected to the network where the server computer also connected with. Then, the mobile device user uses the server access dashboard module allowing the custom mobile application to access dynamically the server by inputting the server IP address and the security code issued by the custom server application. The security code will be an authentication token as client mobile device access the resources of the server. Once the request is authenticated and server access is successfully established, the custom mobile application will now request new messages to the server through a background task execution. Background task execution is triggered from time to time while the application is running to get new message data from server. The request will later on be responded by the server through sending new message in a JSON formatted data. The client mobile device will receive the JSON response using the custom mobile application and parse into a string data. Then, the push notification of the mobile device is activated with the string data as content. Lastly, the new message data will be inserted to the mobile device database such as SQLite. New message data must be stored in the mobile device database to avoid the requesting duplicate message to the server and also allows the user to view old messages even disconnected to the network.

In developing the custom server application following feature must be developed:

- Authentication functionality of the client request;
- Query message record from the server database and parse into a JSON formatted data;
- JSON formatted data is arranged into "message_id" index and "message_content" index as shown in Figure 3.

```
[
  {
    "message_id": "1",
    "message_content": "Sample message 1"
  },
  {
    "message_id": "2",
    "message_content": "Sample message 2"
  }
]
```

Figure 3. JSON formatted Data of Messages from Server

The custom sever application can be a web-based application or a desktop application which provide data to the connected client mobile device through JSON REST API. The server computer receives a request of new messages from a client mobile device over a Local Area Network (LAN). Then, the server will authenticate the client request if it is from a legitimate client computing device. If it turns out that it is legitimate after authentication, the server computer will query a message record from the server database and parse the queried data into a JSON formatted data. The JSON formatted data contains the new messages arranged into a "message_id" and "message_content".

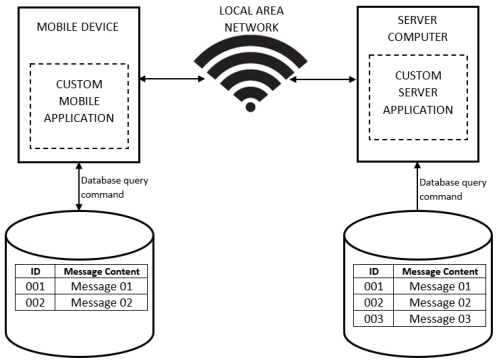

Figure 4: An Offline Push Notification Data Flow

To further illustrate the offline push notification as shown in Figure 4, the custom mobile application should be installed in a mobile device and connect to the LAN where the server computer also connects. When the mobile device successfully connects to the LAN, it can now

access the server computer through the server IP address. The user must input the IP address in the server access dashboard (Figure 2) of custom mobile application. The request access will be validated by the server computer through custom server application. Once the request is validated, the server will respond a success message to the mobile device implying that the server access is successful. The custom mobile application now can execute its background task functionality to request the new messages from the server. The background task request is executed from time to time. Then, the server receives the client request of new messages and authenticate the said request. When the server authenticates that the request is from a legitimate client, the server will now query the new messages from the server database and parse the queried data into a JSON formatted data. The JSON formatted data will be sent to the mobile device as response of the server computer to the client request. The mobile device will receive the JSON formatted data through the custom mobile application and parse it into a string data. The mobile device will then active the push notification service and use the said string as message content. When the user clicks the push notification in mobile device, it will redirect to the custom mobile application and saves the string data into the mobile device database together its "message_id" as shown in Figure 5.

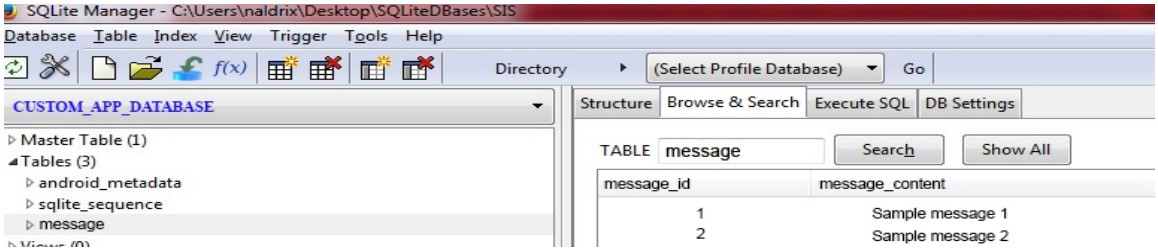

Figure 5: Mobile Database

Discussion

It is really important to build a JSON REST API on the part of the server and allow the mobile application to access it and extract the needed data. In the Figure 3, it shows us how the developer should arrange the message data from the server which will later be sent as per request by the mobile device client. The JSON data will be acquired through Background task execution code from time to time and validate if there is a unique "message_id" as it compares to the mobile database. When unavailable "message_id" is detected, push notification alert will trigger and may redirect into the application if clicked by the user and save the said unique "message_id" and "message_content" in the mobile database to avoid renotification and still be accessible when user desire to read the previous messages received by the mobile device.

The offline push notification has already tested when the researcher had to develop a visitor management system that may be able to send visit notification to the faculty and staff mobile device [9]. The researcher had to develop it offline to establish a pure offline system and not urge the Carlos Hilado Memorial State College to invest in the internet fee to operate the system, which may add burden to the school budget.

Through the developed mobile device application with the feature of offline messaging service using push notification, the server can able to communicate to the connected mobile devices by sharing server textual data. The potential data sharing of offline push notification concept from the server not only involves textual data but also images as it proves in the developed application submitted in the Google Play[9].

Conclusion

In this research, an offline push notification was developed. The offline push notification becomes an alternative to the internet-dependent push notification services that provide messaging service to the client's mobile devices. The developed prototype of the offline push notification needs no internet in operating it. Thus, the internet cost is out of the picture during the use of the offline push notification. The mobile device which has the custom mobile application sends a request to the server computer, and the server responds to the request by sending message data. The message data is then received by the mobile device and triggers the activation of the operating system push notification services. The content of the push notification is taken from the message data received from the server computer. The push notification now appears on the mobile device, and the user can read the notification message. As the user finishes reading and clicks the push notification, it redirects to the custom mobile application and displays the full body of the message. The message is also inserted into the mobile device database to become historical data and able to be read even disconnected to the

server computer. The proponent believes that the idea is very useful to those offline LAN-based systems. It can help those organizations to run their system without using the internet both to the server computer and connected mobile devices. The cost of operating the system would be much lessened since the internet is now out of the picture, and it would not cause any delays since it is a LAN-based messaging service.

References

1. Turel, O., Serenko, A., & Bontis, N., (2007). User acceptance of wireless short messaging services: Deconstructing perceived value. Information & Management, 44(1), 63-73. doi:10.1016/j.im.2006.10.005.
2. Ray, P. P., (2016). A survey of IoT cloud platforms. Future Computing and Informatics Journal, 1(1-2), 35-46, doi:10.1016/j.fcij.2017.02.001.
3. Marques, G., & Pitarma, R., (2019). An Internet of Things-Based Environmental Quality Management System to Supervise the Indoor Laboratory Conditions. Applied Sciences, 9(3), 438, doi:10.3390/app 9030438.
4. Yang, S., Chen, X., & Alty, J., (2003). Design issues and implementation of internet-based process control systems. Control Engineering Practice, 11(6), 709-720, doi:10.1016/s0967-0661(02)00181-8.
5. Lewis, H. F., & Slotnick, S. A., (2002). Multi-period job selection: Planning work loads to maximize profit. Computers & Operations Research, 29(8), 1081-1098, doi:10.1016/s0305-0548(00)00105-2.
6. Sackman, H., Erikson, W. J., & Grant, E. E., (1968). Exploratory experimental studies comparing online and offline programming performance. Communications of the ACM, 11(1), 3-11, doi:10.1145/362851.362858.
7. Qadeer, M. A., Goel, G., & Rana, A., (2014). An empirical study on computer networking and social networking for business development research scholar. Globus An International Journal of Management & IT, 6(1), 58-68.
8. Sharma, M., & Parihar, R., (2019). Information System and Its Implication in New Emerging Society. Globus An International Journal of Management & IT, 11(1).
9. Rivera, J. D., (2016, January 23). VMS Support (Version 1.0) [Computer software]. Retrieved February 6, 2021, from https://play.google.com/store/apps/details?id=com.naldrix.vmssupport.