Firebase Authentication Cloud Service for RESTful API Security on Employee Presence System

Conference Paper · December 2021

DOI: 10.1109/ISRITI54043.2021.9702776

2 authors, including:

Luthfan Hadi Pramono
Universitas Teknologi Digital Indonesia
6 PUBLICATIONS **14** CITATIONS

Firebase Authentication Cloud Service for RESTful API Security on Employee Presence System

Luthfan Hadi Pramono
dept. of Computer Engineering
Universitas Teknologi Digital Indonesia
Yogyakarta, Indonesia
luthfanhp@utdi.ac.id

Yohanes Krisna Yana Javista
dept. of Informatics
Universitas Teknologi Digital Indonesia
Yogyakarta, Indonesia
krisnayanajavista@gmail.com

***Abstract*—Authentication is essential in identifying users to access or use the system. One application of the Authentication process is the Presence System. The old Presence System at Amigo Company is prone to misuse and data manipulation, so it is necessary to develop a new Presence System based on Smartphones. Data communication between Back-end and Front-end architecture in Presence System using RESTful API. This study aims to implement security on the RESTful API by using JSON Web Token generated by Firebase Authentication Cloud Service. The results of the study indicate that the user cannot manipulate data or use the identity of another user because of Firebase Authentication security and the strict data verification process.**

Keywords—Firebase, Authentication, JSON Web Token, RESTful API, Presence System

I. INTRODUCTION

The authentication process is widely implemented in applications, especially those related to the use of personal data. In the authentication process, the system will ensure an introduction or an acknowledgment, who actually has interacted with the system. The concept of authentication, in general, includes several factors, namely something the user knows (knowledge), something the user has (ownership), and something that is in the user (biometrics) [1]. The use of a smartphone cannot be separated from the authentication process, because on a smartphone the user must subscribe to be able to use it.

Data from the International Data Corporation (IDC) notes that the Estimated market share of smartphones is 1.38 billion units in 2021, an increase of 7.7% over 2020 [2]. By 2023, mobile connectivity will be owned Over 70 percent of the global population. By 2023 The total number of global mobile subscribers will grow from 66% of the population (5.1 billion) in 2018 to 71% of the population (5.7 billion) [3]. This proves that there are more and more smartphone users every year and it always increases as if smartphones are an inherent attribute of users who are always used in daily activities.

In the case study of Amigo Company with limited investment funds, the work attendance process only uses an ordinary recording system using a form, so it is very easy to manipulate. Within the existing limitations, a project was developed as a solution for the Presence System by optimally utilizing existing resources. By looking at the potential of using smartphones more and more, then a Presence System was developed using smartphone media. This study aims to produce security on RESTful API with JSON Web Token generated by Firebase Authentication Cloud Service and used for valid data exchange on Amigo Company Presence System.

II. LITERATURE REVIEW

A. Firebase Authentication

Firebase was developed to be a cloud-based SDK service that can be used for developing custom authentication applications, incorporating identities into common online-based services: GitHub, Twitter, Facebook, and Google, or using email/password authentication. Cloud Functions in Firebase allows us to write program code that responds to an event in Firebase elements, one of which is authentication [4]. Google Firebase enables us to develop communication-based applications with relative ease and also a service that provides such a real-time database server, along with a host of other features. Google Firebase is a system that will be capable of sending text-based messages and files such as text, video, audio, and images through the internet between two users on the network in real-time. Android operating system and Google Firebase use to handle the Back-end of the communication operation, highlighting the various features of both the operating system and the service [5]. As a cloud service provider, Google introduced Firebase API with its unique characteristics. It makes android apps more efficient and speedier because it eliminates the need for PHP as a third-party language to interface with the database. It also provides a secure path for JAVA to communicate directly with the database [6].

B. RESTful (Representational State Transfer)

A REST API (also known as RESTful API) is an application programming interface (API or web API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services [7]. REST itself allows system requests to access and manipulate text represented from a Web Service (WS). The RESTful API does not have an official standard for its notation because REST is an architecture [8]. When a client request is made via a RESTful API, the RESTful API transfers the resource status information to the user or endpoint. This information, or representation, is conveyed in one of several formats over HTTP: plain text, PHP, Python, XLT, JSON (Javascript Object Notation), or HTML. [7]. PHP language and Laravel framework can be used for the development of RESTful API model construction [9]. RESTful APIs have been used in the Web Service implementation model to exchange data using JSON Web Tokens. This solution is used in sales order applications, it can simplify the sales person's performance in marketing their products [10].

C. JSON Web Token (JWT)

JWT is a token-based authentication method that encodes data in JSON string format. JWT has 3 parts separated by dot characters, namely Header, Payload, and Signature shown in

Fig. 1 [11]. The Header section specifies the token type and hash algorithm to be used for the JWT signature. The Payload section is where the actual information is stored for the token to be used. Information is a name and value pair and can store arbitrary data, such as token issuer, token expiration time, and user information. The Signature section allows token issuers to sign tokens using a hash-based message authentication code (HMAC), RSA, or ECDSA to maintain token integrity [11, 12].

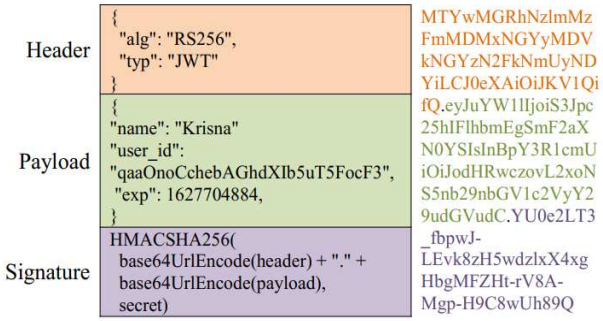

Fig. 1. JSON Web Token (JWT)

Authentication technique implemented using token-based authentication method with a JSON Web Token, which will produce a secure authentication and make web service multi-platform [13]. REST (Representational State Transfer) based architecture with JWT emerged enormously in the recent development of client-server applications. JWTs (JSON Web Token) are used for authentication of subsequent client requests without making frequent calls to the resource server or database [14].

D. International Mobile Equipment Identity (IMEI) & Android Device ID

International Mobile Equipment Identity, typically found behind the battery, is a unique number given to every single mobile phone. (EIR - Equipment Identity Register) containing all valid mobile phone equipment is a database of IMEI numbers of cellular phones connected to a GSM network [15]. The use of IMEI is also used to validate mobile phone owners so that mobile phones cannot be used by other Users [16]. User authentication methods using IMEI and JWT in the smart home system solved the problem of unauthorized smart home device registration of hackers by the application of IMEI and JWT technology [12].

The Android Device ID is a one-of-a-kind alphanumeric code generated the first time we setup our Android phone. This code essentially identifies the device in the same way that the IMEI number does. However, instead of tracking device, Android Device ID is specifically used for identification [17]. Deterministic attribution employs a Device ID to identify the same user across multiple channels and interactions, accurately measuring their user behavior. Because of it's deterministic method of measurement, matching the Device ID to user interactions is one of the most reliable and accurate methods of attribution. Android Device ID can found using a dial pad code or a third-party app [18].

III. PROPOSED SOLUTION

The Presence System is separated into two parts. The first is the Authentication Process section and the second is the Presence Process section. The implementation of the proposed solution is in the Authentication Process section. The detail description of the two parts are explained as follows:

A. Authentication Process

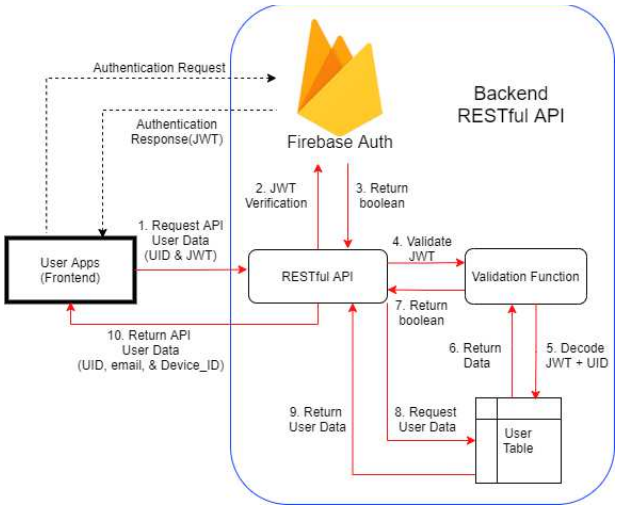

Fig. 2. The system architecture of the Authentication Process

The system architecture in Fig. 2 shows the flow of data exchange in a presence application with Firebase Authentication assuming that the User has logged in and made an Authentication Request to Firebase. Firebase then provides an Authentication response in the form of a JWT and is sent back to the mobile application to be used as a data transaction on the WS. By using a RESTful API based on Firebase authentication security, the stages of the user authentication process are as follows:

1) Request API User Data: The application performs data transactions on the WS to retrieve user data using JWT.

2) JWT Verification with Public Key: The JWT is sent to Firebase Authentication to check whether the token sent is valid or not before being sent back to the WS RESTful API.

3) Return Boolean (JWT Verification Success/Fail): Tokens that have been checked and succeeded will be returned to the Web Service to carry out the next stage, namely the validation process. If the checked token fails then the process will not continue and return to the login page.

4) Validate JWT: The validation function will validate the data and then decode the JWT.

5) Decode JWT + UID: The decoded JWT will be taken by the UID, in this way data leakage can be avoided with the Insecure Direct Object Reference (IDOR) method because the data from the decoded token will be checked for its UID in the User table whether it is available or not.

6) Return Data: If the data in the validation function matches the data in the database, it will return the data.

7) Return Boolean: The validation function will return true if the data in the validation function matches the data in the user database.

8) Request Data: Next, the API will request data from the database.

9) Return User Data. Database returns User data to API.

10) Return API Data User: This stage is the end of an API request, where the User Application receives the requested data containing the UID, email, and Device_ID then generates a QR Code to be processed at the Presence stage.

The login process steps are shown in Fig. 3 where there is a User as an actor, and Login_page, User_Check, User_Data, and Homepage as objects. The process begins with the User filling in the login form on the login screen, then User_Check is done by sending login data in the form of Username and Password. The user data is then validated, if the data is valid it will immediately display the Homepage, if it is not valid it will be returned to the User in the form of an Unauthorized device error message.

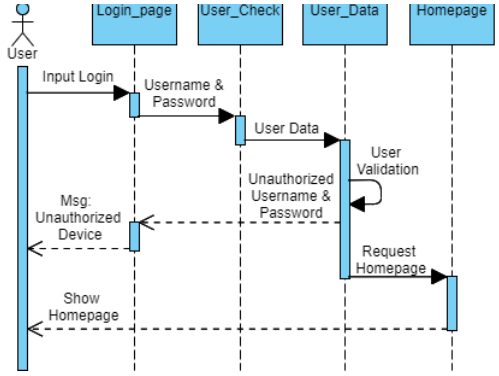

Fig. 3. Authentication Process sequence diagram

B. Presence Process

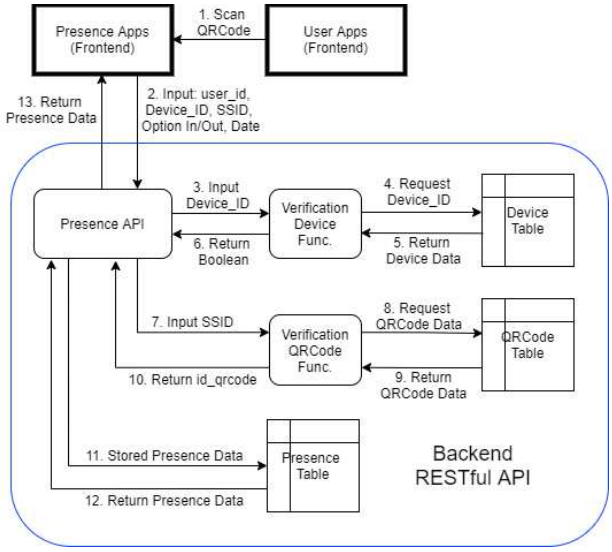

Fig. 4. The system architecture in Presence Process

Fig. 4 shows the system architecture in the Presence Process where there are two stages of verification and the final stage.

1) Device Verification: Verification of the QR Code on the User Application will be scanned through the QR Code scan function (Presence Apps). From the results of the scan, the QR Code is converted to text and sent to the Presence API, then enters the first verification stage, where the Presence API sends Device_ID to the device verification function, then Device_ID is checked whether it is registered or not in the device table and matches or not with the Device_ID of the device Employee. If the Device_ID matches the data in the device table, it will return true on the Presence API.

2) QR Code verification: The next verification where the QR Code data sent from the employee application will be checked. The Presence API sends the SSID data to the QR Code verification function and then checks the QR Code table. The results of the check will be sent back to the QR Code Verification function and will return the id_qrcode to the Presence API.

3) Final stage: The final stage of the Presence process is the Presence API store incoming attendance data from the user to the user table, then send a presence notification to the application indicating the presence is successful.

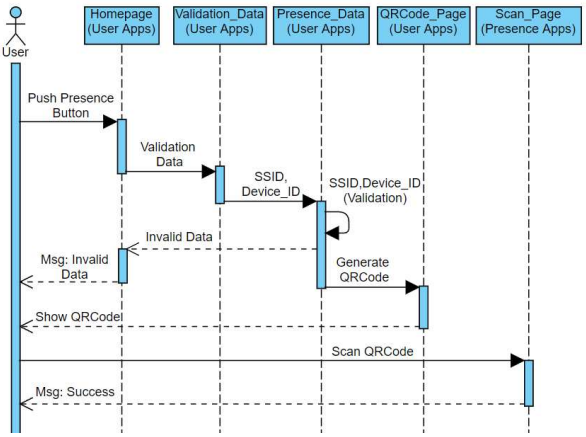

Fig. 5. Presence Process sequence diagram

In the presence process in Fig. 5, there are users as actors then Homepage, Validation_Data, Presence_Data, QRCode_Page, and Scan_Pages as objects. The user uses the presence button on the Homepage, then a QR code will be generated as well as checking location data, Device_ID Number, and user network. The presence application runs locally in the office network using the SSID (abimanyu-club) so that only user applications connected to the local office network can perform attendance. The user application will send the SSID and it will be matched against the SSID in the database. If when the data is invalid, the system will return the data to the user in the form of a message that the data is invalid, if the data obtained is valid it will display a QR Code. The next step is to scan the QR Code to make attendance, and the user will get a message from the presence.

IV. RESULT AND DISCUSSION

Evaluation and testing perform using Postman for API lifecycle and operation testing. The task is carried out in two stages, which are Authentication Process evaluation and Presence Process evaluation that will describe as follows:

A. Authentication Process evaluation

It is assumed that the user has signed up before so that they have access to login to the system using a google account. The user will get a UID and the Device_ID will be recorded in the database after the signup process. The app will send the registered user UID data to Firebase Authentication. The UID data received by Firebase Authentication will be checked and if the data is available it will return the JWT to the application and stored it in the application's local storage. The user will authenticate to the Firebase Authentication Cloud Service when the application is run, then the RESTful API will verify the JWT of the User apps with the JWT in the Firebase Authentication Cloud Service. Fig. 6 is a sample of a JWT print from Firebase that is owned by user Krisna. The JWT will be the key to be used to access all REST APIs later. JWT has a lifetime of 1 hour, so if it is more than 1 hour the token will refresh again.

Fig. 6. JWT results from Firebase

TABLE I. THE RESULTS OF THE IDENTIFICATION OF UID & JWT FROM THE LOGIN PROCESS OF 5 USERS

User	UID	50 last char of JWT
Krisna	qaaOnoCchebAGhGdXIb5uT5FocF3	vjw0s8m6lPxoaGeemJccJ61tzhYQxupvWyxHDe1lMI6siP9eEg
Aura	THufYiyoyST31QMwzf4AHCC2e0j1	7waJFdcw_gsSL2J2E5M4BbVcTCPMZShFdmM5_Ej3ziaHQ0NX_A
Indri	ivuE9rbP0aaApCuxXLNClLnIX0y1	O243bhIYuerK_L71DuwWC9gHzK2M8fBX0l_GHYp16-osEH-TSw
Wijati	W4GBJD1D3eeTTynwMkqMhJEtO113	4K7FkM9Jm_NOrju_apyXEw-wRdThZQ5W7LcOMTuZDTObbGfsLg
Yahya	NdgeXtc5nJNhI97EmBaou2oU3MQ2	meuZndSnOGV-QiXwXhm8a2YG0Lv955CSnRQgYZGwi3pfoNPV4g

TABLE II. THE RESULTS OF THE IDENTIFICATION OF DEVICE_ID FROM THE AUTHENTICATION PROCESS OF 5 USERS.

UID	Device_ID
qaaOnoCchebAGhGdXIb5uT5FocF3	20b5343bc87fb4d0
THufYiyoyST31QMwzf4AHCC2e0j1	c549e37cb3392218
ivuE9rbP0aaApCuxXLNClLnIX0y1	ac05116b0381d15a
W4GBJD1D3eeTTynwMkqMhJEtO113	155fe6fddfef43e0
NdgeXtc5nJNhI97EmBaou2oU3MQ2	12d1daa89058hf3

The UID will always be bound to the JWT. With the IDOR method, the user will not be able to exchange the JWT with the UID of another user and will display an Unauthorized message, this is because the UID of the token does not match the UID of the employee data sent. Table 1 shows the results of each user's login with JWT. As a note, because the character encoding in the JWT is very long, only the last 50 characters of the JWT will be displayed. Table 2 shows the Authentication results for each user with Device_ID. For privacy purposes, email data is not included.

In the Authentication Process described previously in Fig. 2. The performance of the process written in Table 3 presents information of Time (ms) which indicates the time required to perform the Authentication Process in milliseconds and DT (KB) which is the Data Transfer that occurs in Authentication Process in KiloBytes. It was recorded from the Authentication Process trial that had been carried out with 5 users, the average time was 116.67 ms with an average DT of 1.39 KB.

TABLE III. PERFORMANCE OF AUTHENTICATION PROCESS

Device_ID	Time (ms)	DT (KB)
20b5343bc87fb4d0	149	1.44
c549e37cb3392218	111	1.38
ac05116b0381d15a	194	1.38
155fe6fddfef43e0	77	1.37
12d1daa89058hf3	79	1.37

Fig. 7 is a display of the response received from User Krisna data by Firebase Authentication where the User data request was successfully obtained because the JWT sent from the front-end is received and the back-end will process the front-end request, namely the User data request and become the User JSON.

Fig. 7. The response is successful (accepted)

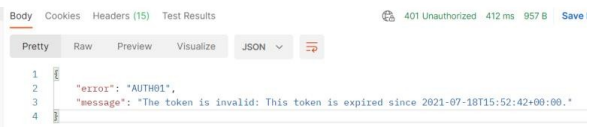

Fig. 8. Response is rejected

Fig. 8 is the display error “401 Unauthorized” of the rejected response of user data from Firebase Authentication which contains Token information if it is “invalid” because it is “expired”. The front-end will send a token when accessing one of the REST APIs on the back-end protected by Firebase Authentication, if when checked the token is “invalid” because it has “expired” then the process will be rejected by the back-end and display the message "Token is invalid" then

the front-end will not be allowed to access the data on the database.

The data manipulation test is carried out by combining the UID and token pairs randomly taken from the valid UID and token pairs from Table 1. The results of the data manipulation test are shown in Table 4. Fig. 9 shows the display of the data manipulation test if user tries to authenticate using another user's UID. Request test of user data using the IDOR method which accesses user data with a valid token but uses the UID of another user. The resulting response is an "401 Unauthorized" message, this is because the UID of the token does not match the UID of the user data.

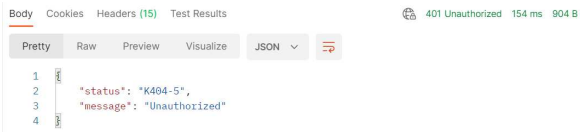

Fig. 9. Unauthorized data

TABLE IV. THE DATA MANIPULATION TEST RESULT

UID	50 last char of JWT	Result
qaaOnoCchebAGhGdXIb5uT5FocF3	meuZndSnOGV-QiXwXhm8a2YG0Lv955CSnRQgYZGwi3pfoNPV4g	401 Unauthorized
THufYiyoyST31QMwzf4AHCC2e0j1	vjw0s8m6lPxoaGeemJccJ61tzhYQxupvWyxHDe1lMI6siP9eEg	401 Unauthorized
ivuE9rbP0aaApCuxXLNClLnIX0y1	7waJFdcw_gsSL2J2E5M4BbVcTCPMZShFdmM5_Ej3ziaHQ0NX_A	401 Unauthorized
W4GBJD1D3eeTTynwMkqMhJEtO113	O243bhIYuerK_L71DuwWC9gHzK2M8fBX0l_GHYp16-osEH-TSw	401 Unauthorized
NdgeXtc5nJNhI97EmBaou2oU3MQ2	4K7FkM9Jm_NOrju_apyXEw-wRdThZQ5W7LcOMTuZDTObbGfsLg	401 Unauthorized

B. *Presence Process evaluation*

In the presence process, 5 users are tested. Table 5 is the information of users who make attendance. In the attendance process, the Presence application will convert the QR Code data (Fig. 10, user sample: Krisna) from the user application to a URL and used it for attendance requests. This Presence application is located in the local office area, therefore only user applications connected to the local network (SSID: abimanyu-club) in the office can perform attendance.

TABLE V. USER PRESENCE

Device_ID	Incoming attendance (date)	Outgoing attendance (date)
20b5343bc87fb4d0	04/08/2021 08:01:00	04/08/2021 15:08:23
c549e37cb3392218	04/08/2021 08:04:12	04/08/2021 15:02:18
ac05116b0381d15a	04/08/2021 08:06:00	04/08/2021 15:04:27
155fe6fddfef43e0	04/08/2021 08:08:13	04/08/2021 15:09:38
12d1daa89058hf3	04/08/2021 08:00:04	04/08/2021 15:00:25

Fig. 10. (left) QR Code for incoming attendance, (right) QR Code for outgoing attendance

TABLE VI. USER PRESENCE PERFORMANCE

User	Performance			
	Incoming attendance (date)		Outgoing attendance (date)	
	Time (ms)	DT (KB)	Time (ms)	DT (KB)
Krisna	99	1.014	434	1.04
Aura	85	1.02	165	1.05
Indri	144	1	104	1.05
Wijati	119	1.018	105	1.05
Yahya	107	1	127	1.04

Attendance performance can be seen in Table 6, where Time (ms) shows the time required to perform the attendance process in milliseconds while DT (KB) is the Data Transfer that occurs in the attendance process in KiloBytes. Performance attendance shows a very fast time. From the trial of the attendance process with data from 5 users, the average time of incoming attendance was 110.8 ms with an average data transfer is 1.006 KB. The average time of outgoing attendance is 112 ms with an average data transfer of 1.05 KB.

Fig. 11 & 12 are result of response data from the user Krisna. Fig. 11 shows the results of the JSON response data "201 Created" when the user performs Incoming attendance, while Fig. 12 is the result of the JSON response data "200 Ok" when the user performs Outgoing attendance. In the resulting JSON response, the attribute indicating Incoming attendance (date) is written with *jam_masuk*, while the attribute indicating Outgoing attendance (date) is written as *jam_keluar*.

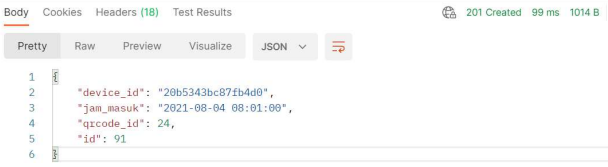

Fig. 11. Response JSON data of incoming attendance

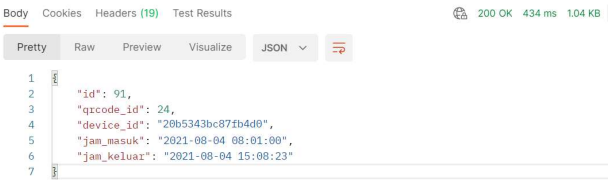

Fig. 12. Response JSON data of outgoing attendance

Several test simulations for user fraud were carried out to ensure the system could work safely and the data could not be manipulated. If the user makes a presence with an Device_ID that is not his/her registered Device_ID, an error

message “404 Not Found” will appear in the form of Device Not Found as shown in Fig. 13.

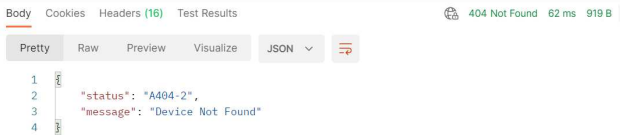

Fig. 13. Device not found

To control presence errors made by the user if the user has made an attendance before, then the system creates error “409 Conflict” and handled with a Duplicate Entry message notification. The error message is shown in Fig. 14.

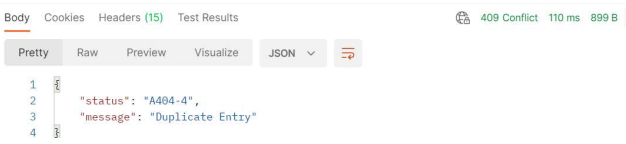

Fig. 14. Duplicate entry

QR Code has a lifetime of 5 minutes, this is one of the considerations of the security factor so that the data cannot be misused. If it has expired the system will generate error code “401 Unauthorized”, the notification message displayed is “QR Code expired” as shown in Fig. 15. The system will generate a new QR Code that can be used by user.

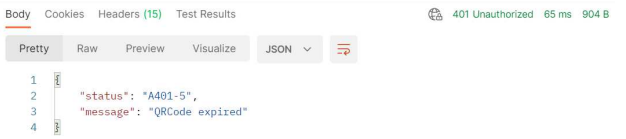

Fig. 15. QR Code expired

V. Conclusion

After carrying out the authentication test process, it was concluded that Firebase authentication can be used on RESTful APIs as security authentication and data communication protection, and also can avoid user abuse in the authentication process on Presence System. With various test simulations, it can be concluded that each user on Presence System can only use his own device and the user cannot manipulate data using other identities or device.

The performance of the RESTful API is very fast when the front-end makes a request because the used of SDK from Firebase and does not require long code to check, and the data is generated in JSON format which does not require a large response DT size.

The proposed system will be tested continuously and will be developed to ensure security and data manipulation by users in the Authentication Process and Presence Process. For the development of further research projects, multi-factor authentication will be tested by adding biometric authentication to the presence application.

Acknowledgment

The authors would like to thank Amigo Company for providing data and infrastructure accommodation to support this research.

References

[1] Ometov A, Bezzateev S, Mäkitalo N, Andreev S, Mikkonen T, Koucheryavy Y. Multi-Factor Authentication: A Survey. *Cryptography*. 2018; 2(1):1. https://doi.org/10.3390/cryptography2010001.

[2] IDC. “2021 Smartphone Growth to Reach Its Highest Level Since 2015, According to IDC”. May 2021. available online: https://www.idc.com/getdoc.jsp?containerId=prUS47770921 (accessed on 4 June 2021).

[3] Cisco. “Cisco Annual Internet Report (2018–2023)”. *White Paper, 2020*. 2020. available online: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf (accessed on 7 July 2021).

[4] Moroney L. Using Authentication in Firebase. In: *The Definitive Guide to Firebase*. Apress, Berkeley, CA. 2017. https://doi.org/10.1007/978-1-4842-2943-9_2.

[5] N. Chatterjee, S. Chakraborty, A. Decosta, and A. Nath, “Real-time Communication Application Based on Android Using Google Firebase,” *Int. J. Adv. Res. Comput. Sci. Manag. Stud.*, vol. 6, no. 4, pp. 74–79, 2018.

[6] Chunnu Khawas and Pritam Shah, "Application of Firebase in Android App Development-A Study", *International Journal of Computer Applications*, vol. 179, pp. 49-53, 2018.

[7] Redhat, “What is a REST API?, ” 2020, available online: https://www.redhat.com/en/topics/api/what-is-a-rest-api (accessed on 20 July 2021).

[8] F. Doglio, *Pro REST API Development with Node.js*. La Paz, Canelones: Apress, 2015.

[9] X. Chen, Z. Ji, Y. Fan, and Y. Zhan, “Restful {API} Architecture Based on Laravel Framework,” *J. Phys. Conf. Ser.*, vol. 910, p. 12016, Oct. 2017, doi: 10.1088/1742-6596/910/1/012016.

[10] E. Edy, F. Ferdiansyah, W. Pramusinto, and S. Waluyo, “Pengamanan Restful API menggunakan JWT untuk Aplikasi Sales Order”, *RESTI*, vol. 3, no. 2, pp. 106 - 112, Aug. 2019.

[11] Sebastián E. Peyrott, “JSON Web Tokens in Detail,” in *JWT Handbook*, Auth0 Inc, 2018.

[12] Hong N, Kim M, Jun M-S, Kang J. A Study on a JWT-Based User Authentication and API Assessment Scheme Using IMEI in a Smart Home Environment. *Sustainability*. 2017; 9(7):1099. https://doi.org/10.3390/su9071099.

[13] M. Haekal and Eliyani, "Token-based authentication using JSON Web Token on SIKASIR RESTful Web Service," *2016 International Conference on Informatics and Computing (ICIC)*, 2016, pp. 175-179, doi: 10.1109/IAC.2016.7905711.

[14] S. Ahmed and Q. Mahmood, "An authentication based scheme for applications using JSON web token," *2019 22nd International Multitopic Conference (INMIC)*, 2019, pp. 1-6, doi: 10.1109/INMIC48123.2019.9022766.

[15] Samsung, “What is (IMEI) International Mobile Equipment Identity?,” Sept. 2020. available online: https://www.samsung.com/ph/support/mobile-devices/what-is-international-mobile-equipment-identity-or-imei/ (accessed on 29 July 2021).

[16] N. Hermanto, Nurfaizah, W. M. Baihaqi and Sarmini, "Implementation of QR Code and IMEI on Android and Web-Based Student Presence Systems," *2018 3rd International Conference on Information Technology, Information System and Electrical Engineering (ICITISEE)*, 2018, pp. 276-280, doi: 10.1109/ICITISEE.2018.8721009.

[17] Henny Leidiyana, & Yusuf, I. “Aplikasi Kehadiran Karyawan Berbasis Android Menggunakan QR Code Scanning dan Location Based Service,” *Journal of Informatic and Information Security*, 2021, 2(1). https://doi.org/10.31599/jiforty.v2i1.569.

[18] Anonymous, “Device ID,” available online: https://www.appsflyer.com/glossary/device-id/ (accessed on 30 July 2021).