

IMPLEMENTATION OF MICROSERVICES IN FINTECH APPLICATIONS USING EXPRESSJS FRAMEWORK

Bernardus Redika Westama Putra and Evangs Mailoa

Satya Wacana Christian University

E-mail: redikawest@gmail.com, evangs.mailoa@uksw.edu

ARTICLE INFO

Received:
June, 28th 2021
Revised:
July, 9th 2021
Approved:
July, 14th 2021

ABSTRACT

Along with the development of software architecture, the emergence of microservice architecture has become a trend used by developers in recent years. This study aims to analyze the implementation of microservices in Fintech applications using the ExpressJs framework. The method used in this research is descriptive qualitative method. This study uses a library research method that refers to resources available both online and offline such as: scientific journals, books and news sourced from trusted sources. The method used in making the application is DevOps. In this method, there are eight stages of software development, plan, code, build, test, release, deploy, operate, monitor. Based on the discussion and testing that has been carried out, several conclusions can be drawn namely microservice makes application development easier and more flexible by adding services as needed. Make it easier for other applications to access services that have been made via HTTP Requests, another thing is to facilitate application integration between platforms, both mobile, web and desktop.

KEYWORDS

Microservice, Devops, Applications



This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International

INTRODUCTION

In this era, almost all activities that humans do are always dependent on technology. The effect of the development of information and communication technology is to make various fields experience changes, such as in the economic, social, and other fields which

How to cite:

E-ISSN:

Published by:

Bernardus Redika Westama Putra and Evangs Mailoa. (2021).
Implementation Of Microservices In Fintech Applications Using Expressjs
Framework. Journal Eduvest. 1(7): 559-567
2775-3727
<https://greenvest.co.id/>

take place so quickly (Ngafifi, 2014). In this technological era, the financial sector is also developing towards a more practical and modern direction by utilizing technology (Rahadiyan & Sari, 2019). Currently, it is very important to provide technological innovation and its implementation in the economic field (Supriyanto & Ismawati, 2019). Business people are starting to change the concept of business to digital by utilizing technology. All these changes are part of business competition. Technological development is a challenge that must be turned into an opportunity because it provides benefits in several aspects (Darman, 2019).

With the rapid development of technology, various applications in the financial sector have emerged that combine financial systems and technology called Financial Technology. FinTech is an application of digital technology that is used to provide solutions to public financial problems. Currently, Fintech has various functions and can develop more easily. Currently, Fintech has been able to serve many services such as e-money, loans, fundraising, installments, and others (Muhamad Rizal, Erna Maulina, 2018). From the digitization of the business, there are now many websites and mobile-based applications that facilitate transactions and payments without having to be limited by space and time as long as the smartphones and computers used are connected to the internet. The bigger the business that is run, the bigger the application that will be made.

Rest API (Representational State Transfer) is a style of software architecture in which there are rules for creating services (Afifah & Setiaji, 2019). The HTTP protocol is always used for data exchange and data sent or received in the form of JSON (Rajagukguk, 2018). This method is often applied in the development of an application, including the microservice-based application that we are currently developing. By using rest there are many advantages obtained, the system becomes easier to develop in the exchange and communication of data for better and faster performance.

A web service is a software system created to support interaction between two different applications over a network. The network commonly used in web services is HTTP. Web services are also applications that are exposed or published so that clients can access or use data provided by web services (Perwira & Santosa, 2017). Web services also provide us with services to interact between web services. The URL or endpoint on the web service contains the required information and contains commands such as Get/Post and others. Web services allow clients to exchange data regardless of the type of database, the framework used. These factors make web services become widely used today.

Along with the development of software architecture, the emergence of microservice architecture has become a trend used by developers in recent years. Microservice is an architectural style that separates large systems into small functional units to provide better modularity (Karabey Aksakalli, Çelik, Can, & Tekinerdoğan, 2021). Microservices provide flexibility for developers to develop software in a fast time. The development of microservice architecture is also supported by the lack of reliable monolithic architecture in dealing with system failures because in one application having the same codebase, it is certain that monolithic architecture is also a single point of failure that will occur if one service malfunctions or an error occurs.

Microservice is a collection of independent and small services that have different functions and services (Rahmanda, 2018). These services can interact with each other to create a complex application through an API. This microservice consists of many small, separate services that focus on their respective tasks or functions in system development (Putra, 2018).

Currently, many Fintech companies are starting to develop new businesses into the world of technology, and in the future, there will be more and more services that can be used. Developers must use the right method so that application development becomes

easier and more flexible and can be integrated into many platforms. The purpose of this research is to produce a system design that can be easily integrated into many platforms and make the development process easier and more flexible.

The research that has been done previously by (Ekasmara & Santoso, 2020) showed that Microservice with JSON data exchange format provides ease of implementation to the frontend. While ExpressJs itself is a framework owned by NodeJs which is deliberately designed to be flexible and simple to help the backend application development stage. ExpressJs is very different from other frameworks such as Laravel where this library is given the freedom for developers to design applications in such a way that it is possible for application developers to design software that is being built (Bachri, Priyambadha, & Rusdianto, 2018). In contrast to previous research, this study aims to analyze the implementation of microservices in Fintech applications using the ExpressJs framework.

RESEARCH METHODS

The method used in this study is a qualitative descriptive method. This study uses library research methods that refer to sources available both online and offline such as: scientific journals, books and news sourced from trusted sources. While the method used in making the application is DevOps. In this method, there are eight stages of software development, namely:

a. Plan

At this stage, planning for the manufacture and development of software is carried out. After the planning is mutually agreed upon, then proceed to the code stage.

b. Code

At this stage, the developer begins to create or develop software according to the agreement in the previous stage.

c. Build

At this stage, the application that has been developed or made by the developer will use the development server so that it can be used or tested by the tester.

d. Test

At this stage, the software system will be tested by Quality Assurance to ensure that the system is running well and look for bugs in the system to find out the main problems by the developer. The software system will always be tested if there are changes.

e. Release

At this stage, a system launch will be carried out with a focus on aspects of change management, launch approval, and automation of system launch.

f. Deploy

After going through the testing process and getting the agreed results without finding any bugs, the software system is built or deployed to the production server.

g. Operate

At this stage, infrastructure installation will be carried out, infrastructure scalability changes, infrastructure management and scalability will be carried out.

h. Monitor

At this stage, the software system that has been used by the user will carry out monitoring to determine the performance or performance of the system.

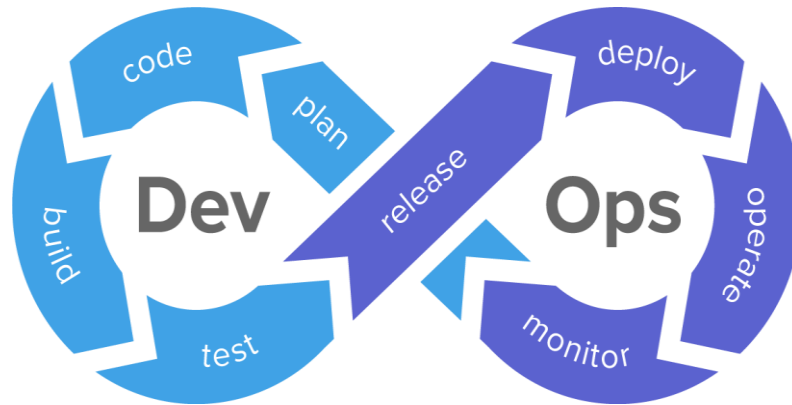


Figure 1 Devops Method

RESULTS AND DISCUSSION

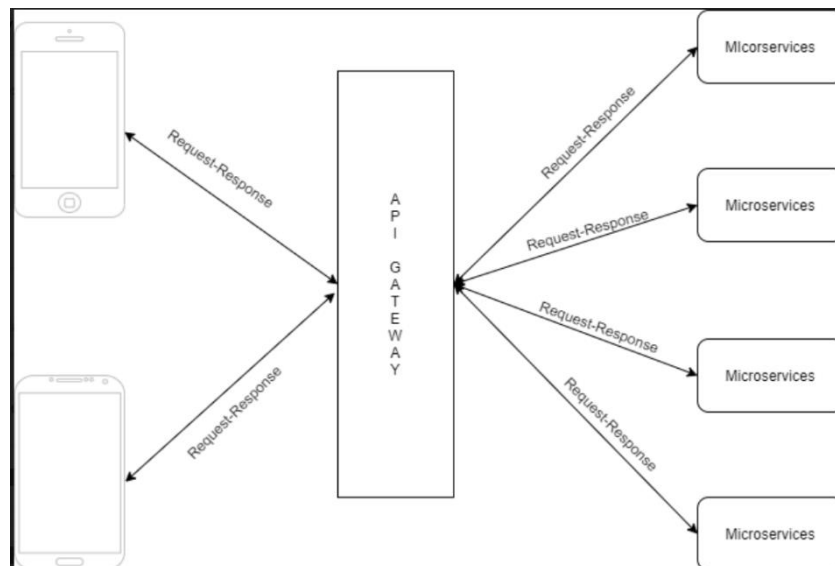


Figure 2. API Gateway Architecture

Based on Figure 2 in the depiction of the API gateway architecture, all client requests are directed and then forwarded to the microservice according to the specified endpoint. After the request from the client is in accordance with the provisions, the microservice will send a response that will be directed to the API gateway again, then forwarded to the client. Each microservice will have its own endpoint according to its purpose, so the API Gateway will load all endpoints that point to all existing microservices.

The advantage of using a microservice architecture is the ease of integration on various application platforms more easily such as website, mobile and desktop applications. With the use of API, the process of exchanging data becomes easier and faster. From the client side, this is done by sending a request body and header as needed via an HTTP request, then the server will respond via an HTTP response. The client creates a UI and then integrates it with the API Gateway services to get a response from the microservices.

If the application has a need for adding new features, then it is necessary to add new microservices, so application development becomes easier because there is no need to completely overhaul the application. In addition to adding services, by using the microservices architecture, the system will become easier to accept new clients.

Microservices should not be accessed directly by the public, so as not to cause data leakage. Therefore, an authentication is required for each microservice. Authentication on microservices needs to be done to secure the services themselves from crimes on the internet. To overcome and avoid the above problems, an API gateway application was made.

The function of the API Gateway is to connect the services that have been created with the client. API Gateway can add several other functions such as logging, authentication, rate limiting, caching, transforming and load balancing. So that clients can access the API Gateway application service on the internet indirectly, and only clients who have been given access can access the API Gateway.

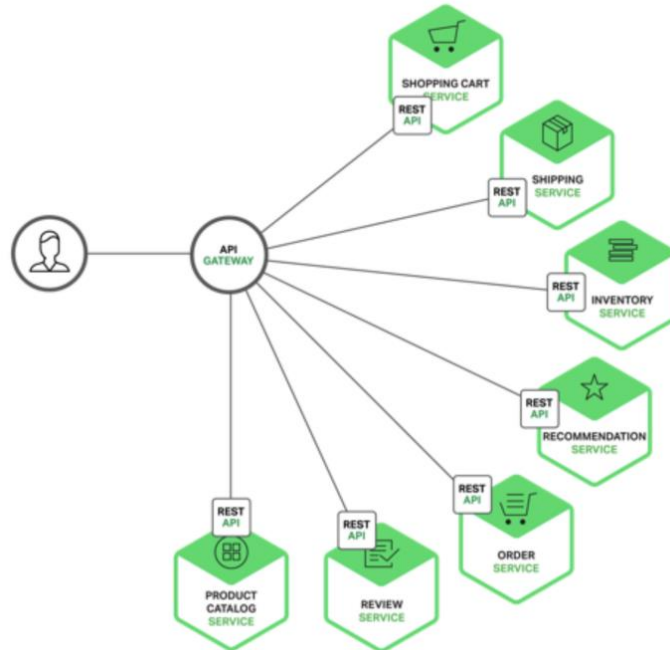


Figure 3 API Gateway Illustratuion

Microservices applications can be accessed via API Gateway. so every client must fire API Gateway first to be able to access Microservices. Because the API Gateway can only be accessed by the public, so it needs a security on the API Gateway. For this reason, API Gateway added Authentication or Authorization features. The author uses Json Web Token or better known as JWT which is a random token in the form of a string with functions to authenticate the system and exchange information. The token contains the information or data required by the user. JWT tokens are usually sent using a url and using

the post method or in the header (Rahmatulloh, Sulastri, & Nugroho, 2018) (Edy, Ferdiansyah, Pramusinto, & Waluyo, 2019).

```
if (!login) {
  res.json({
    message: "user not found",
  });
} else {
  console.log(login);
  if (login.password == req.body.password) {
    jwt.sign({ payload }, "secretkey", { expiresIn: req.body.tokenexp }, (err, token) => {
      res.json({
        token: token,
      });
    });
  } else {
    res.json({ message: "Invalid username or password" });
  }
}
```

Figure 4 Examples of Functions in the API

In the application that is made, the JWT formula consists of the payload or request body, "secretkey", and expressIn. The "Secretkey" in the formula can be changed as needed, as well as the expressin or expiration time of the token. The gateway also requires a username and password, so every client who wants to access the gateway needs to first register a username and password in the database. Otherwise, the client cannot get the token to access the gateway. After the username and password are registered, they can access the gateway.

Testing using the Postman application. Request body contains username, password, and tokenexp in JSON format. The request is sent to the server, then the server will give two types of response, namely a successful response and a failed response. The response is successful if the username and password are correct and registered in the database. Response fails if the username has not been registered in the database. The response received also has a JSON format, each username has a different token. Tokens that have been successfully generated also have a time limit. If the token has passed the time limit, then the token cannot be used again and must be generated again.

```
{
  "username" : "testing",
  "password" : "testing",
  "tokenexp" : "60s"
}
```

Figure 5 Data in Json

```
"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9,\neyJwYXlsb2FkIjp7InVzZXJuYV11IjoiaGVzdG1uZyIsIn8hczN3b3kiIjoiaGVzdG1uZyJ9LCJpYXQ1OjE2MjQxNTMwODIsImV4cCI6MTYyNDI1MzE0Mn0.\nYRo1CsE0X0o96d2go_LRxRHZgm1Hn2gsrLcIbpt-tU"
```

Figure 6 Tokenization

The next experiment, testing the endpoint that has been created using the token that was generated earlier. As a result, the server will provide a response containing the required data.

Example: endpoint account statement

```
{
  "statusCode": "000",
  "message": "Operation success",
  "data": [
    {
      "id": 4442,
      "type": "Withdrawal",
      "transactionDate": "2021-5-19",
      "amount": 303549,
      "note": "25511782201621399647-MNCPLAY"
    },
    {
      "id": 4440,
      "type": "Withdrawal",
      "transactionDate": "2021-5-19",
      "amount": 142350,
      "note": "25511782201621399647-MNCPLAY"
    }
  ]
}
```

Figure 7 JWT Token Implementation

The last experiment is implementation to several platforms, namely website and mobile. Making a website using the ReactJs framework and mobile using Flutter, in Figures 1.8 and 1.9 it can be seen that the microservice that has been built can run well.

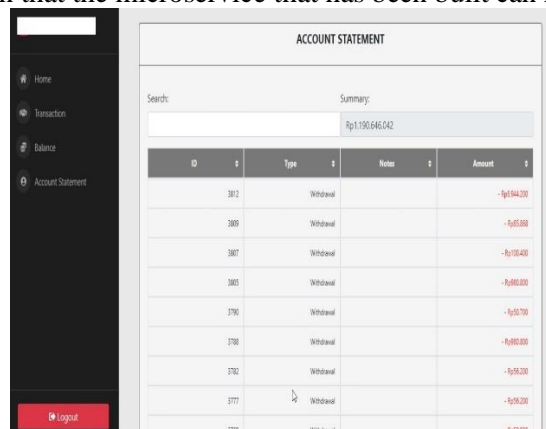


Figure 8 Implementation on the web

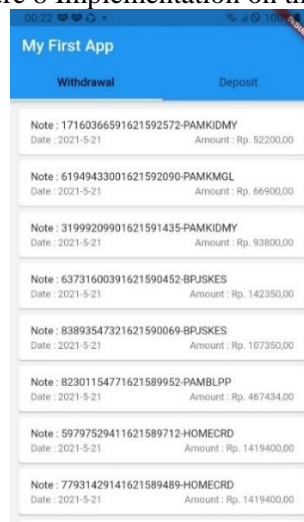


Figure 9 Implementation on mobile

CONCLUSION

Based on the discussion and testing that has been carried out, several conclusions can be drawn namely Microservice makes application development easier and more flexible by adding services as needed. Make it easier for other applications to access services that have been made via HTTP Requests, another thing is to facilitate application integration between platforms, both mobile, web and desktop.

Json web token on api gateway creates application becomes more secure because the application cannot be accessed if not using token. Framework ExpressJs is proven to be able to well build microservices and supports restful api development.

The microservice application system that has been created still has shortcomings that must be improved. The author's suggestion for further researchers is that the authentication system needs to add certain encryption or other methods to further secure important data in the database.

REFERENCES

- Afifah, Khairina, & Setiaji, Hari. (2019). *Pengembangan Rest Api Sebagai Teknologi Interoperabilitas Pada Aplikasi Uii Training Center*.
- Bachri, Hendro Febrian, Priyambadha, Bayu, & Rusdianto, Denny Sagita. (2018). Pengembangan Aplikasi Manajemen Event Berbasis Web (Studi Kasus: Fakultas Ilmu Administrasi Universitas Brawijaya Malang). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer E-Issn*, 2548, 964x.
- Darman. (2019). Financial Technology (Fintech): Karakteristik Dan Kualitas Pinjaman Pada Peer To Peer Lending Di Indonesia. *Jurnal Manajemen Teknologi*, 18(2), 130–137. <https://doi.org/10.12695/Jmt.2019.18.2.4>
- Edy, Edy, Ferdiansyah, Ferdiansyah, Pramusinto, Wahyu, & Waluyo, Sejati. (2019). Pengamanan Restful Api Menggunakan Jwt Untuk Aplikasi Sales Order. *Jurnal Resti (Rekayasa Sistem Dan Teknologi Informasi)*, 3(2), 106–112. <https://doi.org/10.29207/Resti.V3i2.860>
- Ekasmara, Alif Sani, & Santoso, Nurudin. (2020). *Pengembangan Web Portal Landing Page E-Commerce Dengan Pola Single Page Application*. 4(8), 2713–2721.
- Karabey Aksakalli, I., Çelik, T., Can, A. B., & TekiNerdoğan, B. (2021). Deployment And Communication Patterns In Microservice Architectures: A Systematic Literature Review. *Journal Of Systems And Software*, 180. <https://doi.org/10.1016/J.Jss.2021.111014>
- Muhamad Rizal, Erna Maulina, Nenden Kostini. (2018). *Fintech As One Of The Financing Solutions For Smes*. 3(61), 89–100.
- Ngafifi, Muhamad. (2014). Kemajuan Teknologi Dan Pola Hidup Manusia Dalam Perspektif Sosial Budaya. *Jurnal Pembangunan Pendidikan: Fondasi Dan Aplikasi*, 2(1).
- Perwira, Rifki, & Santosa, Budi. (2017). Implementasi Web Service Pada Integrasi Data Akademik Dengan Replika Pangkalan Data Dikti. *Telematika*, 14(1), 1–11. <https://doi.org/10.31315/Telematika.V14i01.1962>
- Putra, Rahmad Ade. (2018). Analisa Implementasi Arsitektur Microservoces Berbasis Kontainer Pada Komunitas Pengembang Perangkat Lunak Sumber Terbuka (Opendaylight Devops Community). *Jurnal Sistem Infomasi Teknologi Informasi Dan Komputer (Just It) Universitas Bina Nusantara Magister Manajemen Sistem Informasi Jakarta*, 150–162.

- Rahadiyan, Inda, & Sari, Alfhica Rezita. (2019). Peluang Dan Tantangan Implementasi Fintech Peer To Peer Lending Sebagai Salah Satu Upaya Peningkatan Kesejahteraan Masyarakat Indonesia. *Defendonesia*, 4(1), 18–28.
- Rahmanda, Rama. (2018). *Rancang Bangun Aplikasi Berbasis Microservice Untuk Klasifikasi Sentimen. Studi Kasus: Pt. Yesboss Group Indonesia (Kata. Ai)*. Institut Teknologi Sepuluh Nopember.
- Rahmatulloh, Alam, Sulastri, Heni, & Nugroho, Rizal. (2018). Keamanan Restful Web Service Menggunakan Json Web Token (Jwt) Hmac Sha-512. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi (Jnteti)*, 7(2). <https://doi.org/10.22146/jnteti.v7i2.417>
- Rajagukguk, Rio Chandra. (2018). *Penggunaan Kriptografi Pada Jwt (Json Web Token) Dalam Implementasi Keamanan Api*.
- Supriyanto, Edi, & Ismawati, Nur. (2019). Sistem Informasi Fintech Pinjaman Online Berbasis Web. *Jurnal Sistem Informasi, Teknologi Informasi Dan Komputer*, 9(2), 100–107.