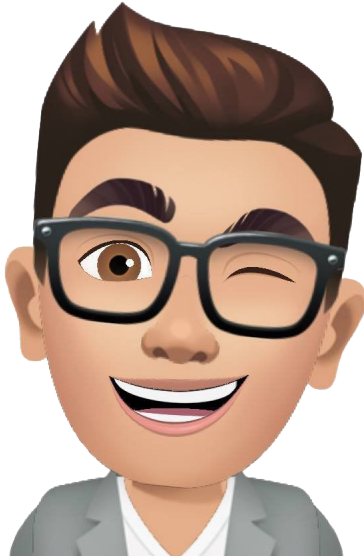




# Backend Introduction

---



# Fahmi Alfareza

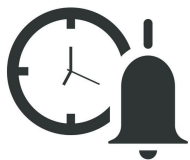
Back-end Facilitator



Don't give up if you have a dream! 😊



# Ground Rules for Students



Masuk lebih cepat  
10 menit/tepat  
waktu



Pastikan jaringan  
internet stabil



Selalu nyalakan  
webcam saat  
kelas berlangsung



Mute saat facil  
menjelaskan  
kecuali Q&A



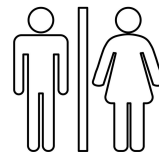
Dilarang Merekam  
webinar



Tekan tombol  
"raise hand" saat  
akan bertanya



Selalu gunakan  
earphone



Toilet Break setiap  
1 jam selama 5-7  
menit



# Ground Rules of The Class



Masuk paling  
lambat 15 menit  
10 menit/tepat  
waktu



Daily Standup  
Saat kelas dimulai



Psychological  
Safety

# Overview

Okay,

We've probably have heard what **Backend** is during the **Induction Meeting**, but maybe you didn't understand it well because it was a very quick explanation.

In this material, we'll learn even more about **what Backend is**. Let's jump into it.

## BACK - END



# BACK - END



## Goals

- You know what **backend engineering** is
- You know **backend's career path**
- You know the **job description** as Backend Engineer
- You know how to **use terminal**
- You know how to **create a hello world application** using Node.js

# What is Backend?

---

# What is Backend?

Backend engineers are software engineers who work on the server components of multi-tier web applications. They focus on web services and the data store (data modeling and databases). They may also be involved with business rule implementation logic.

Backend contrasts with frontend which refers to the customer-facing components such as the user interface and interactive client-side functionality (that in a web application typically uses JavaScript (programming language), AJAX, and frameworks such as jQuery).



# Analogy

Based on this analogy, now you've known that **your job as a backend** is about **cook the data** and **provide the data** for the client side.

And you must now Client Side is the guest room in the restaurant, when customer can order the food. Waiter is the part of it. So, creating **Client Side application** is **Frontend** and **Mobile's responsibility**.





# How do we provide the data for the client side?

Good, question.

Our goal as Backend Engineer is to **create web service (Web API)**. Client Side will communicate with us via this application. By using HTTP Request or another Internet Protocol. We will talk about this later.

# Scopes

As it said earlier, our job is to **make Web API**. That's one of our scope.

**Web API is being used to connect to the Database.** So, maintainig Database and also setting up the database is also our scope.

And also, we have to make our API run as fast as it possible.

Concerning about **Web Security** is also a part of our job.

## Let's sum it up. Here's our scope

- Create Web API
- Maintain Database and everything about Database
- Improve API's performance
- Secure the API

# Career Path



There's **ton of role** that you can land as a Backend Engineer.

Backend Engineer it self is a big job. Because an application is **always scaling up**.

That means our job will be getting bigger and bigger over the time.

Since we're close to the Server, we will have a specific role to **handle the Server and Networking**, it is called **DevOps**.

Most of Backend Engineer upgrading themself to **become DevOps**.

## Developer Roadmap

<https://github.com/kamranahmedse/developer-roadmap/raw/master/img/intro.png>

## Backend Roadmap

<https://gitlab.com/fahmialfareza/materials/-/raw/master/assets/B01-01-BackendRoadmap.png>

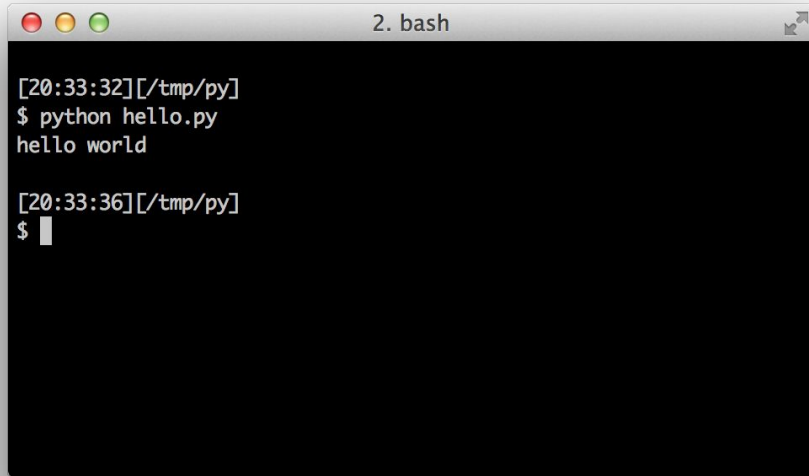
# Terminal

---

# Terminal

What is terminal? **Terminal** is actually a **place where you can command your computer directly via text**

Why via text? Because we can use that dynamically.

A screenshot of a terminal window titled "2. bash". The window has a dark background and shows the following text: [20:33:32][~/tmp/py], \$ python hello.py, hello world, [20:33:36][~/tmp/py], and \$ followed by a cursor. The window has standard macOS window controls (red, yellow, green buttons) in the top left corner.

```
[20:33:32][~/tmp/py]
$ python hello.py
hello world

[20:33:36][~/tmp/py]
$ █
```



BINAR  
ACADEMY

# The Hello World

Now open your terminal (Ctrl + Alt + T), and insert this command to your terminal.

```
echo "Hello World"
```

It prints out the **Hello World** to your Terminal.  
Yay!

```
fahmialfareza@fahmialfareza: ~  
File Edit View Search Terminal Help  
fahmialfareza@fahmialfareza:~$ echo "Hello Worlds"  
Hello Worlds  
fahmialfareza@fahmialfareza:~$
```



## Navigation in Terminal

When you open your terminal, its location will be on

```
/home/yourusername/ # Or it actually just ~
```

It determines your location in the directory. The same way when you open your File Explorer.

It determines your location in the directory. The same way when you open your File Explorer.

In the File Explorer, you can see these folder at the first time open it.

- Documents
- Downloads
- Desktop
- Music
- Pictures
- Videos

That folder is accessible via terminal.



BINAR  
ACADEMY

# Change Directory

It is mentioned, when you open your terminal, by default you're opening the

```
/home/yourusername/
```

Let's say you want to move to the Desktop directory. You can just simply run this command.

```
cd Desktop
```

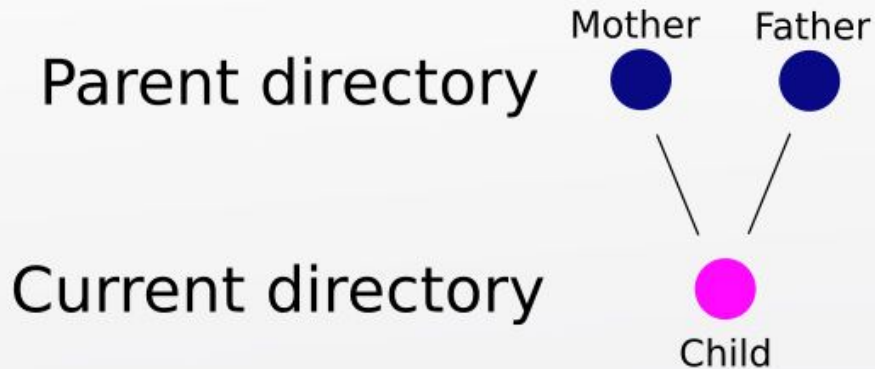
Now, you're on Desktop, how to back in User's Home folder?

```
cd .. # You can use this if your parent folder  
is Home folder
```

```
cd ~ # You can use this anywhere, it will  
direct you to the current user's Home Folder
```

**cd** is a function, and it requires parameter. ~ is parameter, .. is also a parameter, Desktop, ~/Documents, '~'/path/to/the/darkness` are also a parameter.

Why is the parent directory represented by "two" dots and the current dir "one" dot



But from these parameter, you only need to know these three parameter, because it's static.

- . represents current directory that you're currently in.
- .. represents parent directory of your current directory.
- ~ represents Home Folder of current user

# Another Mandatory Command

I won't write the whole command in this Markdown.

You need to learn this:

- **pwd** to show the current path
- **ls** to show list of directory content
- **rm** to remove file/directory
- **mkdir** to create directory
- **touch** to create file
- **cat** to show content of file

<https://www.pcsuggest.com/basic-linux-commands/>

# NodeJS

---



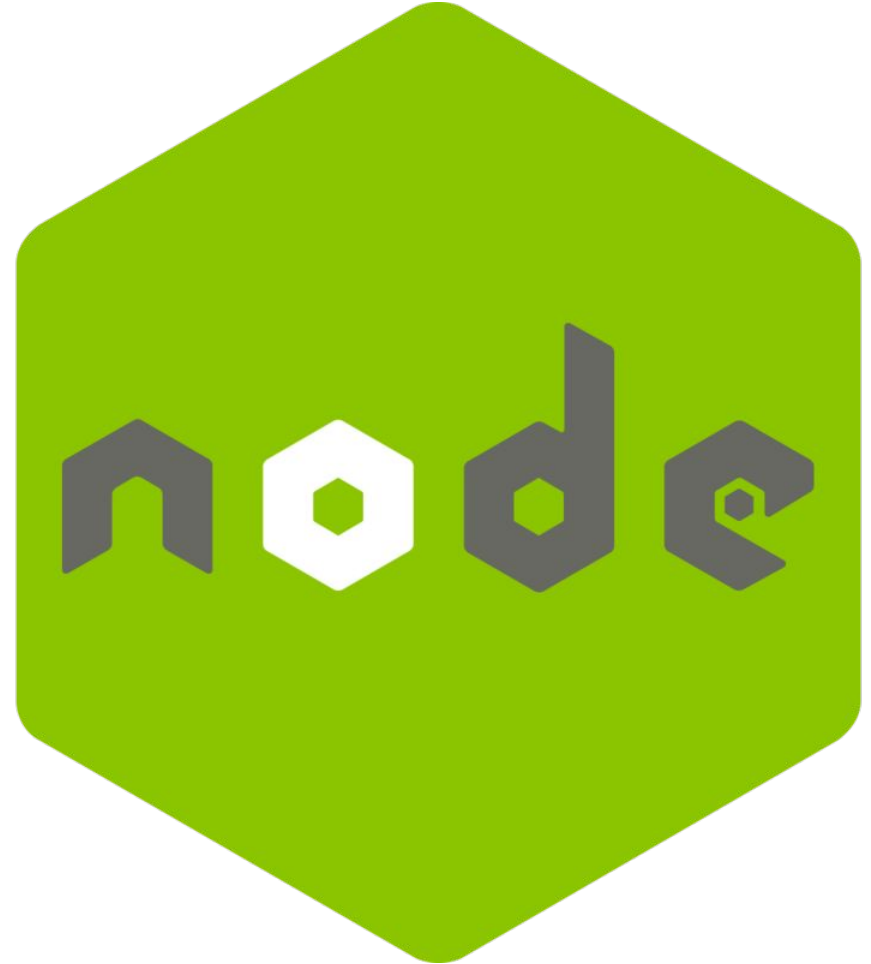
BINAR  
ACADEMY

# What is NodeJS?

Well, you've read about it in the installation instruction.

Basically, it is the **translator** from **Javascript** to the machine language, because **Computer** doesn't understand **Javascript**.

Javascript is a language, that means, not everything understands Javascript. **To make your computer able to speak Javascript**, we need to install Node.js.





Opera



Google Chrome



Safari



Mozilla Firefox



Internet Explorer



Microsoft Edge

**Uh, but Javascript can run in the Browser, why do we need to install Node.JS?**

It only run on Browser, since we're Backend Engineer, we need Node.JS to run Javascript directly on our Machine.

Directly? Since you've installed Node.js, you'll be able to run Node JS inside your terminal. How?



BINAR  
ACADEMY

# Run your first node.js command

```
node -v
```

See the command above? Yes, we can call Node.JS in our terminal. Now, let's make it run our first Javascript Program.

1. Open your text editor (Visual Studio Code)
2. Create new file with .js extension
3. Write this code inside your file.

```
console.log("Hello World!")
```

4. Run that file by using node command.

```
node ~/your/file/location.js
```

5. You'll see hello world printed in the terminal



# Assignment

Make a summary of this session based on Goals of this session!

So, that's it for this session!

# References

- [Developer Roadmap](#)
- [Backend Developer Jobs Description](#)
- [Basic Linux Terminal](#)
- [What is Node.js](#)

**THANK  
YOU**



**BINAR**