workshop.css

forbeginners.html

```
F.E.M ‘Go AWS’ {
  [A course for everyone]

     < Here is where your course begins >

}
```

Programming Language

workshop.css

forbeginners.html

Welcome

<Hello world>

My name is Melkey

I work as an ML infrastructure engineer

I stream live code at twitch.tv/melkey

I make coding videos at youtube.com/@melkey

I make mistake. There will be mistakes. We will solve the mistakes together

Programming Language

workshop.css

forbeginners.html

Table Of ‘Contents’ {

01 Introduction to Course

< Prerequisites and general course info >

02 Go

< Let’s get into learning Go >

03 AWS

< Let’s get into learning AWS >

}

Programming Language

workshop.css

forbeginners.html

```
01 {
    [Introduction to Course ]
        < Prerequisites and general course info >
}
```

Programming Language

workshop.css

forbeginners.html

Prerequisites {

Go

< Have Go installed on your machine >

AWS account

< Free tier >

AWS CLI

< Make sure your .aws/credentials and .aws/config are set up.

`AWS configure` >

CDK

< Have CDK installed on your local computer >

}

Programming Language

Goals of this course {

01 Teach basics of Go

02 Introduce design patterns of Go and recognize common ‘gotchas’

03 Introduce how to use Go with a popular (the most popular) cloud provider AWS

04 Introduce how to deploy infrastructure and run Go code in a production level environment

}

workshop.css

forbeginners.html

What this course is not fake {

01 No hidden expectations

02 Not Perfect [you will get lost,that’s okay]

03 You’re not going to be a pro at Go or AWS

04 You will probably end up doing things differently - and that’ really good!

}

Programming Language

workshop.css

forbeginners.html

1
2
3
4
5
6
7
8
9
10
11
12
13
14

Target audience:

Anyone wanting to learn

Particularly those new to Go and/or AWS

workshop.css

forbeginners.html

How to benefit from this course {

01 DON’T MEMORIZE

02 Look at resources attached

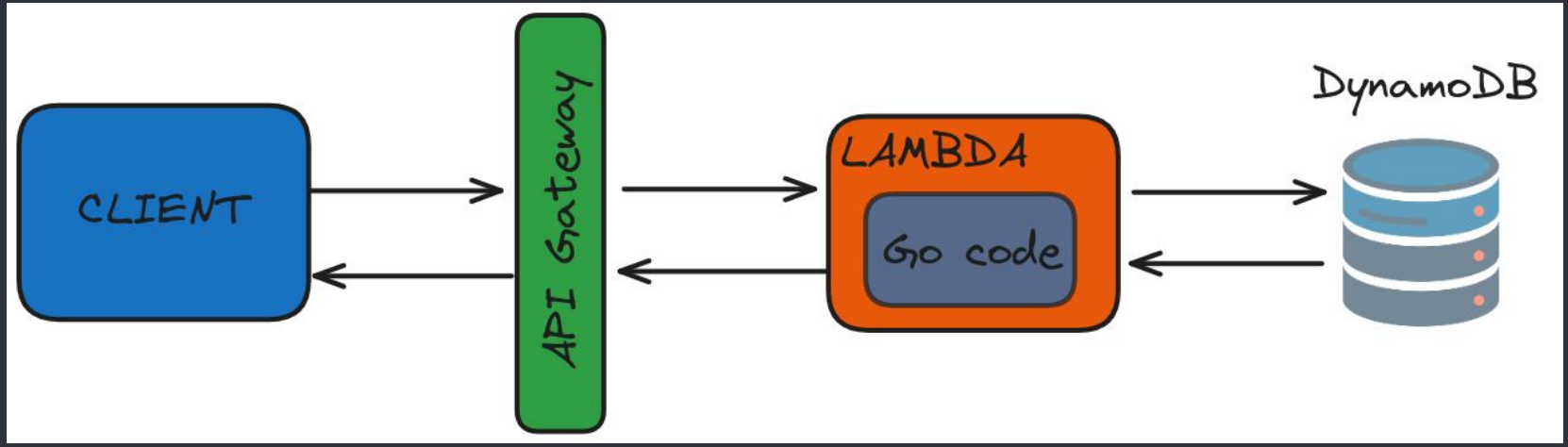
03 Read documentation and best practices

04 Always ask “why is this done the way it is”

}

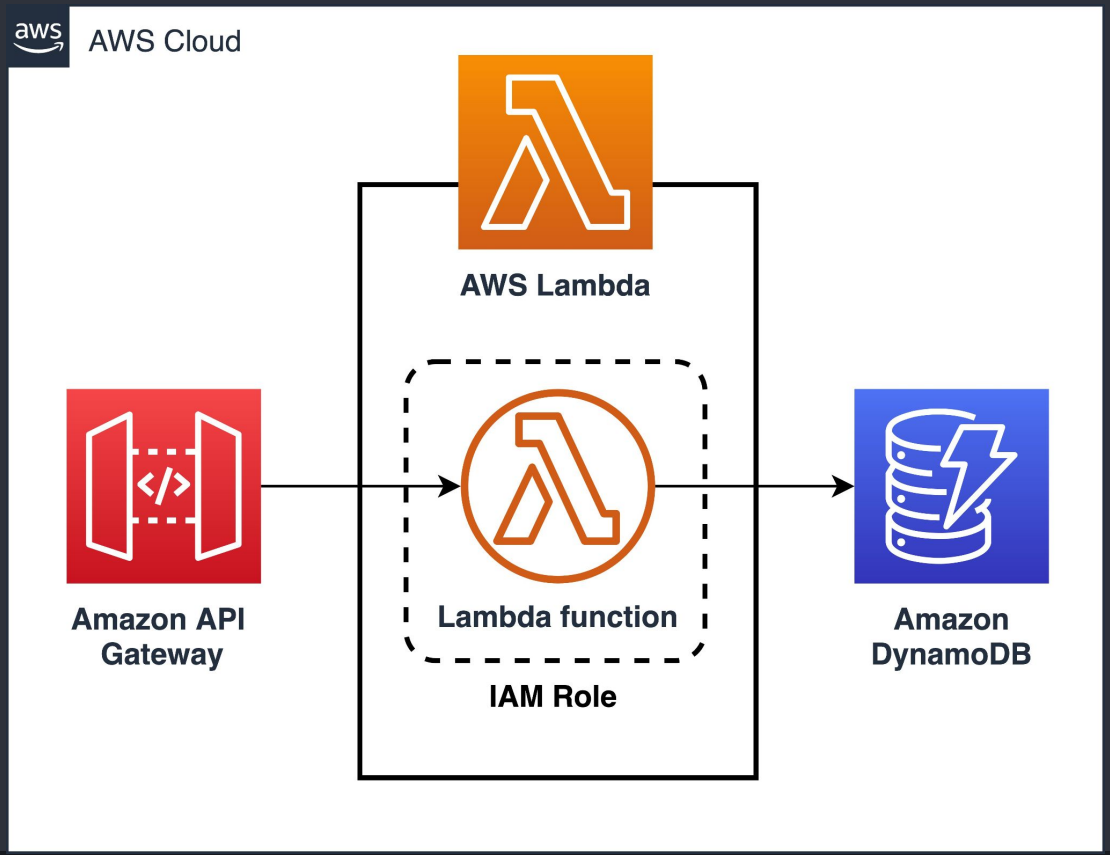
Programming Language

System Diagram



workshop.css

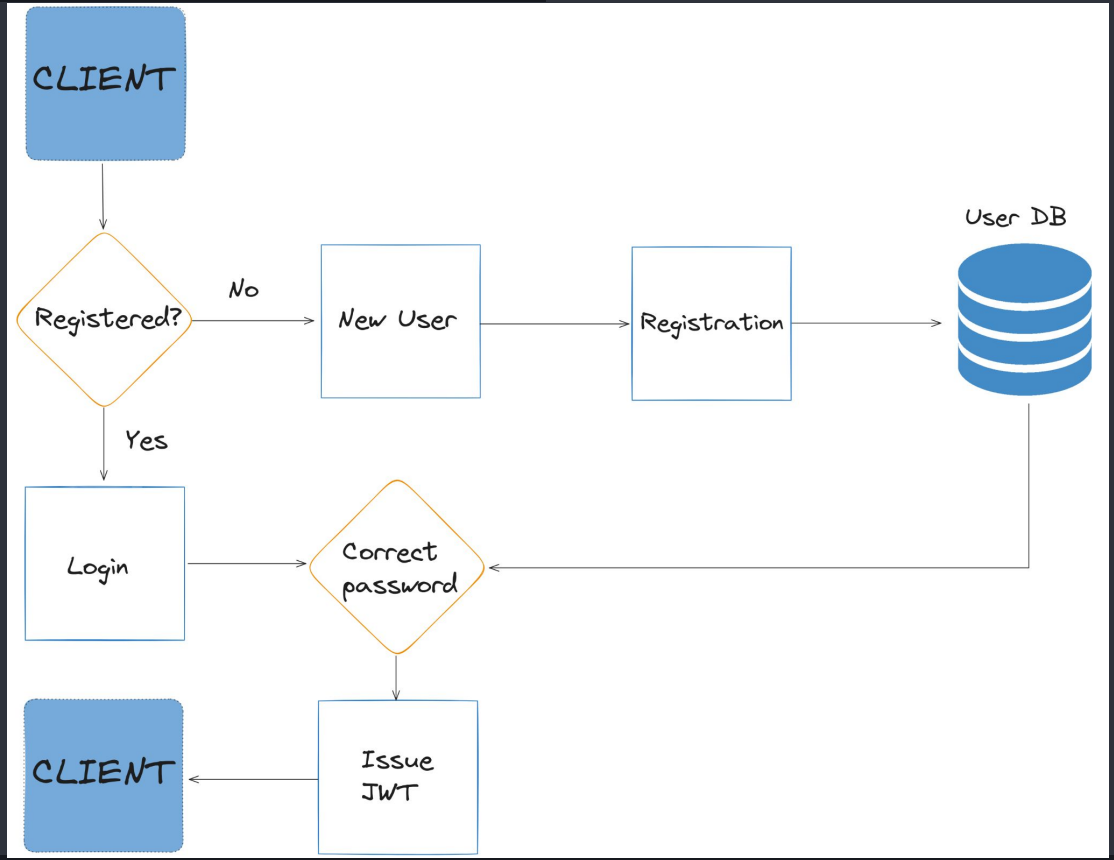
forbeginners.html



Programming Language

workshop.css

forbeginners.html



Programming Language

workshop.css

forbeginners.html

```
02 {

    [Go]

        < Let’s get started with
        Go>

}
```

Programming Language

Go

What?

Statically typed, compiled programming language

Who?

Designed at google by Robert Griesemer, Rob Pike, and Ken Thompson

Why?

Known for its simplicity, efficiency, and strong support for concurrent programming

Created to make writing software easier

Why use Go?

Versatility

Go is really good at a lot of things

Jobs

More and more companies (big tech and startups) are choosing Go to be one of their primary languages = the number of job opportunities for Go may increase

Beginner friendly

Go is great for those new to programming or those who want to switch from a different language

Velocity

Ship applications fast
Write code faster

workshop.css

forbeginners.html

Now is a really good time to learn Go;

Why I Use Golang In 2024

Why We Switched From Svelte Kit To Golang + HTMX

66K views • 2 months ago

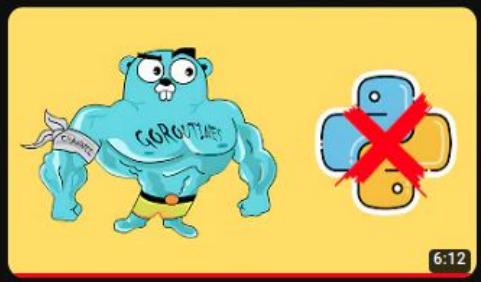
Why I'm learning Go

The Best Resources to Learn Golang (If I Could Start Over)

18K views • 2 weeks ago

I'm Coming Around To Go...

32K views • 8 hours ago



Programming Language

Go resources; {

Websites

< https://go.dev/doc/effective_go
https://gobyexample.com/
https://go.dev/tour/moretypes/2 >

Books

< The Go Programming Language - Donovan Kernighan
Let's Go & Let's Go Further - Alex Edwards>

Conference talks

< Rob Pike conference talks >

} **But, also build stuff!**

LET'S CODE {

Go mod init <name of module>
<name of module> plays a critical role, especially if you make something available for people to use (like an open source package)

Go-blueprint

Package main and func main() entry point

}

forbeginners.html

workshop.css

```
Variables {
    < one line about what a variable is >

    Inferred types

    < one line about what this is >

    Nil values

    < one line about what this is >
}
```

Loops {

< Dynamically-sized arrays
In Go an Array has a fixed size defined at its declaration>

In practice: Slice of animals

< iterate through them with the 2 types of for loops >
< append + delete>
< While loop >

}

Maps {

< The key value store in Go
Equivalent to a hash in other languages >

Nil map

< A map that has no underlying data structure >

< Cannot be directly used to store key-value pairs >

< Attempting to add data to a nil map will cause a runtime panic >

< Useful when you want to declare a map that will later be initialed, possibly conditionally, at some point in your code >

In practice: mySecondMap

< Declare map variable, mySecondMap, that maps strings to integers >

< The variable is initialized to the zero value for maps, which is nil >

}

Structs {

< One line of what this is>

In practice: Person struct

< Pass by copy and not be reference >
< Change the value >

}

forbeginners.html

workshop.css

```
Imports  {
    < One line of what this is>
}
```

Programming Language

Time to Practice

Go exercises

workshop.css

forbeginners.html

```
03 {
    [AWS]
        < Let's get started with
        AWS>
}
```

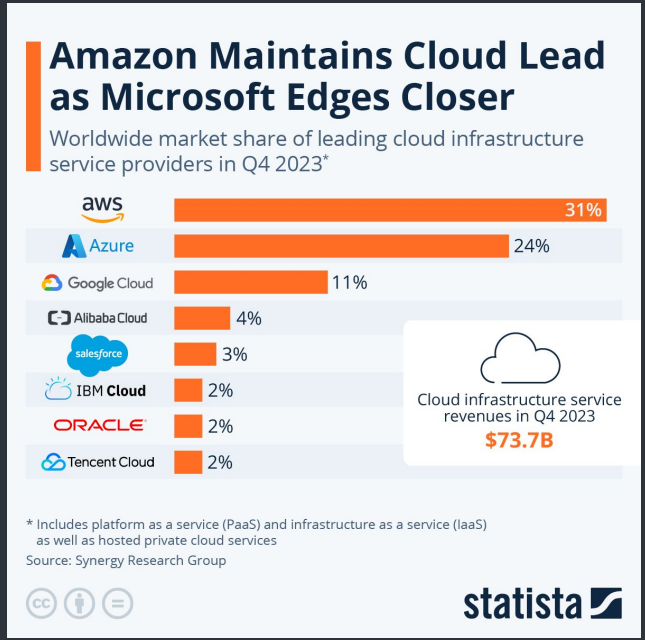
Programming Language

AWS

Amazon Web Services - Cloud

- 200 fully featured services
- Databases
- Computing
- Networking
- Storage options
- Machine learning

- Enables businesses and developers to run virtually everything in the cloud



workshop.css

forbeginners.html

AWS Cloud Development Kit (CDK) {

An open-source software development framework for:

< 1. Defining cloud infrastructure in code >

< 2. Provisioning it through AWS CloudFormation >

Infrastructure as code

Allows us to spin up some light/heavy duty infrastructure services to the cloud without actually interacting with the AWS UI

}

Programming Language

workshop.css

forbeginners.html

What is infrastructure as code? {

Key concept in cloud computing and DevOps practices

Automates the provisioning and management of infrastructure (replaced manual processes)

Infrastructure:

- Defined using code or declarative files
- Allows for infrastructure to be versioned, shared, and reused (similar to application code)

Ensures consistent environments are provisioned every time

- Reduces the likelihood of discrepancies or drifts between development, testing, and production environments

Used a lot in Cloud environments

}

Why AWS + CDK?

Biggest market share

AWS has the biggest market share for Cloud adoption

Mainstream

A lot of popular mainstream companies (Vercel) use AWS under the hood

Many services

AWS has many different services, which CDK allows us to use

Practical

We can combine CDK with Go with create practical, real-world applications

workshop.css

forbeginners.html

Time to practice AWS/CDK {

Step 1 : Install latest version of AWS

https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html

`aws-version`

Step 2 : Configure the AWS user permissions

Create Admin user, get the acess_key_id and the secret_access_key

Step 3 : Configure the AWS CLI

Configure short or long term credentials

https://docs.aws.amazon.com/cli/latest/userguide/getting-started-quickstart.html

Confirm with:

- `aws s3 ls`
- `aws sts get-caller-identity`

Step 4 : Install CDK

https://docs.aws.amazon.com/cdk/v2/guide/getting_started.html#getting_started_install

}

Programming Language

Practice: Chapter 0

< Let's start our CDK project >

`CDK init app --language go`

This spins up a ton of CDK boiler plate for us

`go get`	To install all the dependencies we need
`GOOS=linux GOARCH=amd64 go build -o bootstrap`	To build our executable file
`zip function.zip bootstrap`	Zip the binary for our lambda function
`CDK deploy`	To deploy our stack into the Cloud

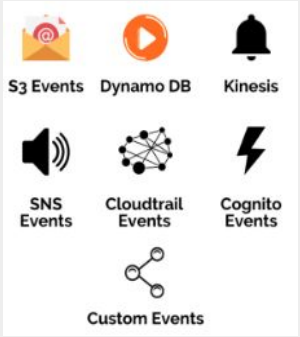
Chapter 0 Continued…

What is AWS Lambda?

Invoked by event

Execute Code

Access any service



Chapter 1

Add better project structure

Create API Handler

Create Database Handler

Create App struct

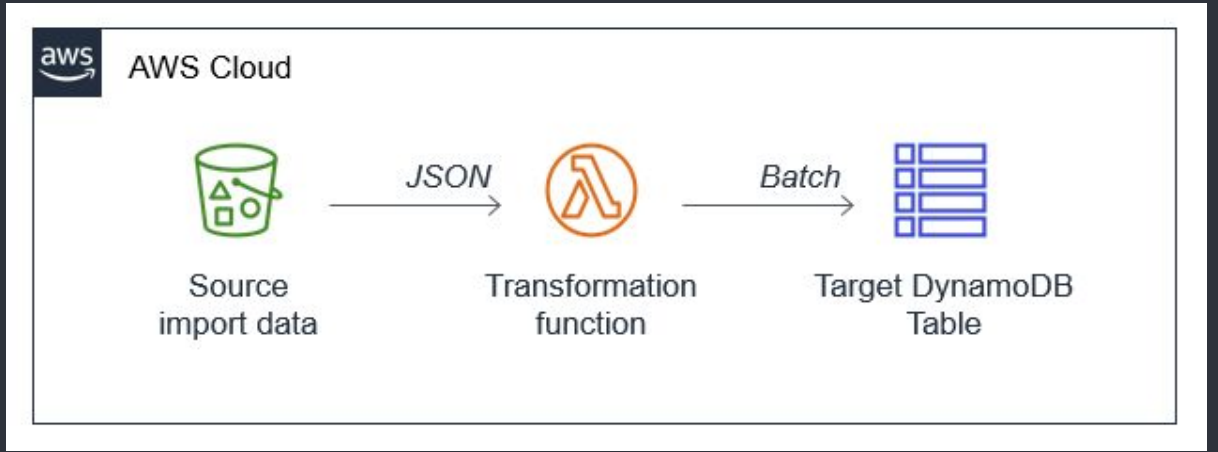
Registering Users

AWS DynamoDB?

Fully managed NoSQL database service

Fast and scalable

Key-value storage



forbeginners.html

workshop.css

Practice Exercise

< Store more values in DynamoDB >

Programming Language

Practice: Chapter 2

Database Interface

Hash user passwords

Create API router with Gateway

Refactor RegisterUsers

Create new LoginUsers function

AWS API Gateway

A fully managed service that makes it easy for developers to create, publish, maintain, monitor and secure APIs at any scale

Acts as a front door for applications to access data, business logic, or functionality from back-end services

Practice: Chapter 3

Add a protected route

Add JWT CreateToken

Add middleware package

cURL with and without token

JSON Web Token

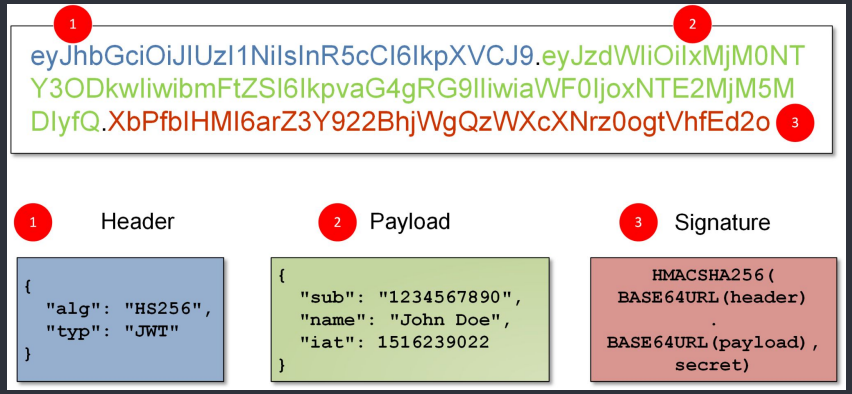
JWT = JSON Web Token

JWT is a compact, URL-safe way of representing claims to be transferred between 2 parties

Enables secure token-based authentication and information exchange

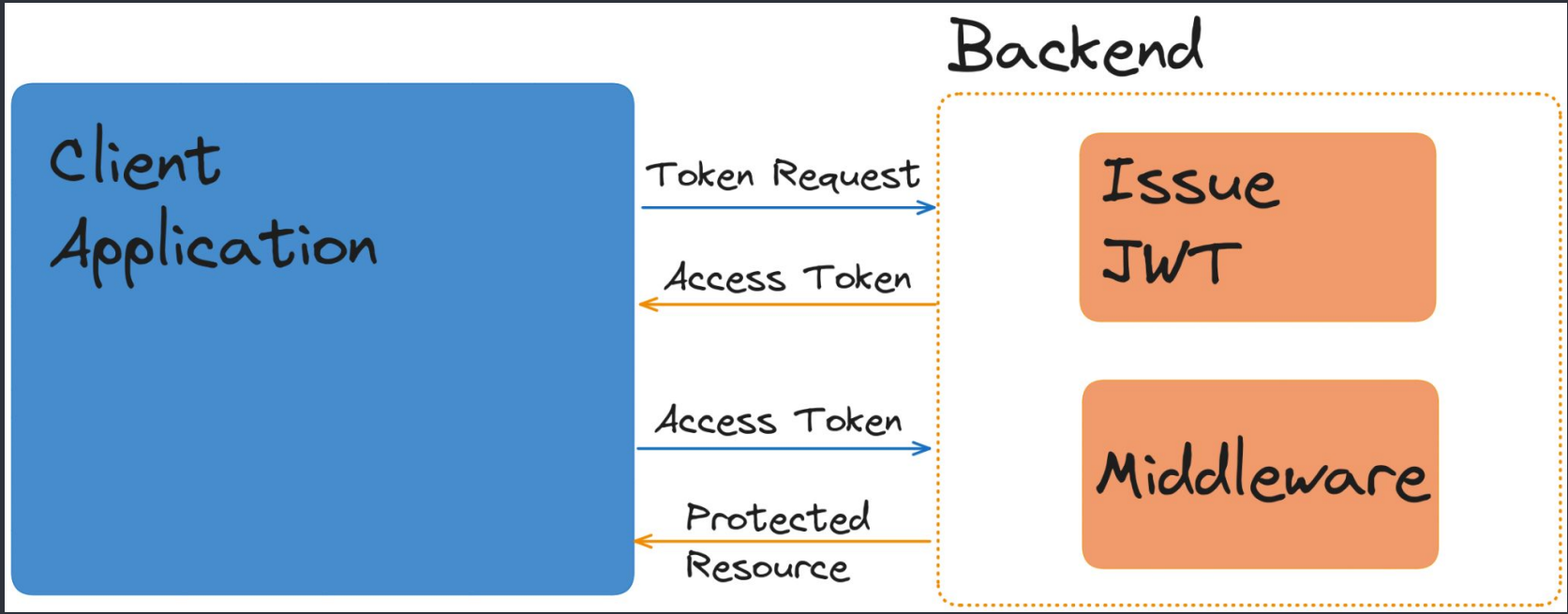
Other methods:

- OAuth 2.0
- LDAP
- OpemIDConnect



workshop.css

forbeginners.html



Programming Language

Thanks; {

‘Do you have any questions?’

youremail@freepik.com
yourcompany.com

}