GoLang 1 CRUD

https://www.youtube.com/watch?v=AQwhAlww0qI

Create a file - main.go

```
package main

// import other packages

import (
    "fmt"
)

func main() {
    fmt.Println("Hello Word!!")
}
```

Get mux

> go get -u github.com/gorilla/mux

Get Grom

> go get -u gorm.io/gorm

MySql Driver

> go get -u gorm.io/driver/mysql

Create model

employee.go

```
package main

type Employee struct {
    EmpName    string `json: "empname"`
    EmpSalary float64 `json: "salary"`
    Email    string `json: "email"`
}
```

main.go

```
package main

func main() {
    DataMigration()
    HandlerRouting()
}
```

db-connection.go

```
package main
import (
    "fmt"
    "gorm.io/driver/mysql"
    "gorm.io/gorm"
)

var Database *gorm.DB

var urlDSN = "root:@tcp(localhost:3306)/mydb?parseTime=true"

var err error

func DataMigration() {

    Database, err = gorm.Open(mysql.Open(urlDSN), &gorm.Config{})
    if err != nil {
        fmt.Print(err.Error())
            panic("Connection error")
    }
    Database.AutoMigrate(&Employee{})
```

Create Routes

routing.go

```
package main

import (
    "log"
    "net/http"

"github.com/gorilla/mux"
```

```
func HandlerRouting() {
    r:= mux.NewRouter()

    r.HandleFunc("/employee", CreateEmployee).Methods("POST")
    r.HandleFunc("/employees", GetEmployees).Methods("GET")
    r.HandleFunc("/employee/{eid}", GetEmployeeById).Methods("GET")
    r.HandleFunc("/employee/{eid}", UpdateEmployee).Methods("PUT")
    r.HandleFunc("/employee/{eid}", DeleteEmployee).Methods("DELETE")

log.Fatal(http.ListenAndServe(":8080",r))
}
```

Create handler

emp-handler.go

```
package main
import (
    "encoding/json"
    "net/http"

    "github.com/gorilla/mux"
)

func CreateEmployee(w http.ResponseWriter, r *http.Request) {
    w.Header().Set("Content-Type", "application/json")
    var emp Employee
    json.NewDecoder(r.Body).Decode(&emp)
    Database.Create(&emp)
    json.NewEncoder(w).Encode(emp)

}

func GetEmployees(w http.ResponseWriter, r *http.Request) {
    w.Header().Set("Content-Type", "application/json")
    var employees []Employee
```

```
Database.Find(&employees)
   json.NewEncoder(w).Encode(employees)
func GetEmployeeById(w http.ResponseWriter, r *http.Request) {
   w.Header().Set("Content-Type", "application/json")
   var employee Employee
   Database.First(&employee, mux.Vars(r)["eid"])
   json.NewEncoder(w).Encode(employee)
func UpdateEmployee(w http.ResponseWriter, r *http.Request) {
   w.Header().Set("Content-Type", "application/json")
   var employee Employee
   Database.First(&employee, mux.Vars(r)["eid"])
   json.NewDecoder(r.Body).Decode(&employee)
   Database.Save(&employee)
   json.NewEncoder(w).Encode(employee)
func DeleteEmployee(w http.ResponseWriter, r *http.Request) {
   w.Header().Set("Content-Type", "application/json")
   var employee Employee
   Database.Delete(&employee, mux.Vars(r)["eid"])
   json.NewEncoder(w).Encode("Employee has been deleted!")
```