



BOOTCAMP BACK-END PROGRAMMING

Go Languages - Phase 2

Full Time Back End Programming

Kisi-Kisi Livecode 1 Phase 2



Program

- **Membuat REST API dari scratch**
- **Mempersiapkan database dengan tabel**
 - **Membuat DDL untuk create table dengan beberapa kolom & constraint (primary key) dengan data type bervariasi(string, int, float)**
- **Membuat endpoint untuk fitur CRUD dari sebuah entitas**

```
router.GET("/products", controller.getAllProducts)
router.GET("/products/:id", controller.getDetailProduct)
router.POST("/products", controller.createProduct)
router.PUT("/products/:id", controller.updateProduct)
router.DELETE("/products/:id", controller.deleteProductById)
```

References

Pelajari terkait routing dan membuat http server dengan net/http dan httprouter

<https://github.com/julienschmidt/httprouter#usage>

```
package main

import (
    "fmt"
    "net/http"
    "log"

    "github.com/julienschmidt/httprouter"
)

func Index(w http.ResponseWriter, r *http.Request, _ httprouter.Params) {
    fmt.Fprint(w, "Welcome!\n")
}

func Hello(w http.ResponseWriter, r *http.Request, ps httprouter.Params) {
    fmt.Fprintf(w, "hello, %s!\n", ps.ByName("name"))
}

func main() {
    router := httprouter.New()
    router.GET("/", Index)
    router.GET("/hello/:name", Hello)

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

References

Pelajari terkait REST API StatusCode dan http method

<https://www.restapitutorial.com/httpstatuscodes.html>

<https://www.restapitutorial.com/lessons/httpmethods.html>

HTTP Verb	Usage	Response Code Success	Response Code Failed
POST	Create Resource	201 (Created)	400 (Bad Request) 500 (Internal Server Error)
GET	Get Resource/s	200 (OK)	500 (Internal Server Error)
PUT	Update Replace Resource	200(OK)	400 (Bad Request) 404(Not found) 500 (Internal Server Error)
PATCH	Update Modify Resource	200(OK)	400 (Bad Request) 404(Not found) 500 (Internal Server Error)
DELETE	Delete Resource	200(OK)	404(Not found) 500 (Internal Server Error)

Case : REST API: Movie Example

Example endpoint, flow, logic, for retrieving all movie from database

```
type MovieRepository struct {  
    DB *sql.DB  
}
```

```
func (mr MovieRepository) FindAll() []Movie {  
    ctx := context.Background()  
    query := `SELECT id, title, rating FROM movies`  
  
    movies := []Movie{}  
    rows, err := mr.DB.QueryContext(ctx, query)  
    if err != nil {  
        panic(err)  
    }  
    defer rows.Close()  
  
    for rows.Next() {  
        m := Movie{}  
        err := rows.Scan(&m.Id, &m.Title, &m.Rating)  
        if err != nil {  
            panic(err)  
        }  
        movies = append(movies, m)  
    }  
  
    return movies  
}
```

```
type MovieController struct {  
    movieRepository models.MovieRepository  
}
```

```
func (c MovieController) Index(w http.ResponseWriter, r *http.Request, p httprouter.Params) {  
    movies := c.movieRepository.FindAll()  
  
    w.Header().Set("Content-Type", "application/json")  
    json.NewEncoder(w).Encode(movies)  
}
```

```
router := httprouter.New()  
  
movieRepository := models.NewMovieRepository(db)  
movieController := controller.NewMovieController(movieRepository)  
router.GET("/movies", movieController.Index)
```

```
router.PanicHandler = func(w http.ResponseWriter, r *http.Request, i interface{}) {  
    w.WriteHeader(http.StatusInternalServerError)  
    json.NewEncoder(w).Encode(ErrorMessage{Message: "Internal Server Error"})  
}
```

REFERENCES

Step-by-step Example

<https://medium.com/@gauravsingharoy/build-your-first-api-server-with-httprouter-in-golang-732b7b01f6ab>

DDL

https://www.w3schools.com/mysql/mysql_create_table.asp