|  |  |  |  |
| --- | --- | --- | --- |
| Модули | Описание | Количество строк кода | Размер (в Кбайтах) |
| 1 | 2 | 3 | 4 |
| .idea\.gitignore |  | 8 | 1 |
| .idea\API\_Kursach.iml |  | 8 | 1 |
| .idea\modules.xml |  | 7 | 1 |
| .idea\workspace.xml |  | 71 | 4 |
| Model\Structs\Struct.go |  | 89 | 3 |
| go.mod |  | 8 | 1 |
| go.sum |  | 4 | 1 |
| main.go |  | 77 | 3 |

1. .idea\.gitignore

# Default ignored files

/shelf/

/workspace.xml

# Editor-based HTTP Client requests

/httpRequests/

# Datasource local storage ignored files

/dataSources/

/dataSources.local.xml

2. .idea\API\_Kursach.iml

<?xml version="1.0" encoding="UTF-8"?>

<module type="WEB\_MODULE" version="4">

<component name="Go" enabled="true" />

<component name="NewModuleRootManager">

<content url="file://$MODULE\_DIR$" />

<orderEntry type="inheritedJdk" />

<orderEntry type="sourceFolder" forTests="false" />

</component>

</module>

3. .idea\modules.xml

<?xml version="1.0" encoding="UTF-8"?>

<project version="4">

<component name="ProjectModuleManager">

<modules>

<module fileurl="file://$PROJECT\_DIR$/.idea/API\_Kursach.iml" filepath="$PROJECT\_DIR$/.idea/API\_Kursach.iml" />

</modules>

</component>

</project>

4. .idea\workspace.xml

<?xml version="1.0" encoding="UTF-8"?>

<project version="4">

<component name="AutoImportSettings">

<option name="autoReloadType" value="ALL" />

</component>

<component name="ChangeListManager">

<list default="true" id="27f9bedf-8b4e-4517-b0d4-bba31a13aa57" name="Changes" comment="" />

<option name="SHOW\_DIALOG" value="false" />

<option name="HIGHLIGHT\_CONFLICTS" value="true" />

<option name="HIGHLIGHT\_NON\_ACTIVE\_CHANGELIST" value="false" />

<option name="LAST\_RESOLUTION" value="IGNORE" />

</component>

<component name="FileTemplateManagerImpl">

<option name="RECENT\_TEMPLATES">

<list>

<option value="Go File" />

</list>

</option>

</component>

<component name="GOROOT" url="file://C:/Program Files/Go" />

<component name="GoLibraries">

<option name="indexEntireGoPath" value="false" />

</component>

<component name="MarkdownSettingsMigration">

<option name="stateVersion" value="1" />

</component>

<component name="ProjectId" id="2HznsSnT9HB2VnOTBmhOtGk6ypx" />

<component name="ProjectViewState">

<option name="hideEmptyMiddlePackages" value="true" />

<option name="showLibraryContents" value="true" />

</component>

<component name="PropertiesComponent">

<property name="DefaultGoTemplateProperty" value="Go File" />

<property name="RunOnceActivity.OpenProjectViewOnStart" value="true" />

<property name="RunOnceActivity.ShowReadmeOnStart" value="true" />

<property name="WebServerToolWindowFactoryState" value="false" />

<property name="go.format.on.save.advertiser.fired" value="true" />

<property name="go.formatter.settings.were.checked" value="true" />

<property name="go.import.settings.migrated" value="true" />

<property name="go.modules.go.list.on.any.changes.was.set" value="true" />

<property name="go.watchers.conflict.with.on.save.actions.check.performed" value="true" />

<property name="last\_opened\_file\_path" value="$PROJECT\_DIR$" />

</component>

<component name="RecentsManager">

<key name="CopyFile.RECENT\_KEYS">

<recent name="D:\GGWP\API\_Kursach" />

</key>

</component>

<component name="RunManager">

<configuration name="go build API\_Kursach" type="GoApplicationRunConfiguration" factoryName="Go Application" temporary="true" nameIsGenerated="true">

<module name="API\_Kursach" />

<working\_directory value="$PROJECT\_DIR$" />

<kind value="PACKAGE" />

<package value="API\_Kursach" />

<directory value="$PROJECT\_DIR$" />

<filePath value="$PROJECT\_DIR$/main.go" />

<method v="2" />

</configuration>

<recent\_temporary>

<list>

<item itemvalue="Go Build.go build API\_Kursach" />

</list>

</recent\_temporary>

</component>

<component name="SpellCheckerSettings" RuntimeDictionaries="0" Folders="0" CustomDictionaries="0" DefaultDictionary="application-level" UseSingleDictionary="true" transferred="true" />

<component name="TypeScriptGeneratedFilesManager">

<option name="version" value="3" />

</component>

<component name="VgoProject">

<integration-enabled>true</integration-enabled>

</component>

</project>

5. Model\Structs\Struct.go

package Structs

type Post struct {

Id\_post int `json:"Id\_post"`

Name\_Post int `json:"Name\_Post"`

}

type Marks struct {

ID\_Marks int `json:"ID\_Marks"`

Name\_Marks int `json:"Name\_Marks"`

}

type Configurat struct {

ID\_Configurat int `json:"ID\_Configurat"`

Name\_Configurat int `json:"Name\_Configurat"`

Price int `json:"Price"`

}

type Model struct {

ID\_Model int `json:"ID\_Model"`

Name\_Model int `json:"Name\_Model"`

Marks\_ID int `json:"Marks\_ID"`

Configurat\_ID int `json:"Configurat\_ID"`

}

type Gears struct {

ID\_Gears int `json:"ID\_Gears"`

Name\_Gears int `json:"Name\_Gears"`

Price int `json:"Price"`

Model\_ID int `json:"Model\_ID"`

}

type Roles struct {

ID\_Roles int `json:"ID\_Roles"`

NameR int `json:"NameR"`

}

type Users struct {

ID\_Users int `json:"ID\_Users"`

Login string `json:"Login"`

Passwor string `json:"Passwor"`

Roles\_ID int `json:"Roles\_ID"`

}

type Cont\_info struct {

ID\_Cont\_info int `json:"ID\_Cont\_info"`

Title int `json:"Title"`

Adress int `json:"Adress"`

Numbe int `json:"Numbe"`

Desribe int `json:"Desribe"`

}

type Employee struct {

ID\_Employee int `json:"ID\_Employee"`

first\_Name int `json:"first\_Name"`

Name\_Employee int `json:"Name\_Employee"`

Middle\_Name int `json:"Middle\_Name"`

Date\_Bithdais int `json:"Date\_Bithdais"`

Employee\_Number int `json:"Employee\_Number"`

Employee\_Email int `json:"Employee\_Email"`

Post\_ID int `json:"Post\_ID"`

Users\_ID int `json:"Users\_ID"`

Active int `json:"Active"`

}

type Soiscatel struct {

ID\_Soiscatel int `json:"ID\_Soiscatel"`

first\_Name int `json:"first\_Name"`

Name\_s int `json:"Name\_s"`

Middle\_Name\_s int `json:"Middle\_Name\_s"`

Confirm int `json:"Confirm"`

Roles\_ID int `json:"Roles\_ID"`

User\_ID int `json:"User\_ID"`

}

type Client struct {

ID\_Client int `json:"ID\_Client"`

first\_Name int `json:"first\_Name"`

Middle\_Name\_s int `json:"Middle\_Name\_s"`

Post int `json:"Post"`

Date\_Bithdais int `json:"Date\_Bithdais"`

Client\_Number int `json:"Client\_Number"`

Client\_Email int `json:"Client\_Email"`

User\_ID int `json:"User\_ID"`

Active int `json:"Active"`

}

type Cheak struct {

ID\_Cheak int `json:"ID\_Cheak"`

Numbers int `json:"Numbers"`

Dates int `json:"Dates"`

Counts int `json:"Counts"`

Employee\_ID int `json:"Employee\_ID"`

Client\_ID int `json:"Client\_ID"`

Model\_ID int `json:"Model\_ID"`

}

6. go.mod

module API\_Kursach

go 1.18

require (

github.com/gorilla/mux v1.8.0

github.com/lib/pq v1.10.7

)

7. go.sum

github.com/gorilla/mux v1.8.0 h1:i40aqfkR1h2SlN9hojwV5ZA91wcXFOvkdNIeFDP5koI=

github.com/gorilla/mux v1.8.0/go.mod h1:DVbg23sWSpFRCP0SfiEN6jmj59UnW/n46BH5rLB71So=

github.com/lib/pq v1.10.7 h1:p7ZhMD+KsSRozJr34udlUrhboJwWAgCg34+/ZZNvZZw=

github.com/lib/pq v1.10.7/go.mod h1:AlVN5x4E4T544tWzH6hKfbfQvm3HdbOxrmggDNAPY9o=

8. main.go

package main

import (

"API\_Kursach/Model/Structs"

"database/sql"

"encoding/json"

"fmt"

"github.com/gorilla/mux"

\_ "github.com/lib/pq"

"log"

"net/http"

)

const (

host = "localhost"

port = 5432

user = "postgres"

password = "1234"

dbname = "Kyrsach"

)

func main() {

r := mux.NewRouter()

r.HandleFunc("/data/user", UserGet).Methods("Get")

r.HandleFunc("/data/models", ModelsGet).Methods("GET")

//r.HandleFunc("/data/post", PostGet).Methods("GET")

//r.HandleFunc("/data/Marks", MarksGet).Methods("GET")

//r.HandleFunc("/data/Configurat", getdata).Methods("GET")

//r.HandleFunc("/data/Gears", getdata).Methods("GET")

//r.HandleFunc("/data/Roles", getdata).Methods("GET")

//r.HandleFunc("/data/Cont\_info", getdata).Methods("GET")

//r.HandleFunc("/data/Employee", getdata).Methods("GET")

//r.HandleFunc("/data/Soiscatel", getdata).Methods("GET")

//r.HandleFunc("/data/Client", getdata).Methods("GET")

//r.HandleFunc("/data/Cheak", getdata).Methods("GET")

log.Fatal(http.ListenAndServe(":8000", r))

}

//var CurentPosition = mas[10][19]

func UserGet(w http.ResponseWriter, r \*http.Request) {

w.Header().Set("Content-Type", "application/json")

userst := Structs.Users{}

psqlconn := fmt.Sprintf("host=%s port=%d user=%s password=%s dbname=%s sslmode=disable", host, port, user, password, dbname)

db, err := sql.Open("postgres", psqlconn)

if err != nil {

fmt.Println("Ошибка подключения")

return

}

defer db.Close()

rows, err := db.Query("select \* from Users")

defer rows.Close()

for rows.Next() {

err = rows.Scan(&userst.ID\_Users, &userst.Login, &userst.Passwor, &userst.Roles\_ID)

json.NewEncoder(w).Encode(&userst)

}

}

func ModelsGet(w http.ResponseWriter, r \*http.Request) {

w.Header().Set("Content-Type", "application/json")

model := Structs.Model{}

psqlconn := fmt.Sprintf("host=%s port=%d user=%s password=%s dbname=%s sslmode=disable", host, port, user, password, dbname)

db, err := sql.Open("postgres", psqlconn)

if err != nil {

fmt.Println("Ошибка подключения")

return

}

defer db.Close()

rows, err := db.Query("select \* from Model")

defer rows.Close()

for rows.Next() {

err = rows.Scan(&model.ID\_Model, &model.Name\_Model, &model.Configurat\_ID, &model.Marks\_ID)

json.NewEncoder(w).Encode(&model)

}

}