Directory /src/GoMS1Assignment

Name

..
restapi
    database
    server
restclient
    client

Command restclient

Package main invokes the client package to start and run the client.

Subdirectories

Name
..
client

Package client

```
import "GoMS1Assignment/restclient/client"
```


Overview ▾

Package client initialises the handler functions for the client web pages and implements its functions for CRUD operations. It is separated into 3 .go files to segregate the functionalities of the application.

```
client.go: Initialises the templates and handler functions, then starts the client to
run
on the designated port.

handler.go: Implements the handler functions for displaying the web pages of the
client
to perform the CRUD operations.

crud.go: Implements the functions for CRUD operations and invokes the REST API.
```

Index ▾

Package files

client.go crud.go handler.go

Constants

```
const baseURL = "http://localhost:5000/api/v1/courses"
```

```
const key = "2c78afaf-97da-4816-bbee-9ad239abb296"
```

Variables

```
var (
    tpl *template.Template
)
```

func **InitClient**

```
func InitClient()
```

InitClient initialises the templates for displaying the web pages of the client

func **StartClient**

```
func StartClient()
```

StartClient initialises the handler functions then listens on the designated port to start the client running.

func **addCourse**

```
func addCourse(code string, jsonData map[string]string) int
```

addCourse invokes the POST method to the REST API to add a course with the course id and JSON data for title given. Returns the response code received from the REST API.

func **addcourse**

```
func addcourse(w http.ResponseWriter, r *http.Request)
```

addcourse is the handler function to retrieve user input for new course details. Validations are performed to ensure valid course details are submitted. CRUD function for addCourse is invoked.

func **delcourse**

```
func delcourse(w http.ResponseWriter, r *http.Request)
```

delcourse is the handler function to delete a course details as selected by the user. CRUD function for deleteCourse is invoked.

func **deleteCourse**

```
func deleteCourse(code string) int
```

deleteCourse invokes the DELETE method to the REST API to delete a course with the course id given. Returns the response code received from the REST API.

func **getCourse**

```
func getCourse(code string) map[string]courseInfo
```

getCourse invokes the GET method to the REST API to retrieve a course with the course id given. If course id is blank, all courses will be retrieved. Returns the map object after unmarshalling the JSON received from the REST API.

func **index**

```
func index(w http.ResponseWriter, r *http.Request)
```

index is the handler function to display the home page of the client. This is also the page where all courses are retrieved and displayed. CRUD function for getCourse is invoked.

func **initaliseHandlers**

```
func initaliseHandlers(router *mux.Router)
```

initaliseHandlers initialises the handlers for the client.

func **updateCourse**

```
func updateCourse(code string, jsonData map[string]string) int
```

updateCourse invokes the PUT method to the REST API to update a course with the course id and JSON data for title given. Returns the response code received from the REST API.

func **updcourse**

```
func updcourse(w http.ResponseWriter, r *http.Request)
```

updcourse is the handler function to retrieve user input for change in course title. Validations are performed to ensure valid course title are submitted. CRUD function for updateCourse is invoked.

func **validateCourseID**

```
func validateCourseID(courseID string) error
```

validateCourseID checks that user input for course id is valid. Returns error type.

func **validateCourseTitle**

```
func validateCourseTitle(courseTitle string) error
```

validateCourseTitle checks that user input for course title is valid. Returns error type.

type **courseInfo**

courseInfo struct for the json

```
type courseInfo struct {
    Title string `json:"Title"`
}
```

Command restapi

Package main invokes the server package to start and run the REST API.

Subdirectories

Name
..
database
server

Package server

```
import "GoMS1Assignment/restapi/server"
```


Overview ▾

Package server initialises the handler functions for the REST API and implements its functions for CRUD operations. It also initialises the database for interfacing with the database package. It is separated into 2 .go files to segregate the functionalities of the application.

```
server.go: Initialises the handler functions and database, then starts the REST API to
run
on the designated port.

handler.go: Implements the functions for CRUD operations as called by the client.
Interface with the database package for database operations.
```

Index ▾

Package files

handler.go server.go

func **StartServer**

```
func StartServer()
```

StartServer initialises the database and handler functions then listens on the designated port to start the REST API running.

func **addCourse**

```
func addCourse(params map[string]string, w http.ResponseWriter, r *http.Request)
```

addCourse implements the POST method invoked by the client and adds a course with the course id and title given.

func **allcourses**

```
func allcourses(w http.ResponseWriter, r *http.Request)
```

allcourses is the handler function to retrieve all courses. It converts the map object retrieved into JSON and passes it back to the client.

func **course**

```
func course(w http.ResponseWriter, r *http.Request)
```

courses is the handler function for CRUD operations sent by the client. The operations for GET, POST, PUT and DELETE will be determined by switch and its respective functions will be called.

func **deleteCourse**

```
func deleteCourse(params map[string]string, w http.ResponseWriter)
```

deleteCourse implements the DELETE method invoked by the client and deletes a course with the course id given.

func **getCourse**

```
func getCourse(params map[string]string, w http.ResponseWriter)
```

getCourse implements the GET method invoked by the client and retrieves the course detail with the course id given.

func **getDBConfig**

```
func getDBConfig() database.Config
```

getDBConfig retrieves the database configurations and returns a struct.

func **home**

```
func home(w http.ResponseWriter, r *http.Request)
```

home is the handler function to display the home page of the REST API.

func **initDB**

```
func initDB()
```

initDB initialises the database

func **initaliseHandlers**

```
func initaliseHandlers(router *mux.Router)
```

initaliseHandlers initialises the handlers for the REST API.

func **updateCourse**

```
func updateCourse(params map[string]string, w http.ResponseWriter, r *http.Request)
```

updateCourse implements the PUT method invoked by the client and update a course title with the course id given.

func **validKey**

```
func validKey(r *http.Request) bool
```

validKey checks that the access keys supplied to the REST API is valid. Returns a bool.

type **courseInfo**

courseInfo struct for the json

```
type courseInfo struct {
    Title string `json:"Title"`
}
```

func convertJSON

```
func convertJSON(reqBody []byte, w http.ResponseWriter, r *http.Request) courseInfo
```

convertJSON converts the client JSON to object and returns the unmarshal data.

Package database

```
import "GoMS1Assignment/restapi/database"
```


Overview ▾

Package database implements the connection to the database server at the designated port. It performs the CRUD operations to the database as invoked by the REST API.

Index ▾

Package files

database.go

Variables

Connector variable used for CRUD operations

```
var DB *sql.DB
```

func **AddCourse**

```
func AddCourse(courseID string, courseTitle string)
```

AddCourse implements the sql operations to insert a new course as invoked by the REST API.

func **Connect**

```
func Connect(connectionString string) error
```

Connect creates the database connection

func **DeleteCourse**

```
func DeleteCourse(courseID string)
```

DeleteCourse implements the sql operations to delete a course as invoked by the REST API.

func **GetAllCourses**

```
func GetAllCourses() map[string]courseInfo
```

GetAllCourses implements the sql operations to retrieve all courses as invoked by the REST API.

func **GetConnectionString**

```
func GetConnectionString(config Config) string
```

GetConnectionString formats the database connection string and returns it.

func **GetCourse**

```
func GetCourse(courseID string) map[string]courseInfo
```

GetCourse implements the sql operations to retrieve a course as invoked by the REST API.

func **UpdateCourse**

```
func UpdateCourse(courseID string, courseTitle string)
```

UpdateCourse implements the sql operations to update a course as invoked by the REST API.

func **ValidKey**

```
func ValidKey(key string) bool
```

ValidKey implements the sql operations to retrieve the access key and validate if the key provided is valid. Returns a bool.

type **Config**

Config struct to maintain DB configuration properties

```
type Config struct {
    ServerName string
    User       string
    Password   string
    DB         string
}
```

type **courseInfo**

courseInfo struct for the json

```
type courseInfo struct {
    Title string `json:"Title"`
}
```