Go Documentation Server

Package main

```
import "GoMS1/client"
```

Overview
Index

Overview

This is the client program that calls the server program and displays the results

Index

Constants
Variables
func AddNewCourseHandler()
func AddNewMaterialHandler()
func ClearConsole(s string) error
func DownloadAllMaterialsHandler()
func DownloadSpecificMaterialHandler()
func PrintMenu()
func RemoveSpecificCourseHandler()
func RemoveSpecificMaterialHandler()
func ReturnAllCoursesHandler()
func ReturnAllMaterialsHandler()
func ReturnSpecificCourseHandler()
func ReturnSpecificMaterialHandler()
func SetupLog() func()
func UpdateSpecificCourseHandler()
func UpdateSpecificMaterialHandler()
func ValidateInt(s *string) int
func createMultiPart(m Material, j []byte) (string, []byte, error)
func getInput(items ...int) (Course, Material, error)
func main()
func sendRequest(m string, url string, ct string, b []byte)
type Course
type Material

Package files

adder.go deleter.go downloader.go helper.go main.go returner.go updater.go

Constants

```
const (
    ContentTypeJSON = "application/json"
    CourseCode      = iota
    CourseTitle
    CourseDescription
    MaterialId
    MaterialSequence
    MaterialFileName
)
```

Variables

```
var CourseURL = "https://127.0.0.1:5000/v1/courses"
```

func **AddNewCourseHandler**

```
func AddNewCourseHandler()
```

AddNewCourseHandler adds a new course.

func **AddNewMaterialHandler**

```
func AddNewMaterialHandler()
```

AddNewMaterialHandler adds a new material to the specified course. The material to be uploaded has to be stored in the upload folder.

func **ClearConsole**

```
func ClearConsole(s string) error
```

ClearConsole detects the os used and clears the console.

func **DownloadAllMaterialsHandler**

```
func DownloadAllMaterialsHandler()
```

DownloadAllMaterialsHandler downloads all materials in the form of a zip file. It is stored in the download folder

func **DownloadSpecificMaterialHandler**

```
func DownloadSpecificMaterialHandler()
```

DownloadSpecificMaterialHandler downloads the specified material. It is stored in the download folder.

func **PrintMenu**

```
func PrintMenu()
```

PrintMenu displays the menu to the console

func **RemoveSpecificCourseHandler**

```
func RemoveSpecificCourseHandler()
```

RemoveSpecificCourseHandler deletes the specified course.

func **RemoveSpecificMaterialHandler**

```
func RemoveSpecificMaterialHandler()
```

RemoveSpecificMaterialHandler deletes the specified material.

func **ReturnAllCoursesHandler**

```
func ReturnAllCoursesHandler()
```

ReturnAllCoursesHandler returns a list of all courses.

func **ReturnAllMaterialsHandler**

```
func ReturnAllMaterialsHandler()
```

ReturnAllMaterialsHandler returns all materials of the specified course.

func **ReturnSpecificCourseHandler**

```
func ReturnSpecificCourseHandler()
```

ReturnSpecificCourseHandler returns the specified course.

func **ReturnSpecificMaterialHandler**

```
func ReturnSpecificMaterialHandler()
```

ReturnSpecificMaterialHandler return the specified material of the specified course.

func SetupLog

```
func SetupLog() func()
```

SetupLog sets up basic logging for the program

func UpdateSpecificCourseHandler

```
func UpdateSpecificCourseHandler()
```

UpdateSpecificCourseHandler updates the specified course. The COURSE_ID cannot be changed. If there is a need to do so, delete and add course again.

func UpdateSpecificMaterialHandler

```
func UpdateSpecificMaterialHandler()
```

UpdateSpecificMaterialHandler updates the specified material in the specified course. The MATERIAL_ID cannot be changed. If there is a need to do so, delete and add material again.

func ValidateInt

```
func ValidateInt(s *string) int
```

ValidateInt converts a string to int

func createMultiPart

```
func createMultiPart(m Material, j []byte) (string, []byte, error)
```

createMultiPart creates a MultiPart content to send both metadata and file

func getInput

```
func getInput(items ...int) (Course, Material, error)
```

getInput prompts for the necessary user input to gather the data needed to perform the next step.

func main

```
func main()
```

main function starts this program

func **sendRequest**

```
func sendRequest(m string, url string, ct string, b []byte)
```

sendRequest sets up and sends a http request

type **Course**

```
type Course struct {
    Code        string `json:"code"`
    Title       string `json:"title"`
    Description string `json:"description"`
}
```

type **Material**

```
type Material struct {
    Id       string `json:"id"`
    Sequence int    `json:"sequence"`
    FileName string `json:"filename"`
}
```