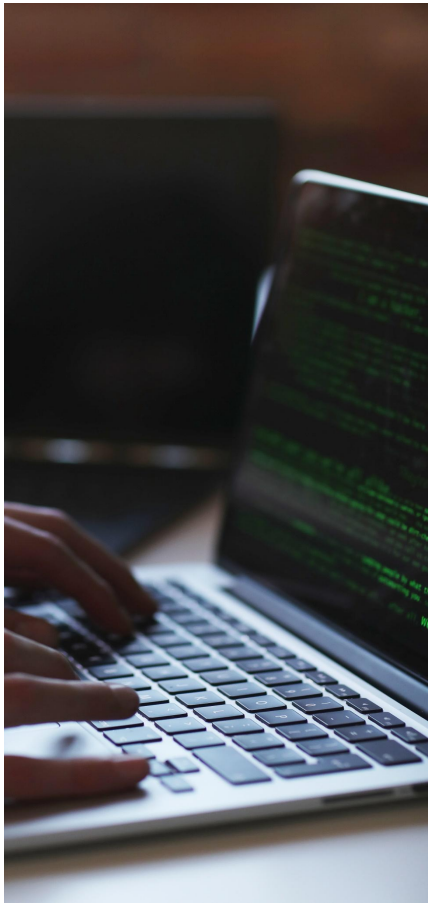
Course-Net
Empowering IT People

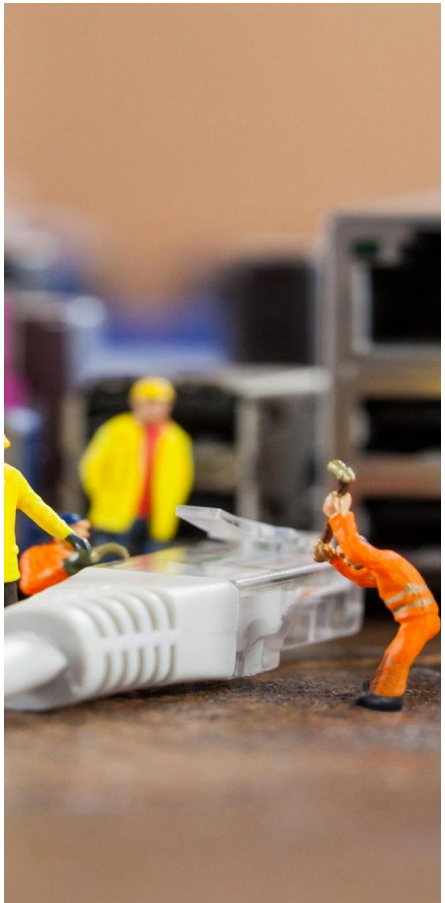
#EmpoweringITPeople

Advanced Golang 3 and Introduction to GORM

Course-Net
Empowering IT People

#EmpoweringITPeople





What to Learn Today?

Advanced Golang

1. Go Modules
2. Database SQL Driver and Library
3. Net/Http Part 1

Course-Net
Empowering IT People

#EmpoweringITPeople

Install MySQL Client Software

Every MySQL client or any DBMS client software are the same, so choose that you understand the most!

1

MySQL Workbench (Ubuntu and Linux User)

Recommended for this course. An example course will use this DBMS client software.

2

XAMPP (Windows User)

Make sure you have understanding in this DBMS client software. This course will not teach you how to use this DBMS client software.

3

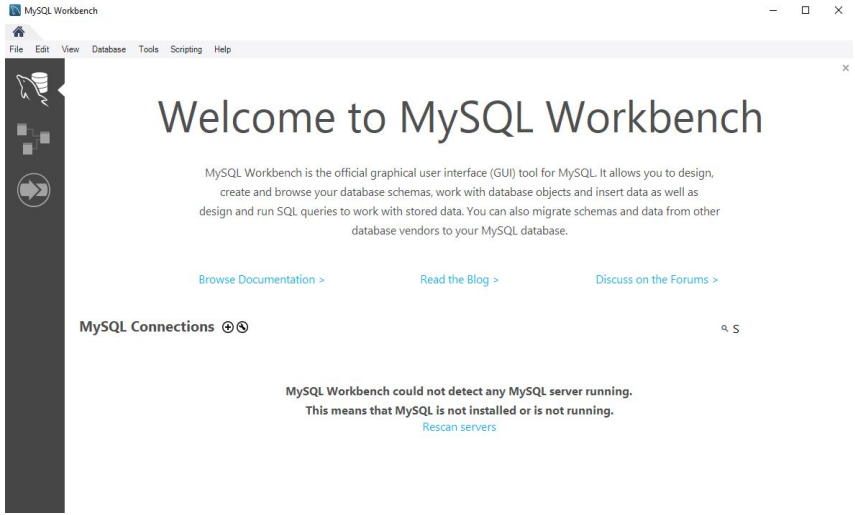
Postgre SQL

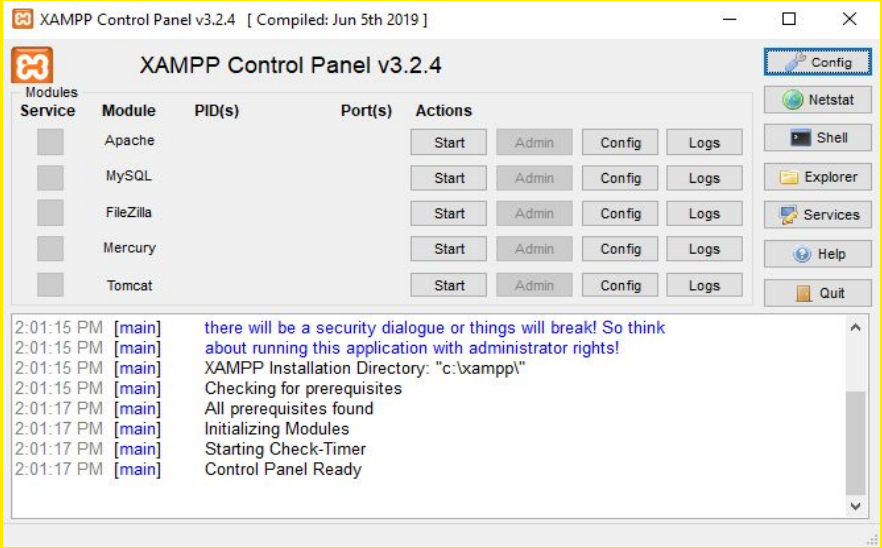
Make sure you have understanding in this DBMS client software. This course will not teach you how to use this DBMS client software.

Course-Net
Empowering IT People

#EmpoweringITPeople

MySQL Workbench vs XAMPP





Helpful Resources To Access Local Database

Ubuntu 20.04:
https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-20-04

Ubuntu 18.04: https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-18-04

MacOS: https://flaviocopes.com/mysql-how-to-install/

How To Access Local Database Using XAMPP?

1. Start MySQL module on XAMPP Control Panel
2. Open any browser
3. Visit localhost/phpmyadmin/

Course-Net
Empowering IT People

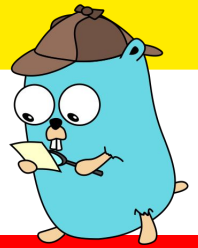
#EmpoweringITPeople

Go Modules

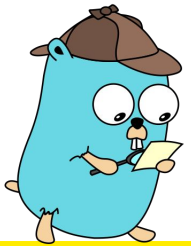
If you're using external library, Go need to know which library do you use, so Go can perform better. Also, when you share your Go code with someone else, Go on the other computer will also recognize what libraries need to be prepared/downloaded. This is also to solve version problems, where it could be a library being updated so that some of the functions you coded for are deprecated. Then, Go will look for the right version of the library so that the function you are using can still be used.

By typing "go mod init" on CLI, you're good to Go!

Library And Version



go.sum



go.mod

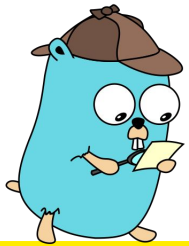
Course-Net
Empowering IT People

#EmpoweringITPeople

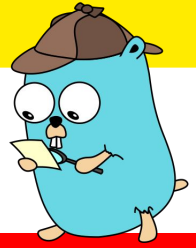
Go Get and Go Mod Tidy

If you're using external library, Go must download the required package that defined by you. After that, Go need to install which package that you have downloaded will be used. Therefore, after executing "go get" on your CLI to download specified library, you should execute "go mod tidy" in let Go determine which library that needs to be installed in order to run your application.

go get -v



go get -u



go get

Try This!

1 Initialize Go Modules

go mod init

2 Download Library

go get
github.com/go-sql-driver/mysql

3 Install The Required Library

go mod tidy

Course-Net
Empowering IT People

#EmpoweringITPeople

SQL Driver

The important thing if you're using SQL driver is to close the connection whenever you've finished the query using SQL driver. This is to prevent multiple connection established to the database that can impact your code performance.

1 Establish Connection

```
db, err := sql.Open("mysql", "user:passwor
d@tcp(host:port)/dbname")
```

Do Query 2

```
_, err := db.Exec(...)
_, err := db.Query(...)
```

3 Close Connection

```
defer db.Close()
```

Course-Net
Empowering IT People

#EmpoweringITPeople

Struct

```
type Student struct {
    ID      int  `db:"id"`
    Name    string `db:"student_name"`
    Address string `db:"student_address"
`
}
```

Sqlx Library

Sqlx library is the one of many library that might be useful for database query. This library helps you to fetch the result from the query result into your struct.

Like JSON, you should define the real database column name in the struct field. So, sqlx will know which database column correspond the struct field.

Sqlx in Action

```
rows, err := db.Query("SELECT * FROM stu
dents")
if err != nil {
    fmt.Println(err)
}

err = sqlx.StructScan(rows,
&studentData)
if err != nil {
    fmt.Println(err)
}
```

Course-Net
Empowering IT People

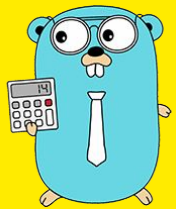
EmpoweringITPeople

Net/Http

Go also prepares a built-in library to deal with user request that is net/http library. So, basically you can build an API without using framework.

```
func main() {
   http.HandleFunc("/", home)
}

func home(w http.ResponseWriter, r *http.Request
) {
   fmt.Fprintln(w, "Hello Web Server Golang")
}
```



Course-Net
Empowering IT People

#EmpoweringITPeople

Do You Think Go Only for Back End? Then, You're Definitely Wrong.

Go also prepares a built-in library to deal with user interface issues. So, with Go alone you can create a complete web without the need for other front-end repositories such as React, JavaScript, and others. So, if you think Go is only for the back-end programming language, you are definitely wrong!

```
var t, err = template.ParseFiles("home.html")
```