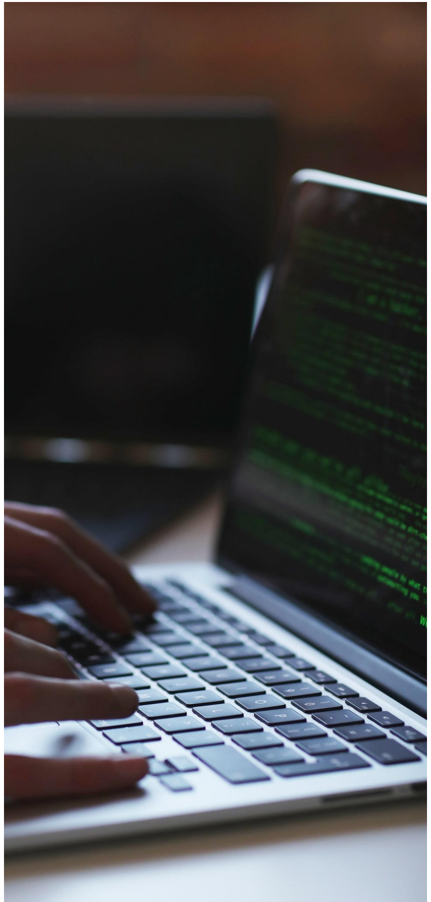
Course-Net
Empowering IT People

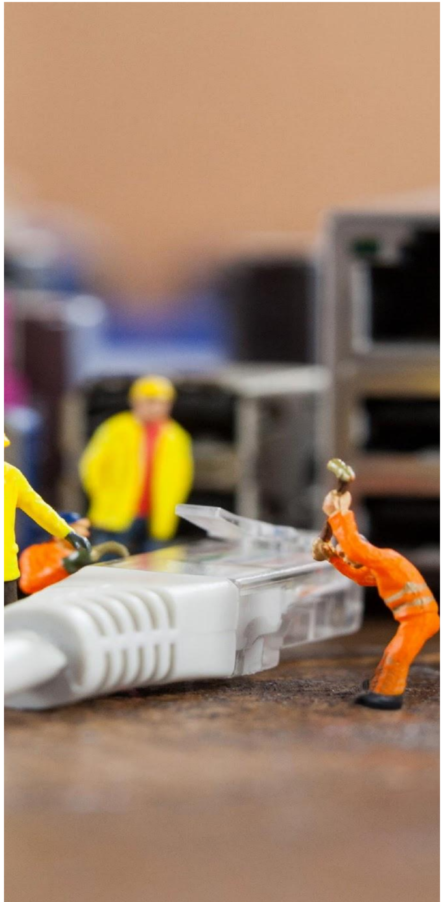
Introduction to GIN Framework

#EmpoweringITPeople

Course-Net
Empowering IT People

EmpoweringITPeople





What to Learn Today?

GIN Framework

1. Introduction to GIN Framework
2. Routing and Group Routing
3. CRUD Implementation by Using GORM

Software Framework

In computer programming, a **software framework** is an abstraction in which software, providing generic functionality, can be selectively changed by additional user-written code, thus providing application-specific software. It provides a standard way to build and deploy applications and is a universal, reusable software environment that provides particular functionality as part of a larger software platform to facilitate the development of software applications, products and solutions. Software frameworks may include support programs, compilers, code libraries, toolsets, and application programming interfaces (APIs) that bring together all the different components to enable development of a project or system.

Reference:
https://en.wikipedia.org/wiki/Software_framework

Introduction to GIN

GIN Installation

```
go get github.com/gin-gonic/gin/
```

Routing in GIN Framework

Normal Routes

```
route.GET("/", getHome)
```

Route Group

```
v1 := route.Group("/api")
{
    v1.GET("/", getHome)
    blogs := v1.Group("/blog")
    {
        blogs.POST("/", postBlog)
    }
}
```

```
route.Run()
```

Common Parameters in GIN Framework

1 Parameters in Path

c.Param(“name”)

2 Query String Parameters

c.Query(“name”)

3 Multipart/Urlencoded Form

c.PostForm(“name”)

Write Log File

By default, GIN will log any request to the console.

If you consider to write the log to file, you should add following code.

```
f, _ := os.Create("gin.log")
gin.DefaultWriter = io.MultiWriter(f
)
```

If you consider to write the log to file and console at the same time, you should add following code.

```
f, _ := os.Create("gin.log")
gin.DefaultWriter = io.MultiWriter(f, os.Stdout)
```