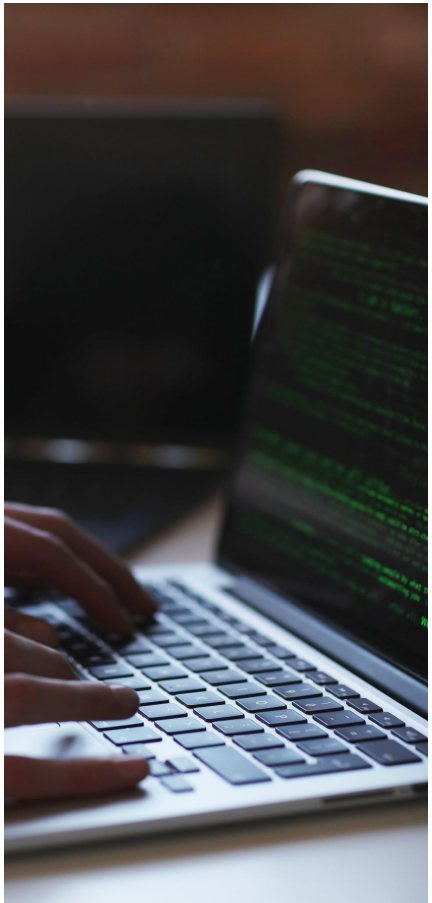
Course-Net
Empowering IT People

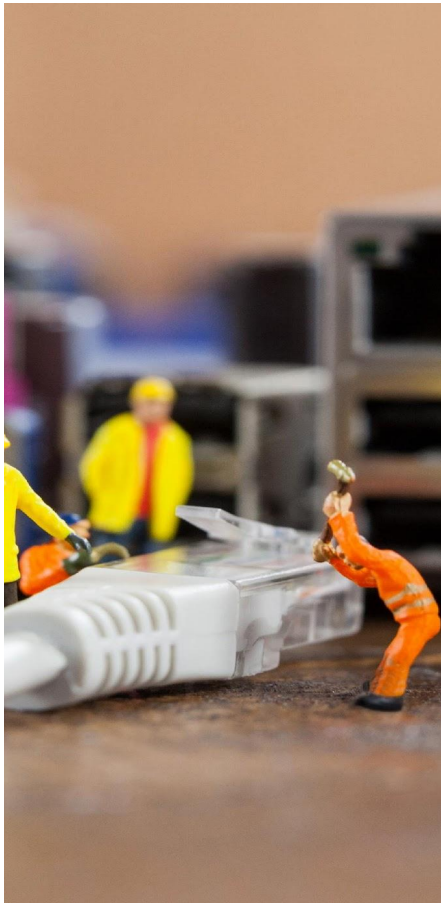
#EmpoweringITPeople

Implement Login, Register, and Middleware

Course-Net
Empowering IT People

#EmpoweringITPeople





What to Learn Today?

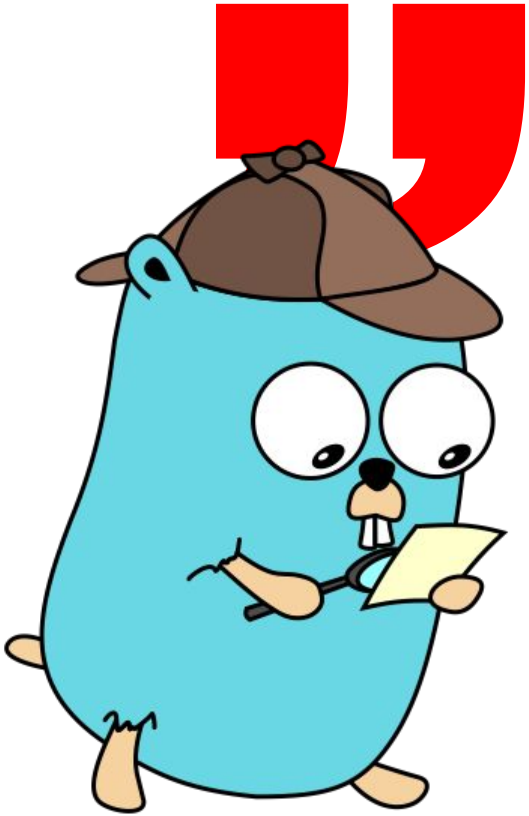
GIN Framework

1. Register & Login
2. Securing Route with Authentication
3. Securing Route with Role Permission

Course-Net
Empowering IT People

#EmpoweringITPeople

JWT (JSON Web Tokens)



JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. This information can be verified and trusted because it is digitally signed. JWTs can be signed using a secret (with the HMAC algorithm) or a public/private key pair using RSA or ECDSA.

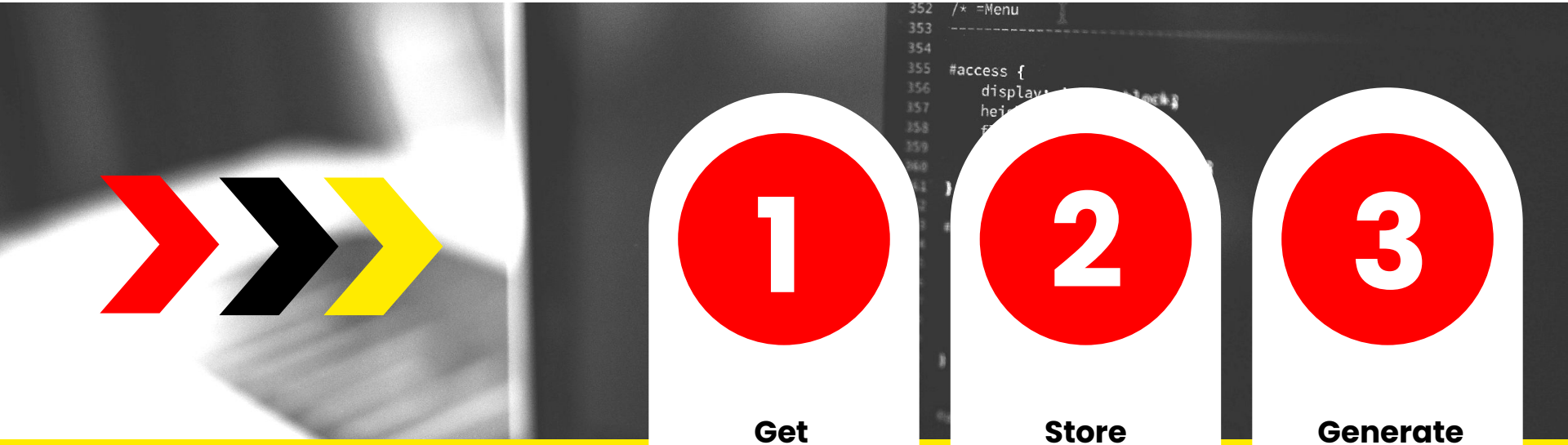
References: https://jwt.io/introduction

Course-Net
Empowering IT People

#EmpoweringITPeople

Implementing Login/Register

Basically, login and register needs to produce token that will be consumed by the user. The difference is that when registering, the system needs to save the user's personal data, but login only requires checking whether the data provided by the user is correct.



Implementing Register

1

Get

Get user's personal data.

2

Store

Save the user's personal data to the database.

3

Generate

Generate new token to the user as their unique key and authentication.

Implementing Login

1

Get

Get user's personal data.

2

Check

Checking of user data, whether it is registered or not.

3

Generate

Generate new token to the user as their unique key and authentication.

Course-Net
Empowering IT People

#EmpoweringITPeople

USER'S DATA IS IMPORTANT!

Remember, if we build an application that store user's data, we should make sure the data saved securely in our database. As a developer, we should be responsible to make a step that will secure our user's data!

So, when our database is hacked, the possibility of user data being known by irresponsible parties is smaller.

Three basics steps to secure our user's data:

1. Secure our database
2. Encrypt important data from user before we save it to the database
3. Protect token with secret key
4. Protect the route with appropriate access rights

Course-Net
Empowering IT People

#EmpoweringITPeople

JWT Checking

References:
https://pkg.go.dev/github.com/dgrijalva/jwt-go#Parse

```
// Token from another example.  This token is expired
var tokenString = "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJmb28iOiJiYXIiLC
JleHAiOjE1MDAwLCJpc3MiOiJ0ZXN0In0.HE7fK0xOQwFEr4WDgRWj4teRPZ6i3GLwD5YCm6Pwu_c"

token, err := jwt.Parse(tokenString, func(token *jwt.Token) (interface{}, error) {
    return []byte("AllYourBase"), nil
})

if token.Valid {
    fmt.Println("You look nice today")
} else if ve, ok := err.(*jwt.ValidationError); ok {
    if ve.Errors&jwt.ValidationErrorMalformed != 0 {
        fmt.Println("That's not even a token")
    } else if ve.Errors&(jwt.ValidationErrorExpired|jwt.ValidationErrorNotValidYet) != 0 {
        // Token is either expired or not active yet
        fmt.Println("Timing is everything")
    } else {
        fmt.Println("Couldn't handle this token:", err)
    }
} else {
    fmt.Println("Couldn't handle this token:", err)
}
```