# go web 学习笔记

1. Web是基于HTTP协议进行交互的应用网络，通过浏览器/App访问各种资源
2. **session、cookie、token**

http是无状态的连接，但是我们不想每次访问服务器都要输入账号密码，所以就有了session、cookie和token来让服务器保持登录状态。

首先浏览器发起一个带有账号密码的HTTP请求。服务器收到并验证成功后会设置带有账号密码信息的cookie，然后发送给浏览器，浏览器会保存起来，这样浏览器以后的每次请求只需带上这个cookie就不用输入账号密码了。但这样是不安全的，只要电脑被黑，存放在浏览器的cookie就会泄露。

然后有了新的概念session也就是会话，服务器进行账号密码验证成功后会对每个用户的会话设置一个唯一的sessionID和会话结束时间，保存在数据库里，并且将sessionID加入到发送给客户端的cookie里，再把会话结束时间设置为cookie的有效期。浏览器拿到cookie后进行保存，加入每次的连接请求中。当cookie失效后，浏览器一般会自行删除这个cookie（会话结束）。用户就得重新输入账号密码了。但当用户群体十分大时，存储大量的sessioID在服务器里会加大服务器的压力，可能会导致服务器的超载。

然后又出现了一种新的技术JWT全称JSON Web Token，浏览器输入账号密码发送给服务器，服务器验证成功后就会生成JWT，而服务器不需要保存JWT只需要保存JWT签名的密文接着把JWT发送给浏览器，可以让浏览器以cookie或者Storage的形式进行存储，这样用户每次发送请求都会把这个JWT发送给服务器，就不需要重新输入账号密码了和session很类似，只是这里的Token是存储在浏览器这边。

Token的安全性：

JWT由三部分组成：header.payload.signature。header部分声明需要用什么算法来生成签名。payload部分是一些特定的数据，比如有效期之类的。header和payload两部分会经由Base64编码，这两段编码会结合服务器保存的密码进行算法运算（header声明的算法），最终得到签名信息signature，这样一个完整的JWT就可以发送给服务器了。

session相当于会话的属性，cookie相当于传送信息的载体，Token相当一个服务器发的令牌，有它就可以访问服务器。

1. 路由：不同的url地址和不同的地址资源进行对应
2. Web程序的两个必须函数：
3. func http.HandleFunc(string,

func(w http.ResponseWriter, req \*http.Request ){...})

//注册了一个处理器函数和对应的模式

//当前端请求到达时第二个参数的功能开始体现

1. func ListenAndServe(addr string, handler Handler)

//监听TCP地址addr，并且会使用handler参数调用Serve函数处理接收到的连接。

//第二参数为nil指的是使用默认路由http.DefaultServerMux

//第二种写法如下：

server:=http.Server{

addr:”:9090”,

Handler:nil,

}

server.ListenAndServe( )

1. 使用https时：

server.ListenAndServeTLS( )------>server.ListenAndServeTLS( )

1. http.Handler是接口,有一个方法ServeHTTP(ResponseWriter,\*Request)