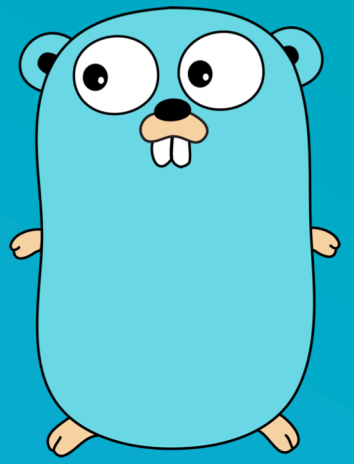
2019 TECH(K)NOW Day Taipei

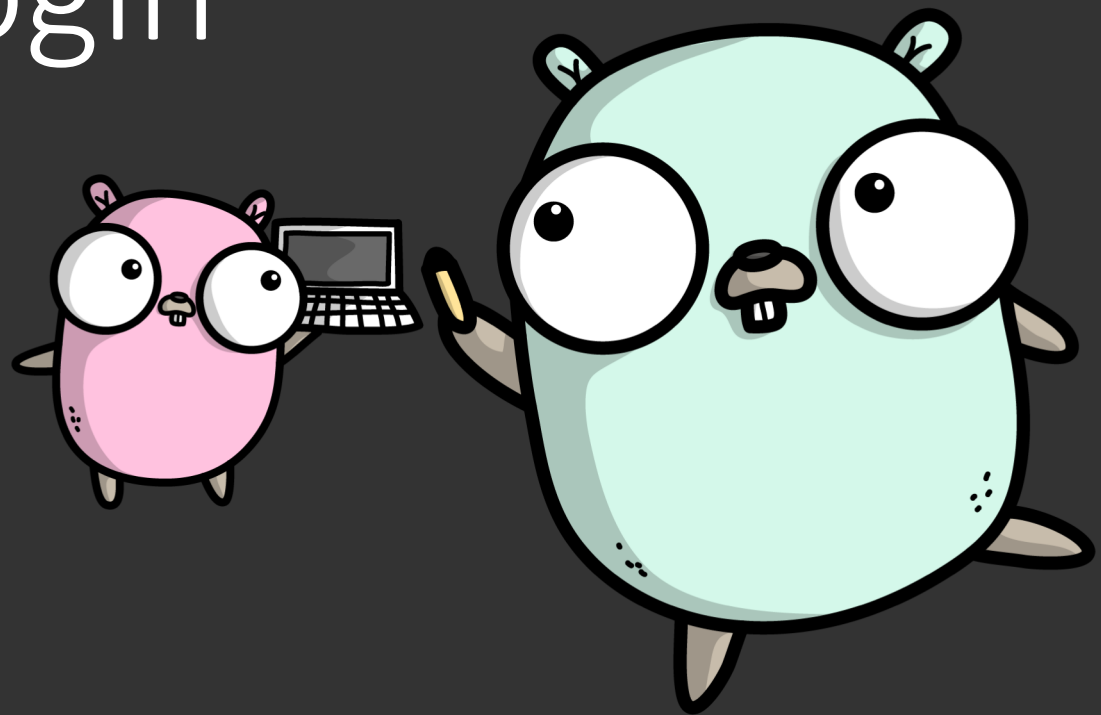
Let's GO For Web!

Benjamin Wang @ Golang Taiwan



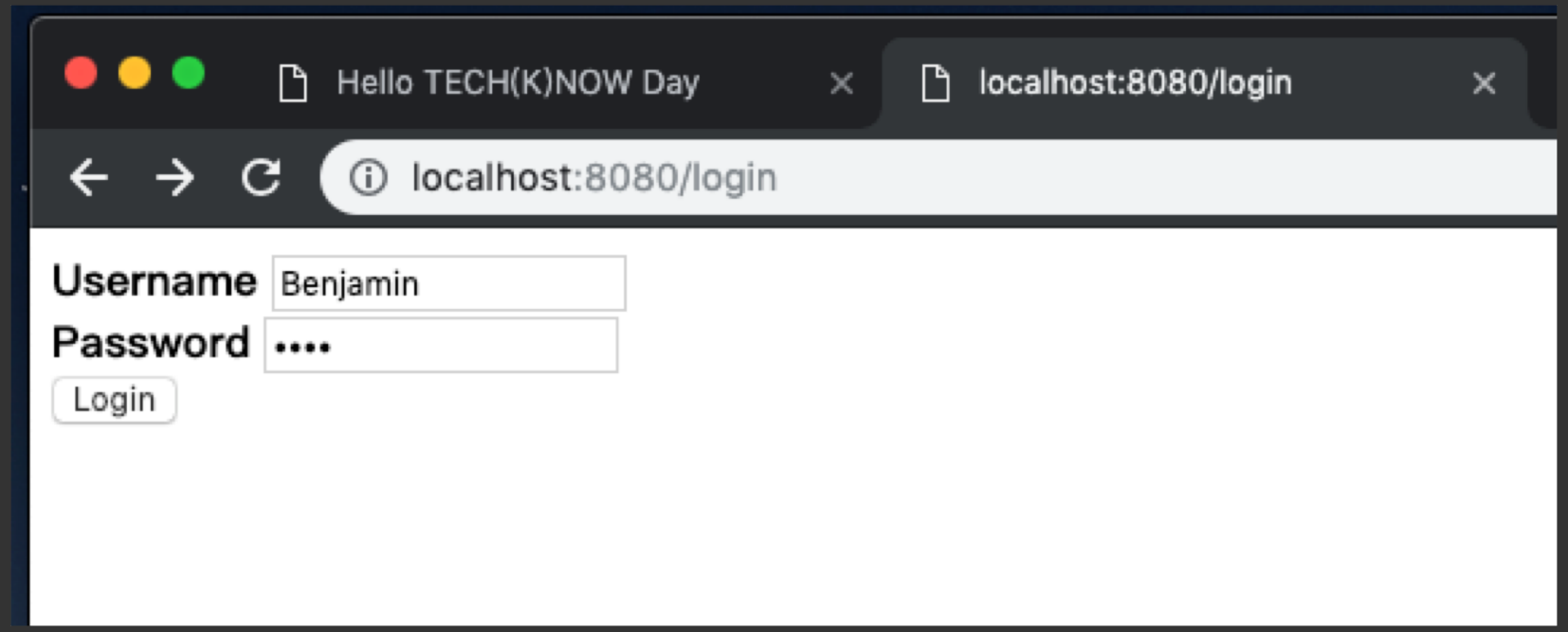
2019.4.20

User Login

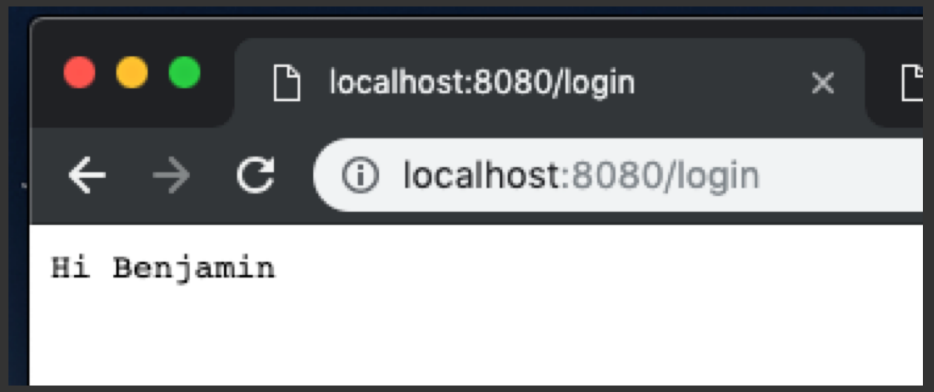


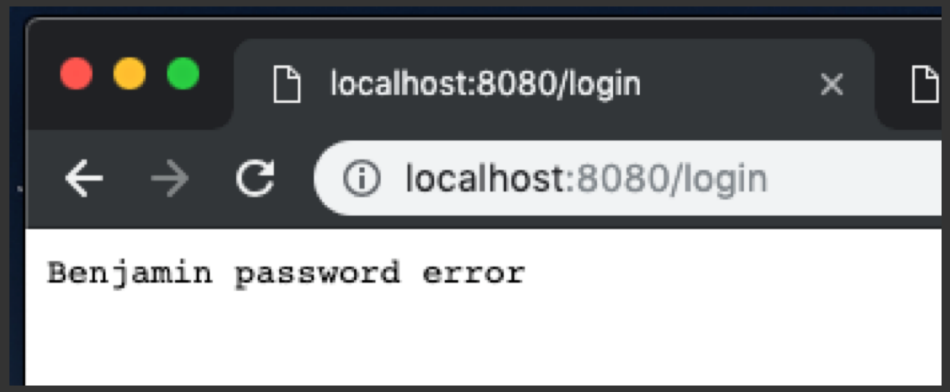
```
r.GET("/login", loginPage)


func loginPage(c *gin.Context) {
    c.HTML(http.StatusOK, "login.html.tmpl", gin.H{})
}
```

```
<form action="login" method="POST">
    <div>
        <label for="username"><b>Username</b></label>
        <input type="text" name="username" id="username" required>
        <br>
        <label for="password"><b>Password</b></label>
        <input type="password" name="password" id="password" required>
        <br>
        <button type="submit">Login</button>
    </div>
</form>
```


Hello TECH(K)NOW Day
localhost:8080/login
localhost:8080/login
Username
Benjamin
Password
••••
Login

```go
func login(c *gin.Context) {
    username := c.PostForm("username")
    password := c.PostForm("password")
    log.Println("username:", username)
    if username == "Benjamin" && password == "1234" {
        c.String(http.StatusOK, "Hi "+username)
    } else {
        c.String(http.StatusOK, fmt.Sprintf("%s password error", username))
    }
}
```


localhost:8080/login
localhost:8080/login
Hi Benjamin


localhost:8080/login
localhost:8080/login
Benjamin password error

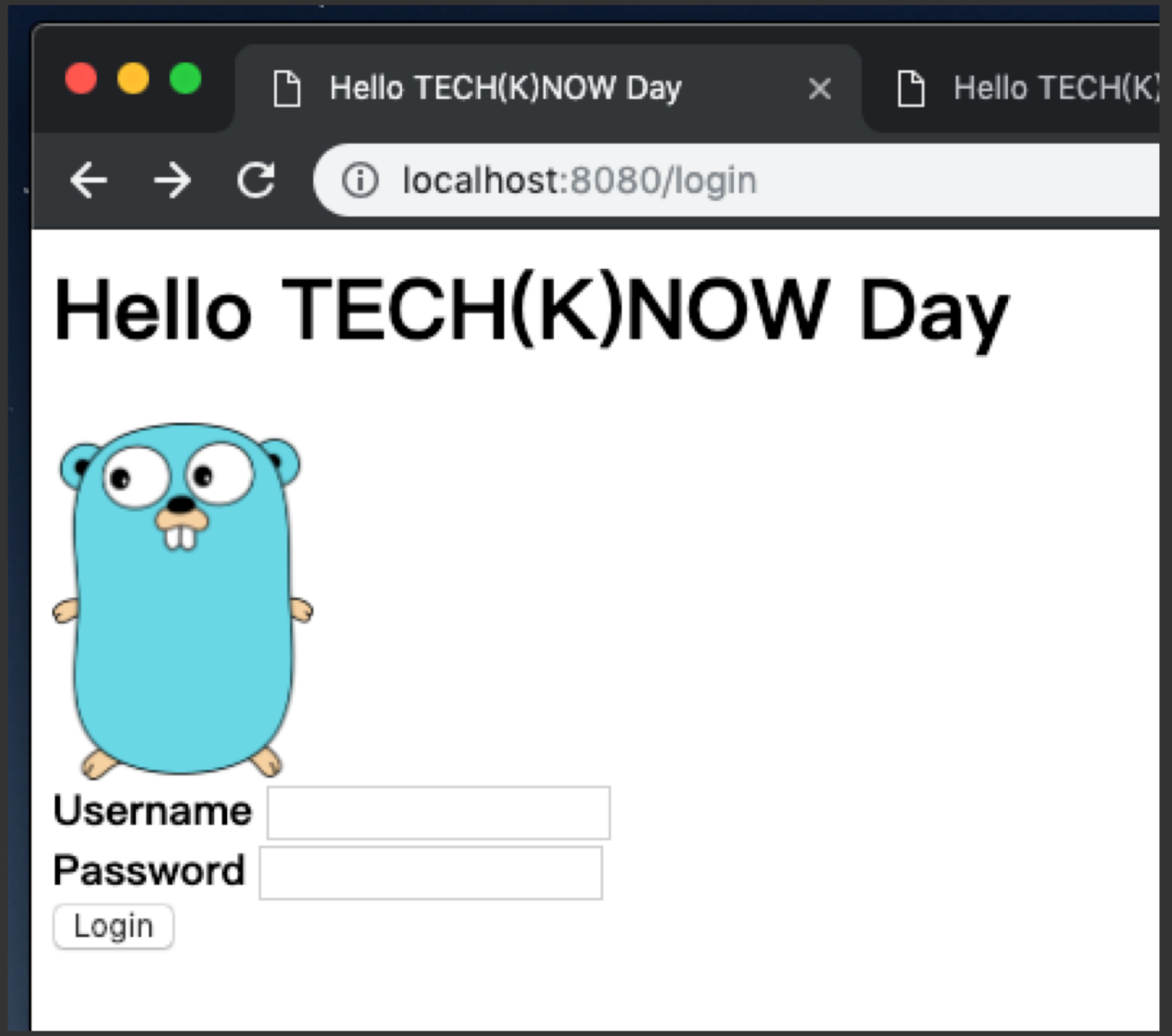
Header HTML Template

header.html.tmpl

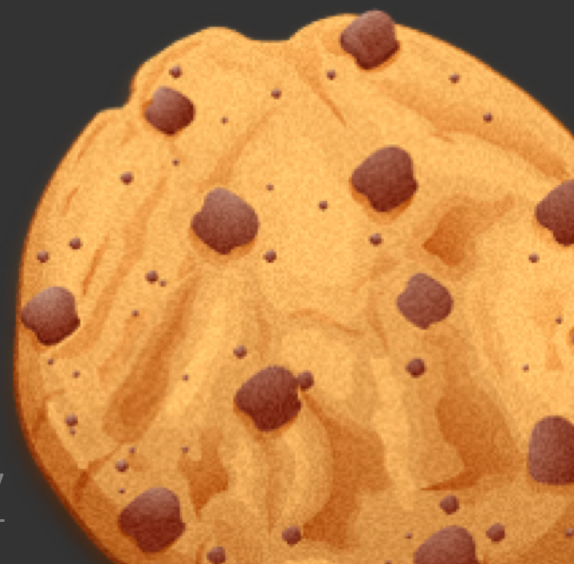
```
<!--header.html.tmpl-->
<html>
    <head>
        <title>Hello TECH(K)NOW Day</title>
    </head>
    <body>
        <h1>{{.Title}}</h1>
        <img src="/assets/images/gopher.png" width="100">
```

login.html.tmpl

```
{{ template "header.html.tmpl" .}}
        <form action="login" method="POST">
            <div>
```


Hello TECH(K)NOW Day
Hello TECH(K)
localhost:8080/login
Hello TECH(K)NOW Day
Username
Password
Login

Cookie

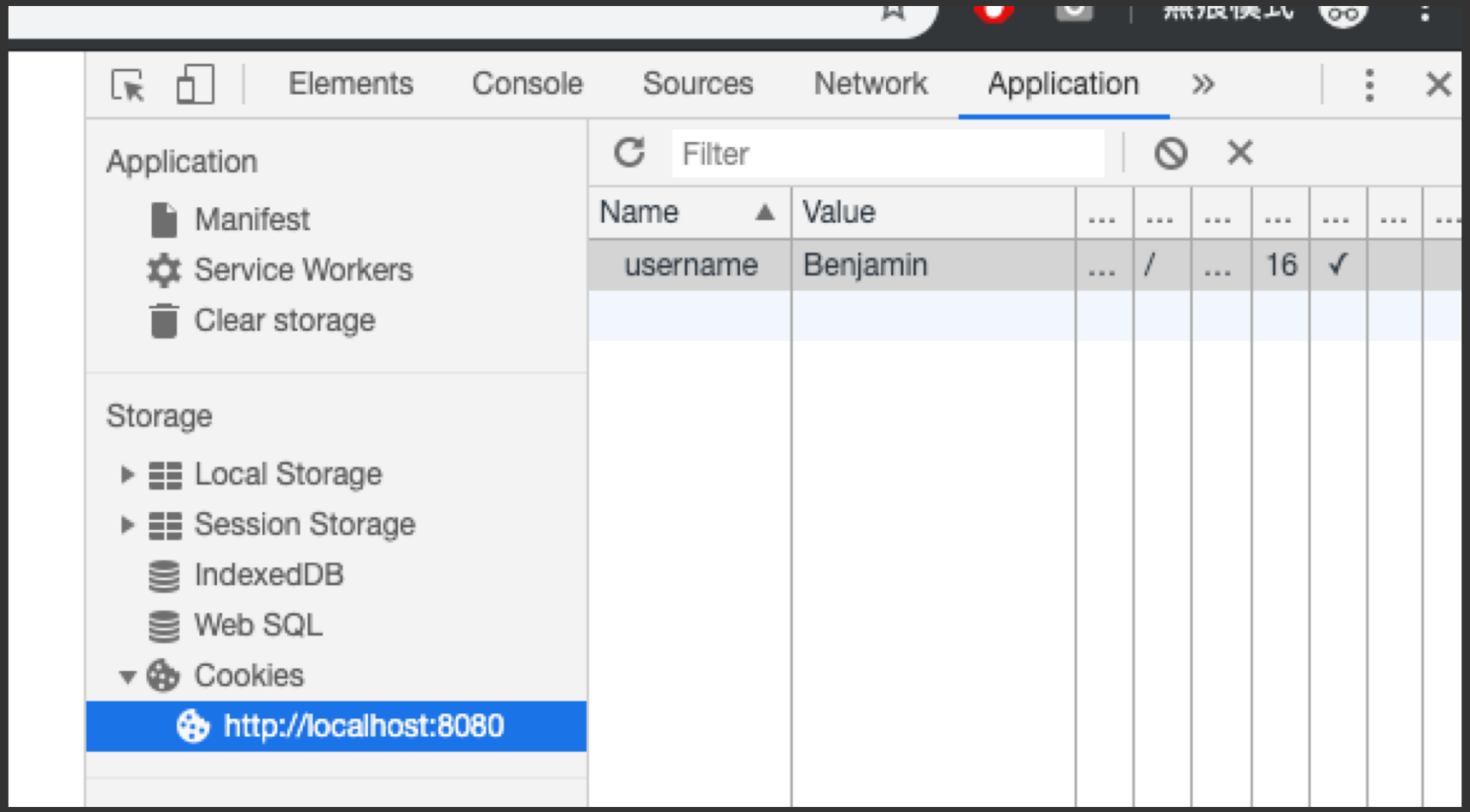


http://pngimg.com/imgs/food/cookie/

```
func (c *Context) SetCookie(name, value string, maxAge
int, path, domain string, secure, httpOnly bool)
```

name:	cookie key
value:	cookie value
maxAge:	live time
path:	available path
domain:	available domain
secure:	only https?
httpOnly:	only http and https?

```
if username == "Benjamin" && password == "1234" {
    c.SetCookie("username", username, 60, "/", "localhost", false, true)
    c.String(http.StatusOK, "Hi "+username)
```


Elements
Console
Sources
Network
Application
Application
Manifest
Service Workers
Clear storage
Storage
Local Storage
Session Storage
IndexedDB
Web SQL
Cookies
http://localhost:8080
Filter
Name
Value
username
Benjamin
/
16

Socket

Chat

16:23:36 <Benjamin> Hi how r u
16:23:43 <Guest278> good!

say something

```
$ go get -u gopkg.in/olahol/melody.v1
```

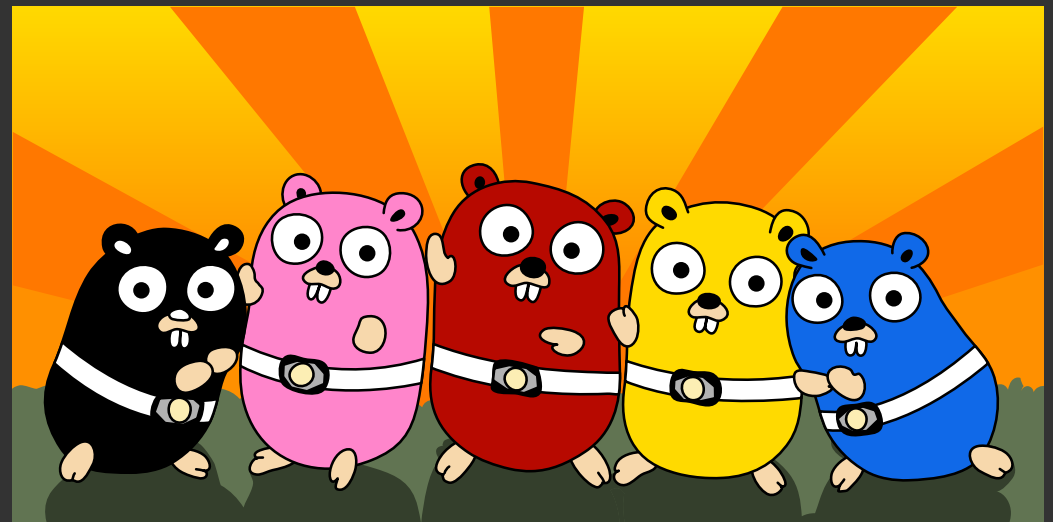
```
import (
    "fmt"
    "log"
    "net/http"

    "github.com/gin-gonic/gin"
    melody "gopkg.in/olahol/melody.v1"
)
```

```go
r := gin.Default()
m := melody.New()

r.GET("/chat", chat)

r.GET("/ws", func(c *gin.Context) {
    m.HandleRequest(c.Writer, c.Request)
})

m.HandleMessage(func(s *melody.Session, msg []byte) {
    m.Broadcast(msg)
})
```

```
func chat(c *gin.Context) {
    username, _ := c.Cookie("username")
    c.HTML(http.StatusOK, "chat.html.tmpl", gin.H{
        "username": username,
    })
}
```


GO

Q & A




syxuan


bingsyuan


syxuan


syxuann@gmail.com