web

web applications

very simple to implement

self-hosted, single binary

very efficient (e.g. built-in HTTP/2)

all aspects covered

- routing
- static files
- OAuth2/OpenIDC
- ….

hello web app

```
package main

import (
  "fmt"
  "net/http"
)

func handler(w http.ResponseWriter, r *http.Request) {
    fmt.Fprintf(w, "Hello from %v.", r.URL.Path[1:])
}

func main() {
    http.ListenAndServe(":8080", http.HandlerFunc(handler))
}
```

routes

```
http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
    fmt.Fprintf(w, "Hello")
})

http.HandleFunc("/welcome", func(w http.ResponseWriter, r *http.Request) {
    fmt.Fprintf(w, "Welcome")
})

http.ListenAndServe(":8080", nil)
```

json

```
import "encoding/json"

type Person struct {
        Name string `json:"name"`
        age int `json:"age"` // private! won't serialize
}

// GET
p := Person{"Gopher", 4}
data, _ := json.Marshal(p) w.Write(data)

// POST
body, err := ioutil.ReadAll(r.Body)
var p Person
err := json.Unmarshal(body, &p)
```

templates

```
type Data struct { Path string }

http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
    t, err := template.ParseFiles("hello.html")
    if err != nil {
        w.WriteHeader(http.StatusInternalServerError)
        fmt.Fprintf(w, "Template error")
        return
    }

    t.Execute(w, &Data{r.URL.Path[1:]})
})
```

static files

```
http.HandleFunc("/static/", func(w http.ResponseWriter, r *http.Request) {
    http.ServeFile(w, r, r.URL.Path[1:])
})
```

or use e.g. bindata to embed data in binary
https://github.com/jteeuwen/go-bindata

encodes files as binary strings

```
http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
    info, _ := AssetInfo(r.URL.Path[1:])
    file, _ := Asset(fp)
    http.ServeContent(wr, hr, info.Name(), info.ModTime(),
bytes.NewReader(file))
})
```

gorilla mux

https://github.com/gorilla/mux

```
r := mux.NewRouter()
r.HandleFunc("/hello/{name}", func(w http.ResponseWriter, r *http.Request) {
    name := mux.Vars(r)["name"]
    fmt.Fprintf(w, "Welcome %v!", name)
}).Methods("GET")

http.Handle("/", r)
```

session 5 – web

5.1 – Address Book Web Service

- handlers
- serialization