

Анатомия веб-сервиса

- Роутинг

`/users/123`

`-> func GetUser(...) { ... }`

```
router.HandleFunc(  
    "/users/{id:[0-9]+}",  
    GetUser,  
)
```

Анатомия веб-сервиса

- Авторизация / аутентификация

```
> GET /secret_url HTTP/1.1
> Host: localhost:8081
> Accept: */*
> Content-Type: application/json
> X-Access-Token: 695a19177a2760MzY3

< HTTP/1.1 401 Unauthorized
< Content-Type: application/json
{"error":"invalid token"}
```

Анатомия веб-сервиса

- Логирование

```
{  
  "level": "info",  
  "ts": 1512040115.3547573,  
  "caller": "logging/main.go:70",  
  "msg": "starting server",  
  "logger": "ZAP",  
  "host": "localhost",  
  "port": 8080  
}
```

Анатомия веб-сервиса

- Обработка ошибок

```
// не делайте так! :)  
if err != nil {  
    panic(err)  
}
```

Анатомия веб-сервиса

- Валидация входных параметров

```
> GET /post?id='ddfr'
```

```
< {"error": "id invalid, want int"}
```

```
> GET /post?id=123
```

```
    post := new(Post)
```

```
    schema.Parse(&post, r.URL)
```

```
< {"post":{"id":123}}
```

Анатомия веб-сервиса

- Шаблонизация

```
<%: func Index(...) %>
```

```
<%~ "base.html" %>
```

```
<%@ body { %>
```

```
<% } %>
```

Анатомия веб-сервиса

- Конфигурирование

```
appname = user
```

```
httpport = 8080
```

```
runmode = dev
```

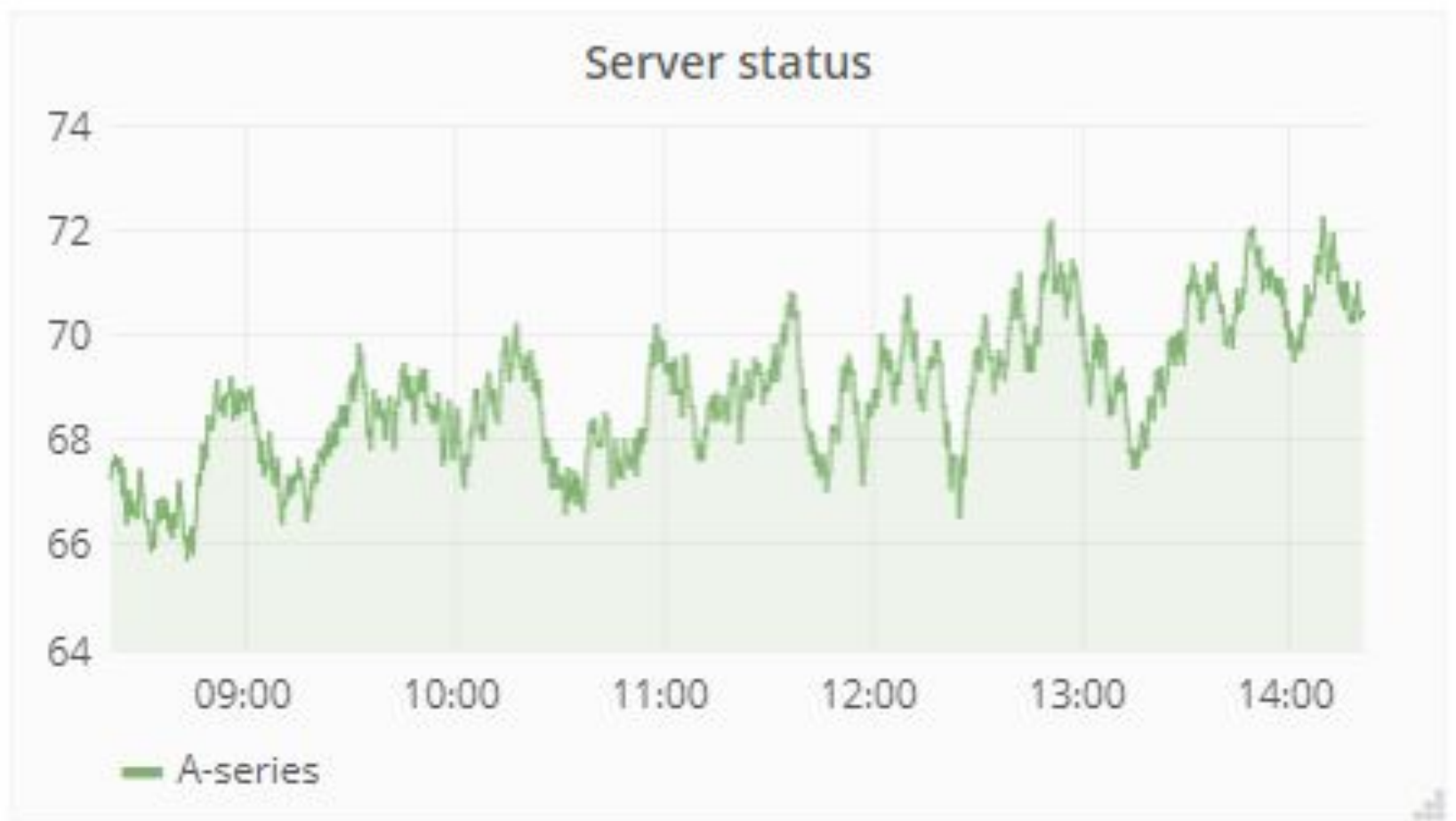
```
autorender = false
```

```
copyrequestbody = true
```

```
EnableDocs = true
```

Анатомия веб-сервиса

- Мониторинг



Анатомия веб-сервиса

- Middleware / фреймворки

```
Site := myServeMux
```

```
Site = checkAuth(Handler)
```

```
Site = accessLog(Handler)
```

```
Site = avoidPanic(Handler)
```

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

Анатомия веб-сервиса

- Работа с БД и хранилищами

```
SELECT COUNT(*), category_id  
FROM products  
WHERE status = 1  
GROUP BY category_id
```