GeekBrains

Выбор роутера и middleware

Backend-разработка на Go. Уровень 1

Что будет на уроке

1. Посмотрим на код нескольких роутеров
2. Поговорим про Middleware и контекст http-запроса
3. Посмотрим код клиента и сервера, сгенерированный swagger’ом

Резюме прошлого урока

- Для обработки http-запросов нам нужен хэндлер (роутер):

```
type Handler interface {

    ServeHTTP(ResponseWriter, *Request)

}
```

- Этот хэндлер мы задаем при создании сервера
- Базового роутера из пакета http может не хватить

Смотрим роутеры от сообщества

- gorilla/mux
- httprouter
- fasthttp
- Из фреймворка gin

Middleware

- При работе с запросами в рамках функций `func(w http.ResponseWriter, r *http.Request)` нам иногда приходится производить повторяющиеся операции
- Если такие операции нужно выполнять для всех функций, входящих в роутер, почему бы не вынести эту логику на уровень роутера в метод `ServeHTTP(ResponseWriter, *Request)`?
- Именно эту проблему и решает Middleware!

Middleware

Формат: `MyHandler(something, .., h http.Handler) http.Handler`

- Middleware принимает на вход и возвращает http.Handler
- Middleware - “обертка” над хэндлером

Пример из gorilla/handlers: LoggerHandler - обертка для логирования запросов

Контекст запроса

При написании middleware может возникнуть потребность изменить запрос или ответ.

В частности, в запрос бывает нужно добавить дополнительные данные.

Для того, чтобы сделать это, часто работают с контекстом запроса. Чтобы обновить его, вызывают функцию WithContext, чтобы прочитать - Context.

Контекст запроса

При написании middleware может возникнуть потребность изменить запрос или ответ.

В частности, в запрос бывает нужно добавить дополнительные данные.

Для того, чтобы сделать это, часто работают с контекстом запроса. Чтобы обновить его, вызывают функцию WithContext, чтобы прочитать - Context.

С самим типом context.Context вы уже знакомы из предыдущих курсов.

Смотрим библиотеки middleware от сообщества

- Пример реализации middleware: gorilla/handlers
- Еще один пример: vulcand/oxy
- Цепочки middleware: https://github.com/justinas/alice/
- Больше примеров: https://github.com/avelino/awesome-go#middlewares

Кодогенерация

- Рассмотри код сервера и клиента, сгенерированные swagger’ом
- Еще пример генерилки кода: https://github.com/takama/caldera
- Больше бойлерплейтов: https://github.com/avelino/awesome-go#project-layout
- Геннерить OpenAPI из аннотаций кода тоже можно: https://github.com/swaggo/swag

Домашка

Выбрать роутер или генератор кода для реализации курсового проекта. Обосновать свой выбор.

Вопросы
и
Ответы