

[◀ Назад к неделе 2](#)[✕ Уроки](#)[Пред.](#)[Дальше](#)

Задание по программированию: DB Explorer

Вы не отправили работу. Для успешной сдачи вам необходимо набрать 1/1 баллов.

Срок сдачи Сдайте это задание до March 11, 11:59 PM PDT

Инструкции

[Моя работа](#)

[Обсуждения](#)

hw6_db_explorer.zip

Программа db_explorer

Эта простой веб-сервис будет представлять собой менеджер MySQL-базы данных, который позволяет осуществлять CRUD-запросы (create, read, update, delete) к ней по HTTP

В данном задании мы продолжаем отработку навыков работы с HTTP и взаимодействуем с базой данных.

Для пользователя это выглядит так:

- GET / - возвращает список все таблиц (которые мы можем использовать в дальнейших запросах)
- GET /\$table?limit=5&offset=7 - возвращает список из 5 записей (limit) начиная с 7-й (offset) из таблицы \$table. limit по-умолчанию 5, offset 0
- GET /\$table/\$id - возвращает информацию о самой записи или 404
- PUT /\$table - создаёт новую запись, данные по записи в теле запроса (POST-параметры)
- POST /\$table/\$id - обновляет запись, данные приходят в теле запроса (POST-параметры)
- DELETE /\$table/\$id - удаляет запись

GET, PUT, POST, DELETE - это http-метод, которым был отправлен запрос

Особенности работы программы:

1. Роутинг запросов - руками, никаких внешних библиотек использовать нельзя.
2. Полная динамика. при инициализации в NewDbExplorer считываем из базы список таблиц, полей (запросы ниже), далее работаем с ними при валидации. Никакого хадкода в виде кучи условий и написанного кода для валидации-заполнения. Если добавить третью таблицу - всё должно работать для неё.
3. Считаем что во время работы программы список таблиц не меняется
4. Запросы придётся конструировать динамически, данные оттуда доставать тоже динамически - у вас нет фиксированного списка параметров - вы его подгружаете при инициализации.
5. Валидация на уровне "string - int - float - null", без заморочек. Помните, что json в пустой итнтерфейс распаковывает как float, если не указаны спец. опции.
6. Вся работа происходит через database/sql, вам на вход передаётся рабочее подключение к базе. Никаких orm и прочего.
7. Все имена полей так как они в базе.
8. В случае если возникает ошибка - просто возвращаем 500 в http-статусе
9. Не забывайте про SQL-инъекции
10. Неизвестные поля игнорируем
11. **В этом задании запрещено использование глобальных переменных. Всё что вы хотите хранить - храните в полях структуры**

Запросы вам в помощь для получения списка таблицы и их структуры:

- SHOW TABLES;
- SHOW FULL COLUMNS FROM `table_name`;

Подсказки:

- Внутри row, который вы получаете из базы лежат не только сами значения полей, но и метаданные - <https://golang.org/pkg/database/sql/#Rows.ColumnTypes>
- Тут будут активно применяться пустые интерфейсы
- Обратите внимание на обработку null-значения
- Придётся вытаскивать неизвестное количество полей из row, подумайте как тут можно применить пустые интерфейсы

Поднять mysql-базу локально проще всего через докер:

- `docker run -p 3306:3306 -v $(PWD):/docker-entrypoint-initdb.d -e MYSQL_ROOT_PASSWORD=1234 -e MYSQL_DATABASE=golang -d mysql`

How to submit

When you're ready to submit, you can upload files for each part of the assignment on the "My submission" tab.

