

ТЗ - Создание и демонстрация GraphQL

БД

[Создать таблицы БД](#)

[Заполнить таблицы БД](#)

GRAPHQL

[Создать Query "getProduct" \(товар\)](#)

[Создать Mutation "addProduct" \(добавить Товар\)](#)

[Создать Mutation "deleteProduct" \(удалить Товар\)](#)

ДЕМО

БД

Создать таблицы БД

Создать таблицу “products” (товары).

Поля:

- id (id товара) INT
- product_name (название товара) TEXT

Заполнить таблицы БД

Заполнить таблицу “products” (товары).

Поля:

- id = 1, product_name = "Мяч футбольный Classic"
- id = 2, product_name = "Баскетбольный мяч №3"
- id = 3, product_name = "Велосипедный шлем синий"
- id = 4, product_name = "Турник для дома 120 кг"
- id = 5, product_name = "Гантели нейлоновые 2 кг"

GRAPHQL

Создать Query "getProduct" (товар)

Параметры

- id (id товара)

Поля

- id (id товара)
- productName (название товара)

Логика

1. Найти товар по id
2. Вывести поля товара

Пример запроса

```
query getProduct{  
  product(id: 1) {  
    id  
    productName  
  }  
}
```

```
}
```

Пример ответа

```
{  
  "id": 1,  
  "productName": "Мяч футбольный Classic",  
}
```

Создать Mutation "addProduct" (добавить Товар)

Параметры

- `productName` (id товара)

Поля

- `id` (id товара)

- `productName` (id товара)

Логика

1. Добавить товар в таблицу `products`.
2. В ответе вывести поля `id` (id товара) и `productName` (id товара)

Пример мутации (заполнены переменные)

```
mutation deleteProduct {  
  deleteProduct(productName: "Мяч волейбольный") {  
    id  
    productName  
  }  
}
```

Пример ответа мутации

```
{  
  "id": "2"  
  "id": "Мяч волейбольный"  
}
```

Создать Mutation "deleteProduct" (удалить Товар)

Параметры

-- `id` (id товара)

Поля

- `id` (id товара)

Логика

3. Удалить товар.
4. В ответе вывести id удаленного товара.
5. Если ошибка, вывести информацию об ошибке.
 - а. Несуществующий id товара.

Пример мутации (заполнены переменные)

```
mutation deleteProduct {  
  deleteProduct(id: 10) {  
    id  
  }  
}
```

Пример ответа мутации

```
{  
  "id": "1"  
}
```

ДЕМО

Продемонстрировать

1. Как установить

- Webonyx

- GraphQL.

- GraphQL Doc.

- * “GraphQL” и “GraphQL Doc” должны работать с созданными запросами и мутациями.

- * Уже установлены: Open Server, Composer.

2. Как создаются запросы и мутации GraphQL описанные в ТЗ.

- * стек: PHP, MySQL.

- * Возможна запись видео с экрана со звуком.

- * Может потребоваться созвон с демонстрацией удаленного рабочего стола.