

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Домашняя работа №4-5

Выполнил:

Буданцев Артём

К3333

Проверил:

Добряков Д. И.

Санкт-Петербург

2023 г.

Задача

Документирование и тестирование API средствами Postman из ЛР 2 (сайт для администратора интернет-магазина)

Ход работы

Рассмотрим реализации документирования и тестирования на примере API goods:

The screenshot shows the Postman interface for an API named "good API". The selected endpoint is "GET Get goods" with the URL "http://localhost:8000/good". The "Example Request" tab shows a curl command: `curl --location 'http://localhost:8000/good'`. The "Example Response" tab shows a JSON array of goods, with the first item expanded to show details like "id", "name", "count", "price", "createdAt", and "updatedAt". The status is "200 OK".

```
curl --location 'http://localhost:8000/good'
```

```
[
  {
    "id": 1,
    "name": "Суп",
    "count": 0,
    "price": 500,
    "createdAt": "2023-04-29T07:14:15.287Z",
    "updatedAt": "2023-04-29T07:20:00.520Z"
  },
  {
    "id": 2,
    "name": "Суп",
    "count": 0,
    "price": 500,
    "createdAt": "2023-04-29T07:14:15.287Z",
    "updatedAt": "2023-04-29T07:20:00.520Z"
  }
]
```

The screenshot shows the Postman interface for the same "good API". The selected endpoint is "GET Get good by ID" with the URL "http://localhost:8000/good/1". The "Example Request" tab shows a curl command: `curl --location 'http://localhost:8000/good/1'`. The "Example Response" tab shows a JSON object representing a single good item, with the same fields as the first item in the previous response. The status is "200 OK".

```
curl --location 'http://localhost:8000/good/1'
```

```
{
  "id": 1,
  "name": "Суп",
  "count": 0,
  "price": 500,
  "createdAt": "2023-04-29T07:14:15.287Z",
  "updatedAt": "2023-04-29T07:20:00.520Z"
}
```

POST Create good

http://localhost:8000/good

Body raw (json)

```
json
{
  "name": "Диван",
  "count": 4,
  "price": 2000
}
```

Example Request

```
curl --location 'http://localhost:8000/good' \
--data '{
  "name": "Диван",
  "count": 4,
  "price": 2000
}'
```

Example Response

Body Headers (7) 200 OK

```
json
{
  "id": 3,
  "name": "Диван",
  "count": 4,
  "price": 2000,
  "updatedAt": "2023-04-30T14:39:41.053Z",
  "createdAt": "2023-04-30T14:39:41.053Z"
}
```

PATCH Update good

http://localhost:8000/good/3

Body raw (json)

```
json
{
  "name": "Диван",
  "count": 5,
  "price": 2000
}
```

Example Request

```
curl --location --request PATCH 'http://localhost:8000/good/3' \
--data '{
  "name": "Диван",
  "count": 5,
  "price": 2000
}'
```

Example Response

Body Headers (7) 200 OK

```
json
{
  "id": 3,
  "name": "Диван",
  "count": 5,
  "price": 2000,
  "createdAt": "2023-04-30T14:39:41.053Z",
  "updatedAt": "2023-04-30T14:41:25.862Z"
}
```

DELETE Delete good

http://localhost:8000/good/3

Example Request

```
curl --location --request DELETE 'http://localhost:8000/good/3'
```

Example Response

Body Headers (4) 204 NO CONTENT

No response body
This request doesn't return any response body

Ссылка на полную документацию в Postman:
<https://documenter.getpostman.com/view/24344181/2s93eSaFbq>

Вывод

В результате работы создана документация API в Postman с примерами запросов.