**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»  
(БГТУ им. В.Г.Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №5  
дисциплина: Технологии Web-программирования  
тема: «REST API»

Выполнила: ст. группы ПВ-41  
Щетинина П.Г.  
Проверил: Картамышев С.В.

Белгород 2020

**Цель:** изучить основы разработки API для web-приложений. Разработать REST API для своего проекта.

**Задание к лабораторной работе:**

1. Изучить структуру формата представления данных JSON.

2. Изучить типы запросов к API: HEAD, GET, POST, PUT, DELETE.

3. Спроектировать и реализовать собственное REST API (Получение, создание, изменение и удаление каких-либо объектов).

4. В отчёт необходимо предоставить документацию к использованию методов. (Либо словесным описание, либо через Swagger)

**Выполнение работы**

Словесное описание методов, реализованных в REST API:

Базовый URL: localhost:1199/api/v1

**Посты**

1. Получение всех постов:

URL: /post/all

Тип запроса: GET

Параметры: нет

1. Получение конкретного продукта по его id:

URL: /post/id

Тип запроса: GET

Параметры: id – id поста, integer, обязательный

1. Получение постов за определенную дату:

URL: /post/idauthor

Тип запроса: GET

Параметры: idAuthor – id авторв поста, integer, обязательный

1. Получение постов определенного автора:

URL: /product/iddata

Тип запроса: GET

Параметры: idData – id дата поста, integer, обязательный

1. Создание поста

URL: /post/create

Тип запроса: POST

Параметры: в теле запроса принимается объект продукта с полями: name (обязательный, string, название продукта), url(обязательный, string, уникальное имя поста в командной строке браузера) iamge (обязательный, string, ссылка на картинку поста), author (обязательный, string, имя автора), data (обязательный, string, дата публикации поста), description (обязательный, txt, описание поста), idAuthor (обязательный, integer, имя автора), idDate(обязательный, id, дата публикации)

**Даты постов**

1. Получение всех дат:

URL: /date/all

Тип запроса: GET

Параметры: нет

1. Получение определенной даты:

URL: /date/id

Тип запроса: GET

Параметры: id – id категории, integer, обязательный

1. Создание даты:

URL: /date/create

Тип запроса: POST

Параметры: в теле запроса принимается объект создаваемой категории с полем name (обязательный, string, название категории)

**Авторы постов**

1. Получение всех авторов:

URL: /author/all

Тип запроса: GET

Параметры: нет

1. Получение постов определенного автора:

URL: /author/id

Тип запроса: GET

Параметры: id – id начинки, integer, обязательный

1. Создание нового автора:

URL: /author/create

Тип запроса: POST

Параметры: в теле запроса принимается объект создаваемой начинки с полем name (обязательный, string, название начинки)

**Пользователи**

1. Получение всех пользователей:

URL: /user/all

Тип запроса: GET

Параметры: нет

1. Вход пользователя:

URL: /user/login

Тип запроса: GET

Параметры: username – имя пользователя, string, обязательный; password – пароль пользователя, string, обязательный

1. Создание пользователя:

URL: /user/register

Тип запроса: POST

Параметры: username – имя пользователя, string, обязательный; password – пароль пользователя, string, обязательный

**Избранное**

1. Получение списка папок избранного:

URL: /favorites/all

Тип запроса: GET

Параметры: нет

1. Получение конкретной папки по его id:

URL: / favorites /id

Тип запроса: GET

Параметры: id – id продукта, integer, обязательный

1. Получение заказов определенного пользователя:

URL: / favorites /iduser

Тип запроса: GET

Параметры: idUser – id пользователя, integer, обязательный

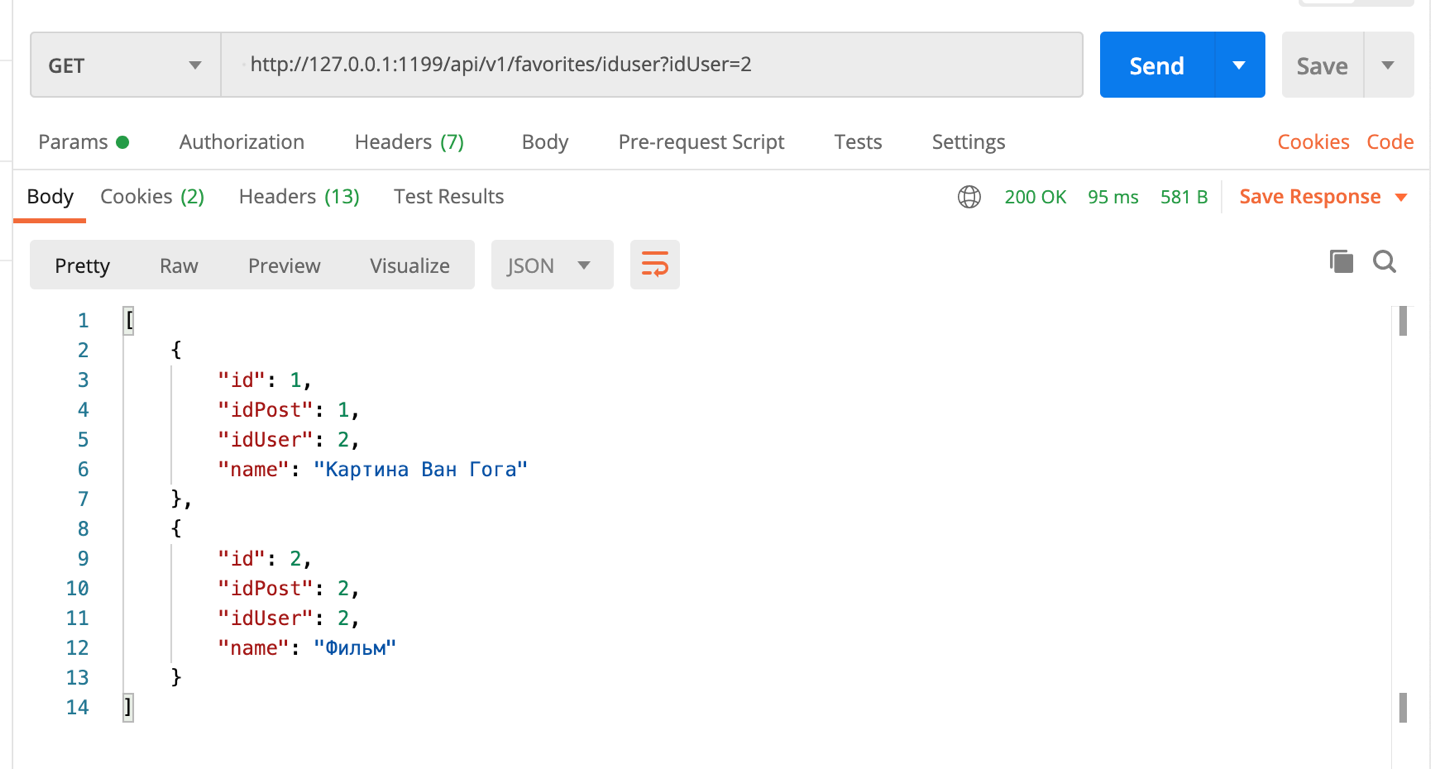
1. Создание новой папки избранного:

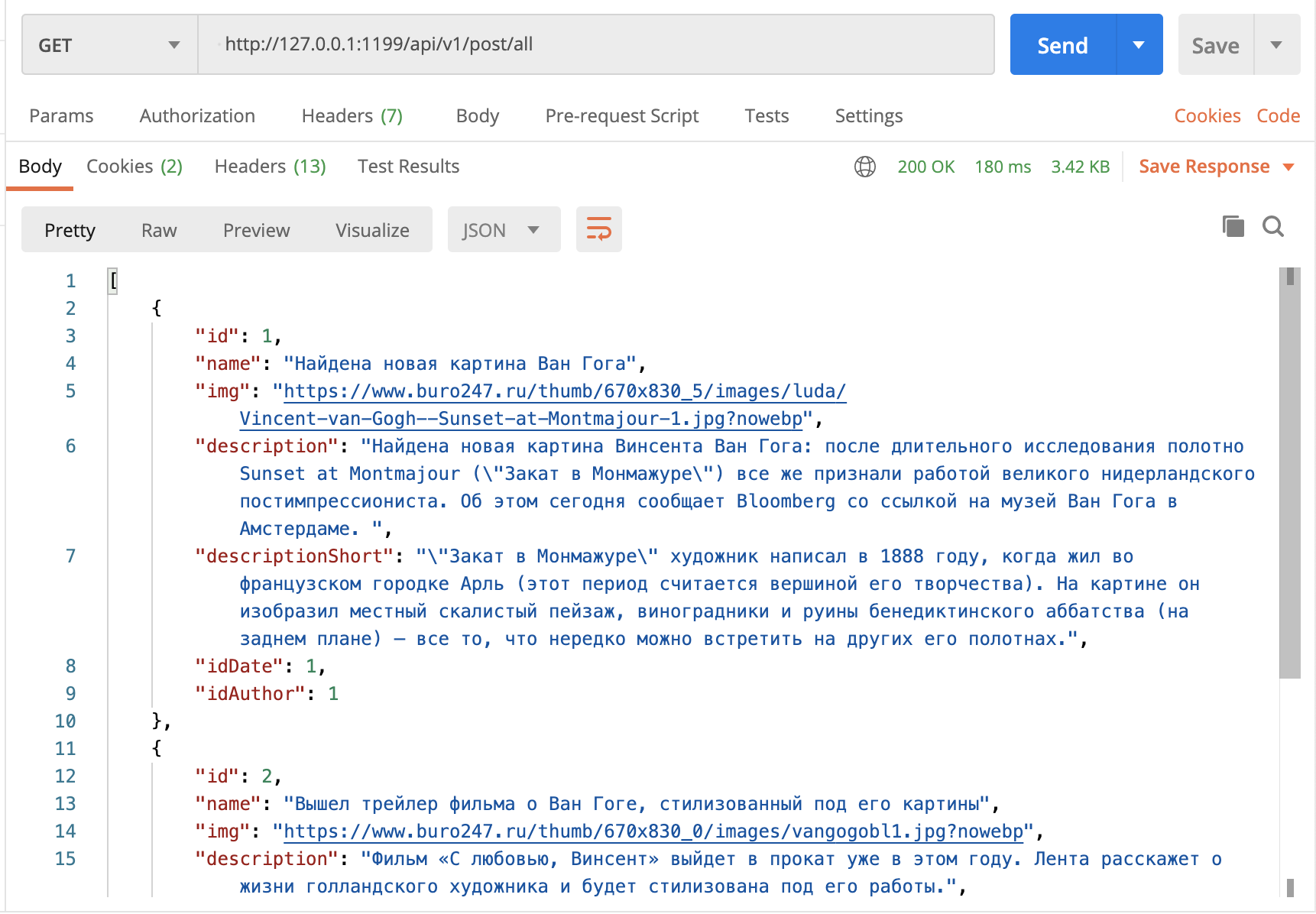
URL: / favorites /create

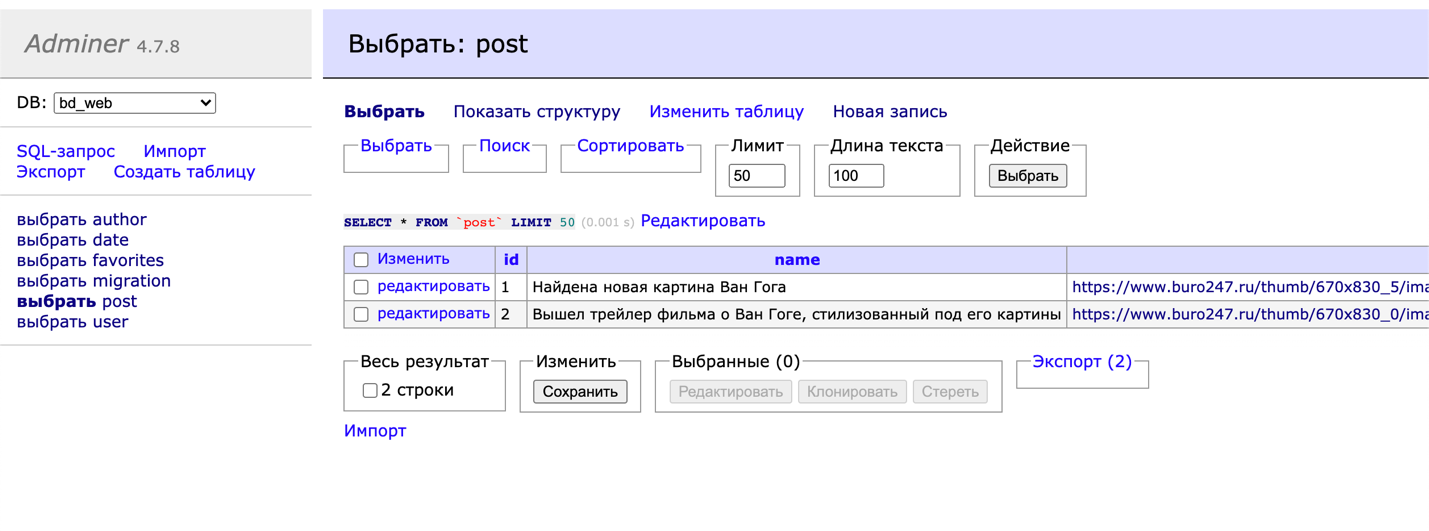
Тип запроса: POST

Параметры: в теле запроса принимается объект заказа с полями: idFavorites (обязательный, integer, продукт в заказе), idUser (обязательный, integer, пользователь, сделавший заказ), name (обязательный, название, string).

Пример выполнения некоторых запросов в postman:







**Вывод:** в ходе выполнения лабораторной работы изучили основы разработки API для web-приложений, разработали REST API для своего проекта.