**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  
Немшилова А.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

**Продукты**

1. Получение всех продуктов:

URL: /product/all

Тип запроса: GET

Параметры: нет

1. Получение конкретного продукта по его id:

URL: /product/id

Тип запроса: GET

Параметры: id – id продукта, integer, обязательный

1. Получение продуктов определенной категории:

URL: /product/idcategory

Тип запроса: GET

Параметры: idCategory – id категории продукта, integer, обязательный

1. Получение продуктов с определенной начинкой:

URL: /product/idfilling

Тип запроса: GET

Параметры: idFilling – id начинка продукта, integer, обязательный

1. Создание продукта

URL: /product/create

Тип запроса: POST

Параметры: в теле запроса принимается объект продукта с полями: name (обязательный, string, название продукта), img (обязательный, string, ссылка на картинку продукта), price (обязательный, float, цена продукта), description (необязательный, string, описание продукта), weight (обязательный, float, вес продукта), idCategory (обязательный, integer, категория продукта), idFilling (обязательный, id, начинка продукта).

**Категории продуктов**

1. Получение всех категорий:

URL: /category/all

Тип запроса: GET

Параметры: нет

1. Получение определенной категории:

URL: /category/id

Тип запроса: GET

Параметры: id – id категории, integer, обязательный

1. Создание категории:

URL: /category/create

Тип запроса: POST

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

**Начинки продуктов**

1. Получение всех начинок:

URL: /filling/all

Тип запроса: GET

Параметры: нет

1. Получение определенной начинки:

URL: /filling/id

Тип запроса: GET

Параметры: id – id начинки, integer, обязательный

1. Создание категории:

URL: /filling/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

**Заказы**

1. Получение всех заказов:

URL: /order/all

Тип запроса: GET

Параметры: нет

1. Получение конкретного заказа по его id:

URL: /order/id

Тип запроса: GET

Параметры: id – id продукта, integer, обязательный

1. Получение заказов определенного пользователя:

URL: /order/iduser

Тип запроса: GET

Параметры: idUser – id пользователя, integer, обязательный

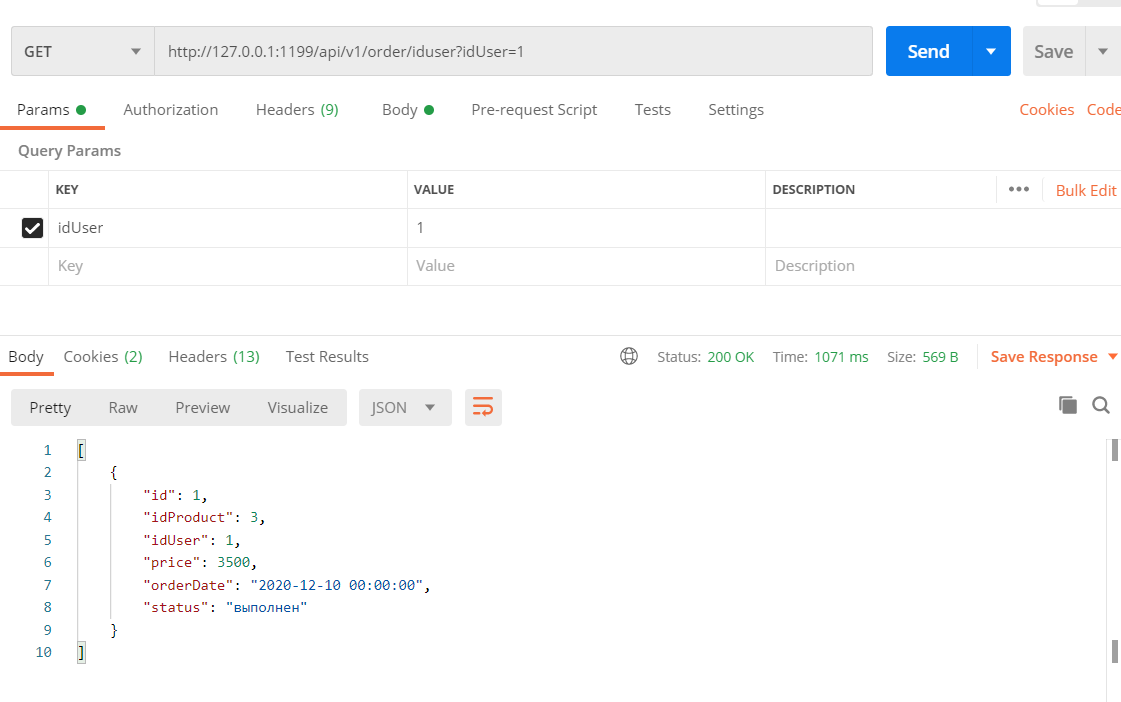
1. Создание заказа

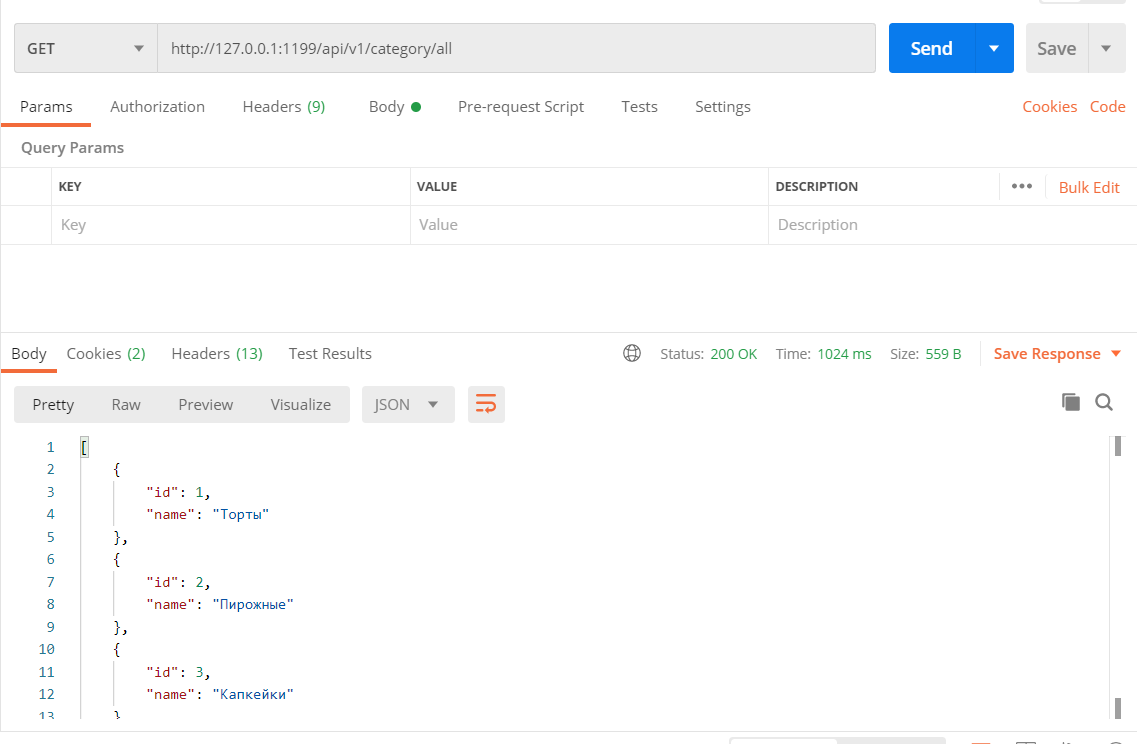
URL: /order/create

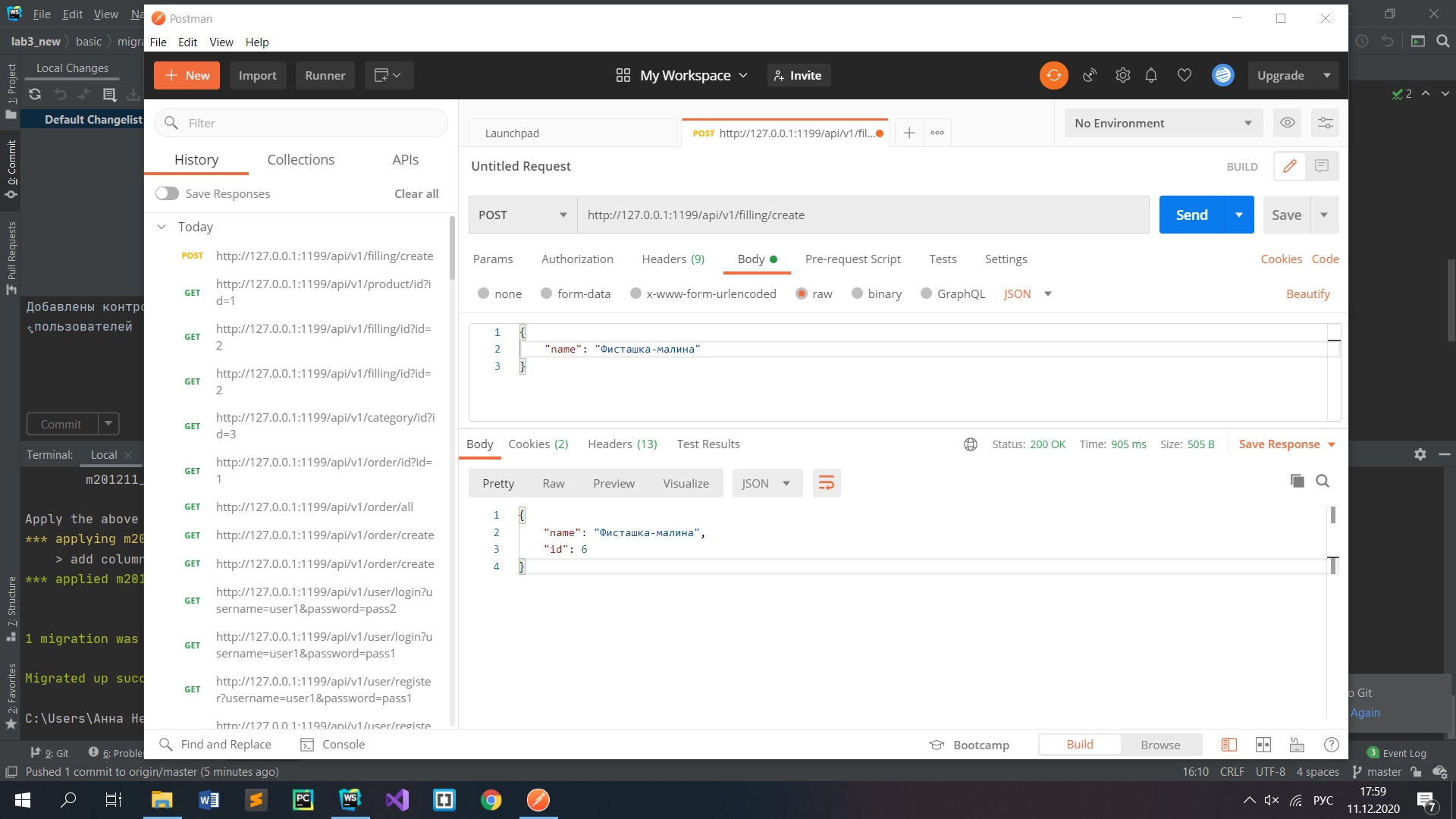
Тип запроса: POST

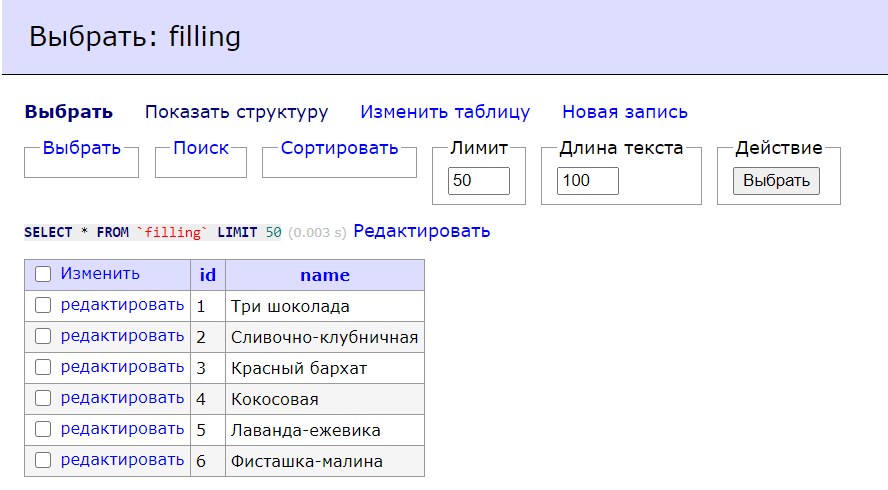
Параметры: в теле запроса принимается объект заказа с полями: idProduct (обязательный, integer, продукт в заказе), idUser (обязательный, integer, пользователь, сделавший заказ), price (обязательный, float, сумма заказа), orderDate (необязательный, string, дата заказа), status (обязательный, string, статус выполнения).

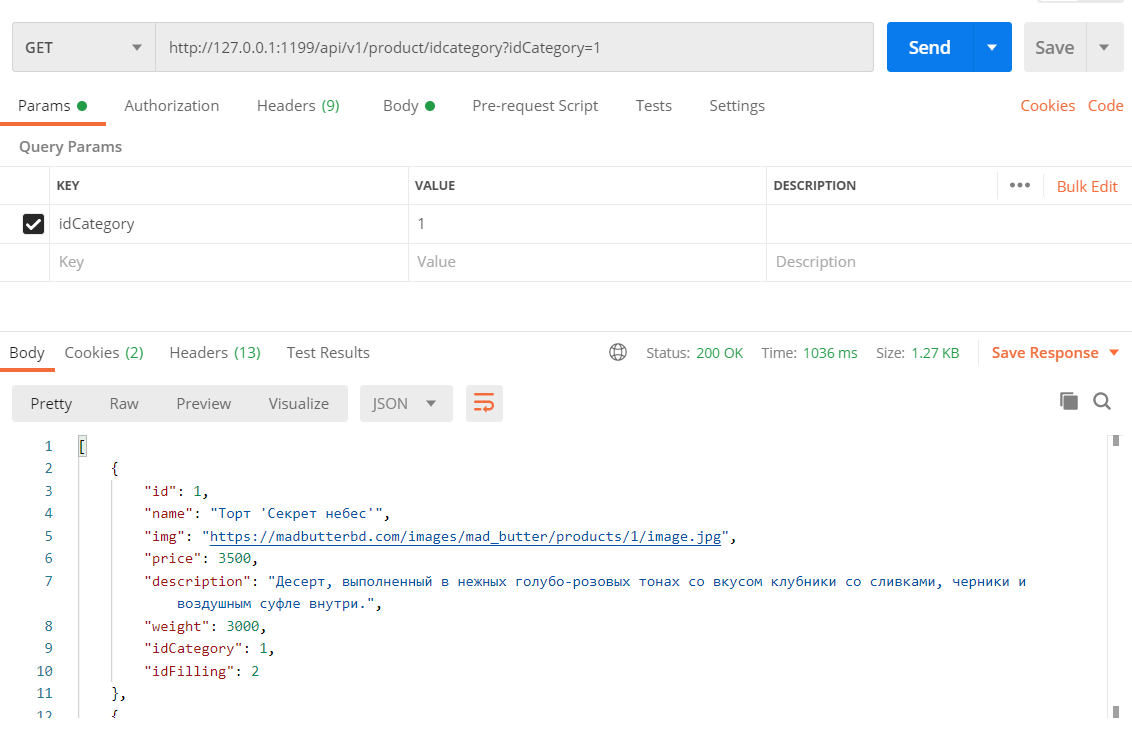
Пример выполнения некоторых запросов в postman:











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