**МИНОБРНАУКИ РОССИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧЕРЕЖДЕНИЕ 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

по дисциплине: Технологии Web-программирования

тема: «**REST API**»

Выполнил: ст. группы ПВ-41

Таран Алексей Николаевич

Проверил: Картамышев С.В.

Белгород 2020

**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)

**Ход работы**

Опишем API-методы для следующих классов:

1) Контроллер главной страницы

2) Город

3) Достопримечательности

4) Варианты проживания

5) Кухня

6) Способы передвижения

7) Пользователь

Базовый URL для всех методов: [*http://127.0.0.1:1199/api/v1/*](http://127.0.0.1:1199/api/v1/)

**Контроллер главной страницы**

*URL:* [*http://127.0.0.1:1199/api/v1/*maincontent](http://127.0.0.1:1199/api/v1/maincontent/)

Методы:

**1) Создание нового раздела главной страницы**

URL: /create

Тип запроса: POST

Параметры: в Body => name: String!, link: String!

*(ПРИМ. «!» соответствует обязательному параметру)*

**2) Получение названий разделов главной страницы**

URL: /names

Тип запроса: GET

Параметры: Нет

**3) Получение разделов главной страницы**

URL: /chapters

Тип запроса: GET

Параметры: Нет

**4) Получение заметок главной страницы**

URL: /notes

Тип запроса: GET

Параметры: Нет

**5) Получение текста информации главной страницы**

URL: /description

Тип запроса: GET

Параметры: Нет

**6) Получение фотографий главной страницы**

URL: /photo

Тип запроса: GET

Параметры: Нет

**7) Получение видео главной страницы**

URL: /getvideo

Тип запроса: GET

Параметры: Нет

**8) Изменение ссылки видео главной страницы**

URL: /addvideo

Тип запроса: GET

Параметры: link – строка, содержащая новую ссылку на видеоресурс

**Город**

*URL: [http://127.0.0.1:1199/api/v1/](http://127.0.0.1:1199/api/v1/city/)*[city/](http://127.0.0.1:1199/api/v1/city/)

Методы:

**1) Создание нового города**

URL: /create

Тип запроса: POST

Параметры: в Body => name: String!, link: String!

**2) Получение города по латинскому названию**

URL: /getbyname

Тип запроса: GET

Параметры: engname (латинское название города): String!

**3) Получение названий городов**

URL: /names

Тип запроса: GET

Параметры: Нет

**4) Получение названия города по id**

URL: /name

Тип запроса: GET

Параметры: id (идентификатор города): Int!

**5) Получение списка разделов города по id города**

URL: /chapters

Тип запроса: GET

Параметры: idCity (идентификатор города): Int!

**6) Получение списка заметок по id города**

URL: /getnotes

Тип запроса: GET

Параметры: idCity (идентификатор города): Int!

**7) Получение фотографий города по id города**

URL: /photo

Тип запроса: GET

Параметры: idCity (идентификатор города): Int!

**8) Получение главной фотографии города по id города**

URL: /mainphoto

Тип запроса: GET

Параметры: idCity (идентификатор города): Int!

**Достопримечательности**

*URL: [http://127.0.0.1:1199/api/v1/](http://127.0.0.1:1199/api/v1/attraction/)*[attraction/](http://127.0.0.1:1199/api/v1/attraction/)

Методы:

**1) Создание новой достопримечательности**

URL: /create

Тип запроса: POST

Параметры: в Body => name: String!

**2) Получение названий достопримечательностей**

URL: /names

Тип запроса: GET

Параметры: Нет

**3) Получение списка разделов достопримечательностей**

URL: /chapters

Тип запроса: GET

Параметры: Нет

**4) Получение заметки по названию достопримечательности**

URL: /note

Тип запроса: GET

Параметры: attractionname (Название достопримечательности): String!

**5) Получение фотографий достопримечательностей**

URL: /photo

Тип запроса: GET

Параметры: Нет

**6) Получение комментарий к странице достопримечательностей**

URL: /coments

Тип запроса: GET

Параметры: Нет

**7) Создание нового комментария**

URL: /new\_comment

Тип запроса: POST

Параметры: accessToken: String!, text: String!

**8) Редактирование существующего комментария**

URL: /edit\_comment

Тип запроса: POST

Параметры: accessToken: String!,

id (идентификатор комментария): String!,

text: String!

**9) Удаление комментария**

URL: /delete\_comment

Тип запроса: POST

Параметры: accessToken: String!,

id (идентификатор комментария): String!,

Для следующих 3 классов укажем лишь url, т.к суть методов для них схожа. А затем опишем методы.

**Варианты проживания**

*URL:* [*http://127.0.0.1:1199/api/v1/*housing/](http://127.0.0.1:1199/api/v1/housing/)

**Варианты передвижения**

*URL:* [*http://127.0.0.1:1199/api/v1/*trip/](http://127.0.0.1:1199/api/v1/trip/)

**Кухня**

*URL:* [*http://127.0.0.1:1199/api/v1/*kitchen/](http://127.0.0.1:1199/api/v1/kitchen/)

Методы для перечисленных 3 классов:

**1) Создание нового (**

**(Варианта проживания |**

**Передвижения|**

**Кухни)**

URL: /create

Тип запроса: POST

Параметры: в Body => name: String!

**2) Получение названий**

**(Варианта проживания |**

**Передвижения|**

**Кухни)**

URL: /names

Тип запроса: GET

Параметры: Нет

**3) Получение списка разделов**

**(Варианта проживания |**

**Передвижения|**

**Кухни)**

URL: /chapters

Тип запроса: GET

Параметры: Нет

**4) Получение заметки по названию**

**(Варианта проживания |**

**Передвижения|**

**Кухни)**

URL: /notes

Тип запроса: GET

Параметры: housing (Название варианта проживания): String! |

trip (Название варианта передвижения): String! |

kitchen (Название раздела кухни): String!

**5) Получение фотографий**

**(Варианта проживания |**

**Передвижения|**

**Кухни)**

URL: /photo

Тип запроса: GET

Параметры: Нет

**6) Получение главной фотографии страницы**

**(Варианта проживания |**

**Передвижения|**

**Кухни)**

URL: /mainphoto

Тип запроса: GET

Параметры: Нет

**Пользователь**

*URL:* [*http://127.0.0.1:1199/api/v1/*user/](http://127.0.0.1:1199/api/v1/user/)

Методы:

**1) Получение списка всех пользователей**

URL: /all

Тип запроса: GET

Параметры: Нет

**2) Получение access token для входа в систему пользователя**

URL: /signin, /signup

Тип запроса: POST

Параметры: login: String!,

password: String!

**3) Выход пользователя из системы**

URL: /exit

Тип запроса: POST

Параметры: accessToken: String!,