

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
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  
по дисциплине: Технологии web-программирования  
тема: «**REST API**»»

Выполнила: ст. группы ПВ-42  
Петрова Алина Ильинична  
Проверил: Картамышев С.В.

Белгород 2021 г.

**Цель работы:** 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 описанием, либо через Swagger)

# Выполнение

## Описание API

### ToDo List 1.0 OAuth2

/openapi.json

API

Authorize 

#### Пользователи

GET	/users/permissions/	Get Users	✓	🔒
GET	/users/permissions/me	Get Me	✓	🔒
GET	/users/permissions/{user_id}	Get User	✓	🔒
PUT	/users/permissions/set_avatar/	Set Avatar	✓	🔒
POST	/users/auth/login	Login	✓	
POST	/users/auth/logout	Logout	✓	🔒
POST	/users/auth/register	Register	✓	

#### Задачи

GET	/tasks/statistic/	Get Statistic	✓	🔒
GET	/tasks/task/	Get All	✓	🔒
POST	/tasks/task/	Create	✓	🔒
PUT	/tasks/task/{task_id}	Put	✓	🔒
DELETE	/tasks/task/{task_id}	Delete	✓	🔒

#### Статистика

GET	/tasks/statistic/	Get Statistic	✓	🔒
-----	-------------------	---------------	---	---

#### Факты

GET	/facts	Get All	✓	
POST	/facts	Create One	✓	
DELETE	/facts	Delete All	✓	
GET	/facts/{item_id}	Get One	✓	
PUT	/facts/{item_id}	Update One	✓	
DELETE	/facts/{item_id}	Delete One	✓	
GET	/facts/get_random/	Get Random	✓	

## Факт дня

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** /facts/get\_random/ (with a sub-label "Get Random")
- Parameters:** No parameters
- Execute:** A blue button to execute the request.
- Clear:** A button to clear the request.
- Responses:**
  - Curl:**

```
curl -X 'GET' \
'http://127.0.0.1:8000/facts/get_random/' \
-H 'accept: application/json'
```
  - Request URL:** `http://127.0.0.1:8000/facts/get_random/`
  - Server response:**
    - Code:** 200
    - Response body:**

```
{
  "id": 3,
  "text": "Микроволновая печь была изобретена, когда исследователь проходил мимо лучевой трубки, и плитка шоколада растаяла в его кармане."
}
```
    - Response headers:**

```
content-length: 253
content-type: application/json
date: Mon, 06 Dec 2021 17:31:20 GMT
server: uvicorn
```

## Задачи

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** /tasks/task/ (with a sub-label "Get All")
- Parameters:** No parameters
- Execute:** A blue button to execute the request.
- Clear:** A button to clear the request.
- Responses:**
  - Curl:**

```
curl -X 'GET' \
'http://127.0.0.1:8000/tasks/task/' \
-H 'accept: application/json'
```
  - Request URL:** `http://127.0.0.1:8000/tasks/task/`
  - Server response:**
    - Code:** 200
    - Response body:**

```
[
  {
    "id": 4,
    "text": "Дело 4",
    "finished": false
  },
  {
    "id": 3,
    "text": "Дело 3",
    "finished": false
  },
  {
    "id": 2,
    "text": "Дело 2",
    "finished": true
  }
]
```
    - Response headers:** (Not visible in the screenshot)

**Вывод:** Была изучена структура формата представления данных JSON, типы запросов к API: HEAD, GET, POST, PUT, DELETE, спроектировано и реализовано собственное REST API.