Министерство науки и высшего образования Российской Федерации   
Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Самарский национальный исследовательский университет   
имени академика С.П. Королева»

Институт информатики и кибернетики

Кафедра технической кибернетики

**Отчет по лабораторной работе №2**

**Разработка RESTful API для сервисов**

**для приложения “Планировщик задач”**

Выполнила

Студент группы 6305-010302D

Радаева Евгения Денисовна

**Самара 2025**

Цель работы

Разработать RESTful API для сервисов.

Основные задачи

1. Реализация API:

* ***Разработать CRUD-методы для работы с моделями:***

Реализованы полные CRUD-операции для трех сущностей (таблицы 1-3):

Таблица 1 – CRUD-операции для сущности Пользователь (User)

|  |
| --- |
| **Пользователи (Users)** |
| * GET /api/users/ - получение списка всех пользователей * GET /api/users/{username}/ - получение пользователя по имени * PUT /api/users/{username}/ - обновление данных пользователя * DELETE /api/users/{username}/ - удаление пользователя * GET /api/users/{username}/profile/ - получение профиля пользователя |

Таблица 2 – CRUD-операции для сущности Категория (Category)

|  |
| --- |
| **Категории (Categories)** |
| * GET /api/categories/ - получение всех категорий * POST /api/categories/ - создание новой категории * GET /api/categories/{id}/ - получение категории по ID * PUT /api/categories/{id}/ - обновление категории * DELETE /api/categories/{id}/ - удаление категории * GET /api/categories/{id}/tasks/ - получение задач категории |

Таблица 3 – CRUD-операции для сущности Задача (Task)

|  |
| --- |
| **Задачи (Tasks)** |
| * GET /api/tasks/ - получение всех задач * POST /api/tasks/ - создание новой задачи * GET /api/tasks/{id}/ - получение задачи по ID * PUT /api/tasks/{id}/ - обновление задачи * DELETE /api/tasks/{id}/ - удаление задачи * GET /api/tasks/{id}/categories/ - получение категорий задачи |

Также были опробованы Django сессии для осуществления простой авторизации (таблица 4).

Таблица 4 – Операции регистрации, авторизации и завершения сессии

|  |
| --- |
| **Аутентификация (Auth)** |
| * POST /api/auth/register/ - регистрация нового пользователя * POST /api/auth/login/ - аутентификация пользователя * POST /api/auth/logout/ - завершение сессии |

* ***Настроить маршруты и обработку запросов.***

Для настройки маршрутов в папке приложения был создан файл urls.py, для регистрации endpoints был использован DefaultRouter().

Помимо этого были созданы файлы представлений views.py и сериализации serializers.py, которые задействуются в процессе обработки запросов.

1. Тестирование API:

* ***Использовать Postman/Curl для проверки запросов;***
* ***Описать примеры запросов и ответов в документации.***

Для проверки запросов была использована программа Postman. Приведем примеры тестовых запросов.

*Запросы для сущности* ***Пользователи (Users).***

Выведем для начала список всех существующих пользователей (рисунок 1а). Затем попробуем вывести только пользователя user2 (рисунок 1б). Теперь изменим почту и временной пояс (рисунок 1в).

|  |  |  |
| --- | --- | --- |
|  |  |  |
| а) | б) | в) |

Рисунок 1 – а) список всех пользователей; б) пользователь user2; в) изменение пользователя user2

Опять выведем список всех пользователей, видим, что изменения прошли успешно (рисунок 2а). Попробуем теперь удалить пользователя user2 (рисунок 2б). Опять выведем список пользователей, видим, что удаленного пользователя больше нет (рисунок 2в).

|  |  |  |
| --- | --- | --- |
|  |  |  |
| а) | б) | в) |

Рисунок 2 – а) после изменения user2; б) удаление user2; в) после удаления

Также есть запрос, выдающий профиль пользователя.

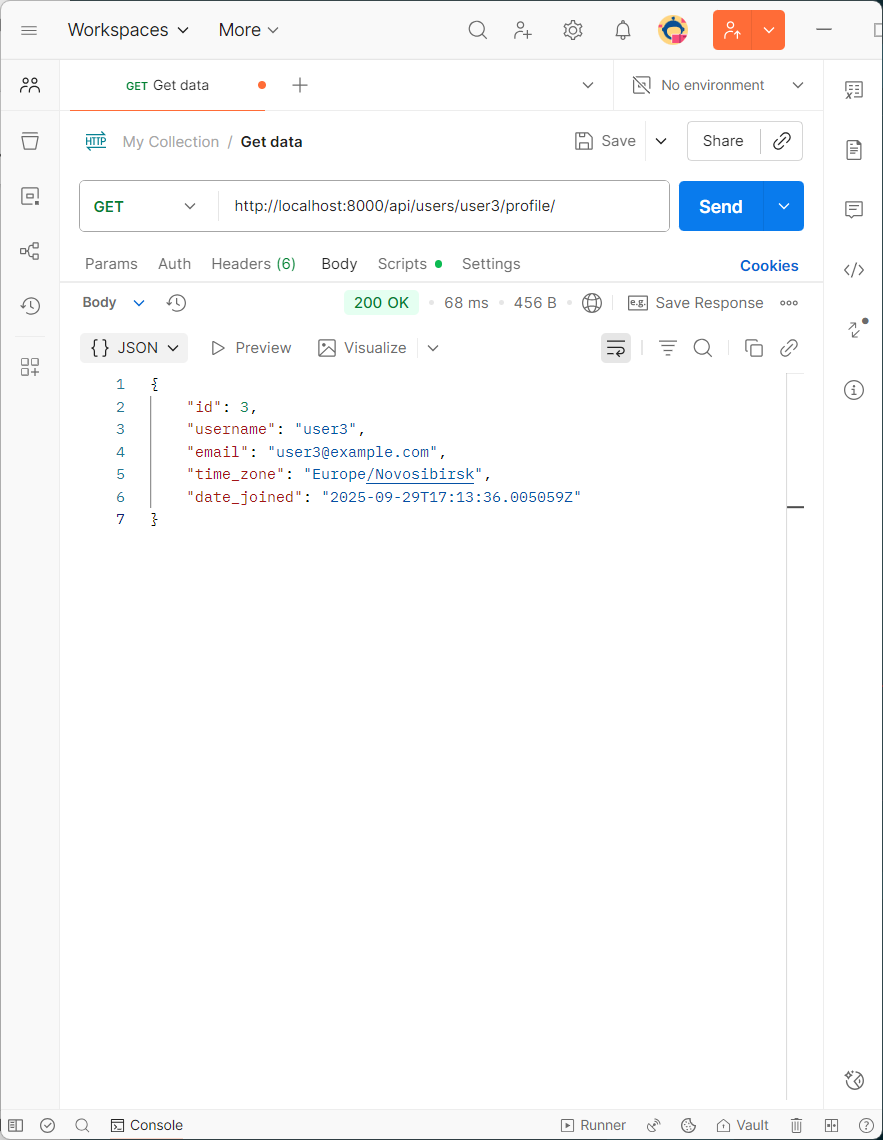


Рисунок 3 – Профиль пользователя

*Запросы для сущности* ***Категории (Categories).***

Вначале выведем список всех категорий (рисунок 4а). Теперь попробуем сделать новую категорию “Семья” (рисунок 4б), а после получим ее по id (рисунок 4в).

|  |  |  |
| --- | --- | --- |
|  |  |  |
| а) | б) | в) |

Рисунок 4 – а) список всех категорий; б) создание новой категории “Семья”; в) получение категории “Семья” по id

Выведем список всех категорий после создания новой категории (рисунок 5а). Попробуем изменить ее название с “Семья” на ”Family”, а также поменяем цвет (рисунок 5б). Теперь удалим эту категорию (рисунок 5в).

|  |  |  |
| --- | --- | --- |
|  |  |  |
| а) | б) | в) |

Рисунок 5 – а) после добавления; б) изменение категории “Семья”; в) удаление категории “Семья”

Можем также получить все задачи, принадлежащие одной категории, например, выведем список задач из категории “Health” (рисунок 6):

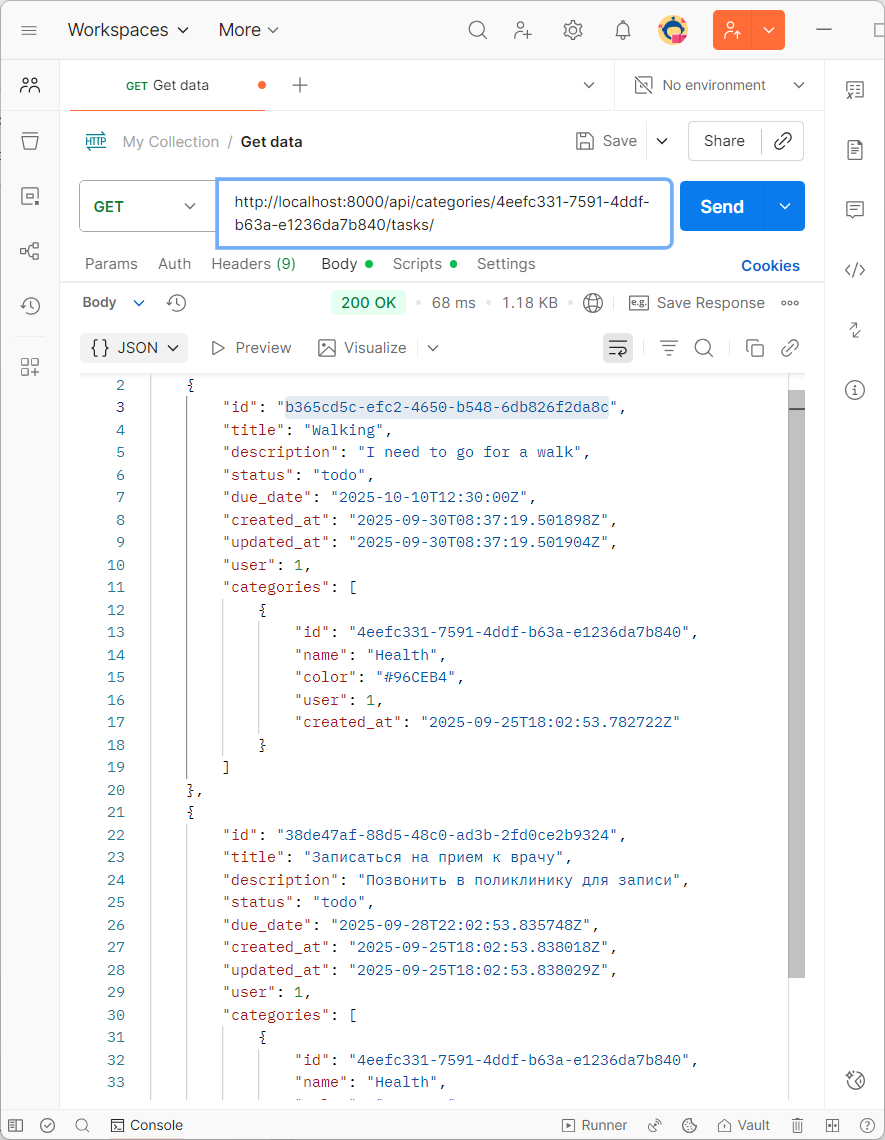


Рисунок 6 – Список всех задач из категории “Health”

*Запросы для сущности* ***Задачи (Tasks).***

Выведем список всех задач (рисунок 7а). Теперь создадим новую задачу “Plants” (рисунок 7б), а после получим ее по id (рисунок 7в).

|  |  |  |
| --- | --- | --- |
|  |  |  |
| а) | б) | в) |

Рисунок 7 – а) список всех задач; б) создание новой задачи “Plants”; в) получение задачи “Plants” по id

Выведем список категорий, которым принадлежит задача “Plants” (рисунок 8а). Теперь изменим описание и статус этой задачи (рисунок 7б), а после удалим ее (рисунок 8в).

|  |  |  |
| --- | --- | --- |
|  |  |  |
| а) | б) | в) |

Рисунок 8 – а) список категорий задачи “Plants”; б) изменение задачи; в) удаление задачи

*Запросы для* ***регистрации, аутентификации и завершения сессии***

На простом уровне в тестовом режиме с помощью Django-сессий были сделаны запросы для регистрации (рисунок 9-10), аутентификации (рисунок 11-12) и завершения сессии (рисунок 13-14).

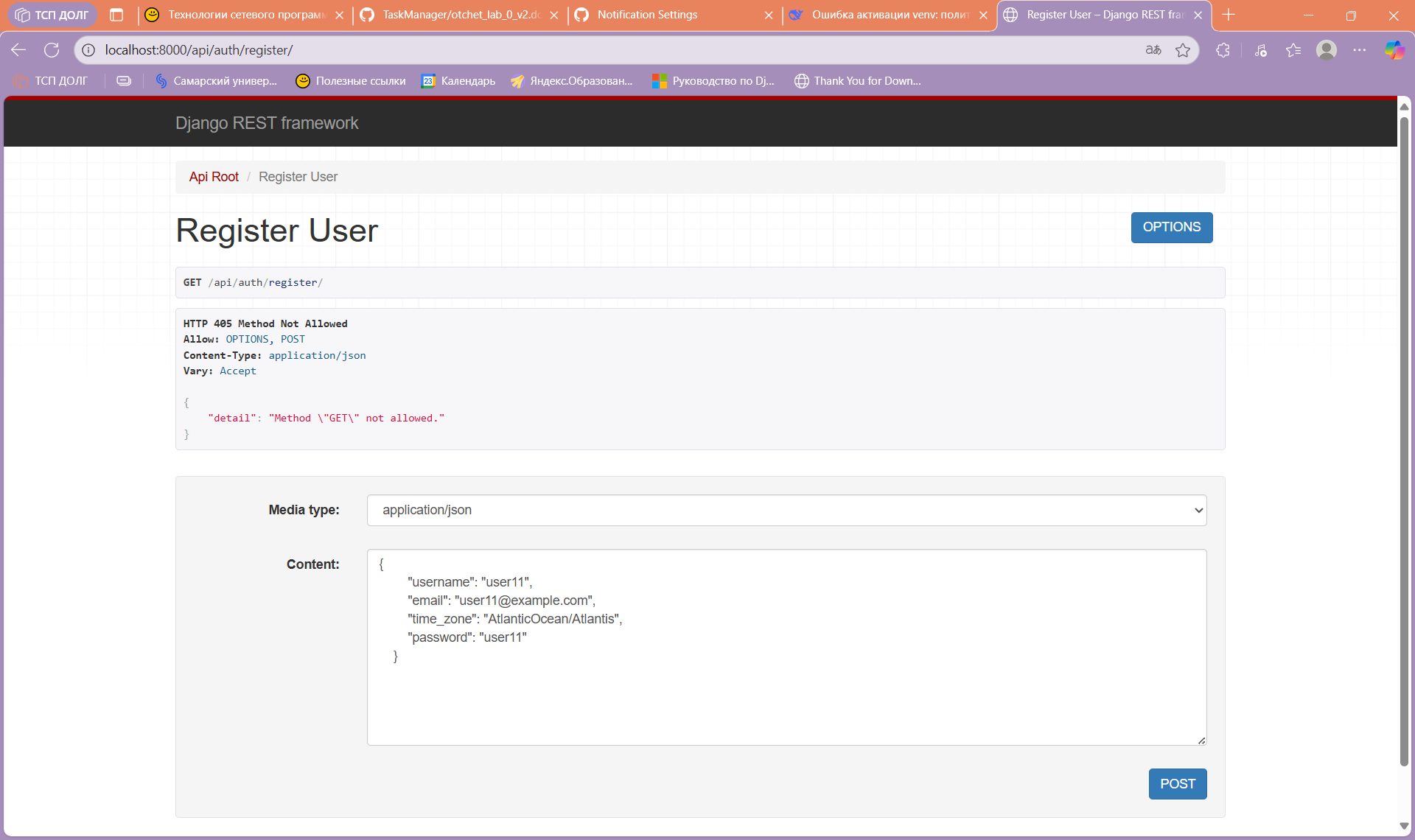


Рисунок 9 – Запрос на регистрацию пользователя user11

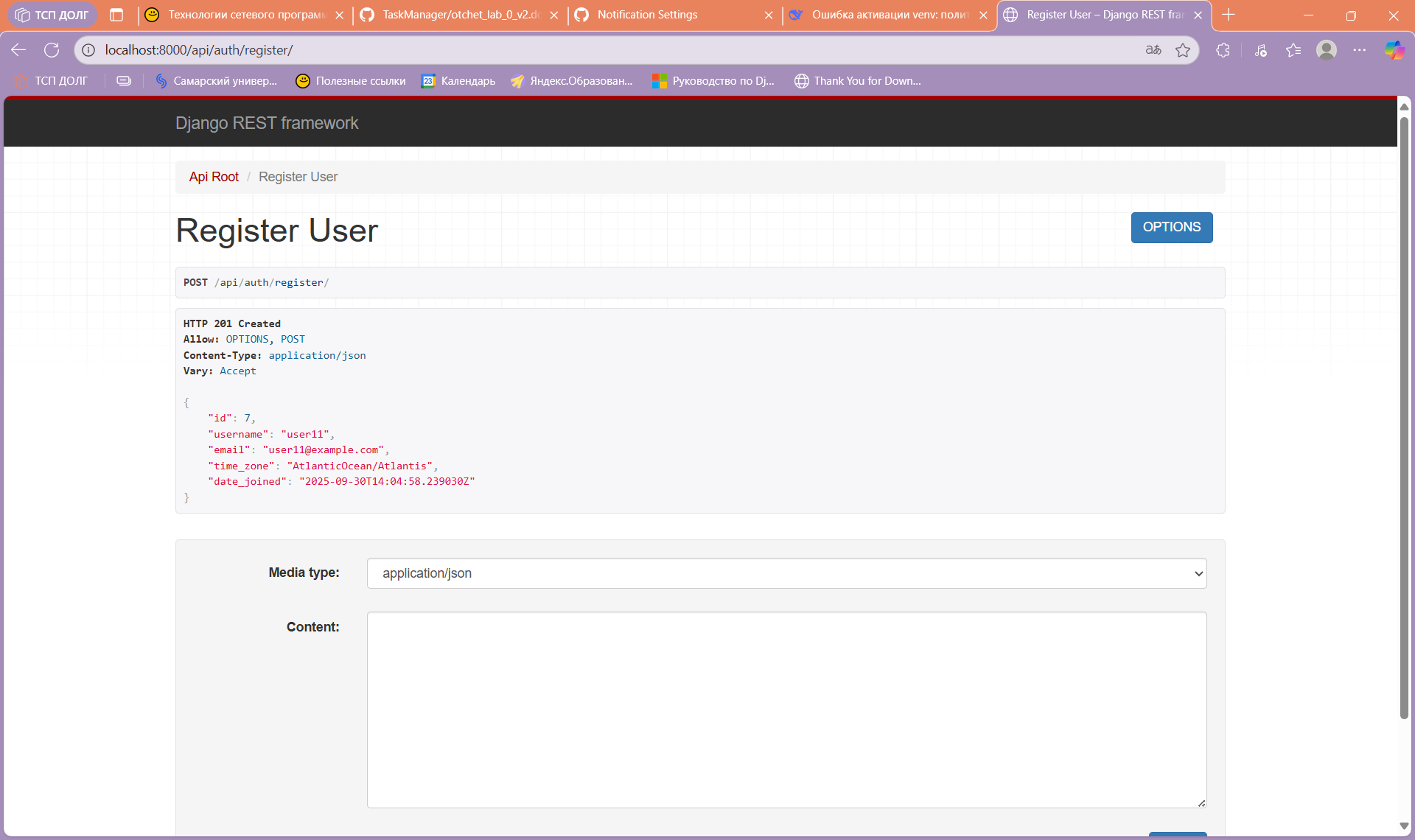


Рисунок 10 – Ответ об успешном создании пользователя

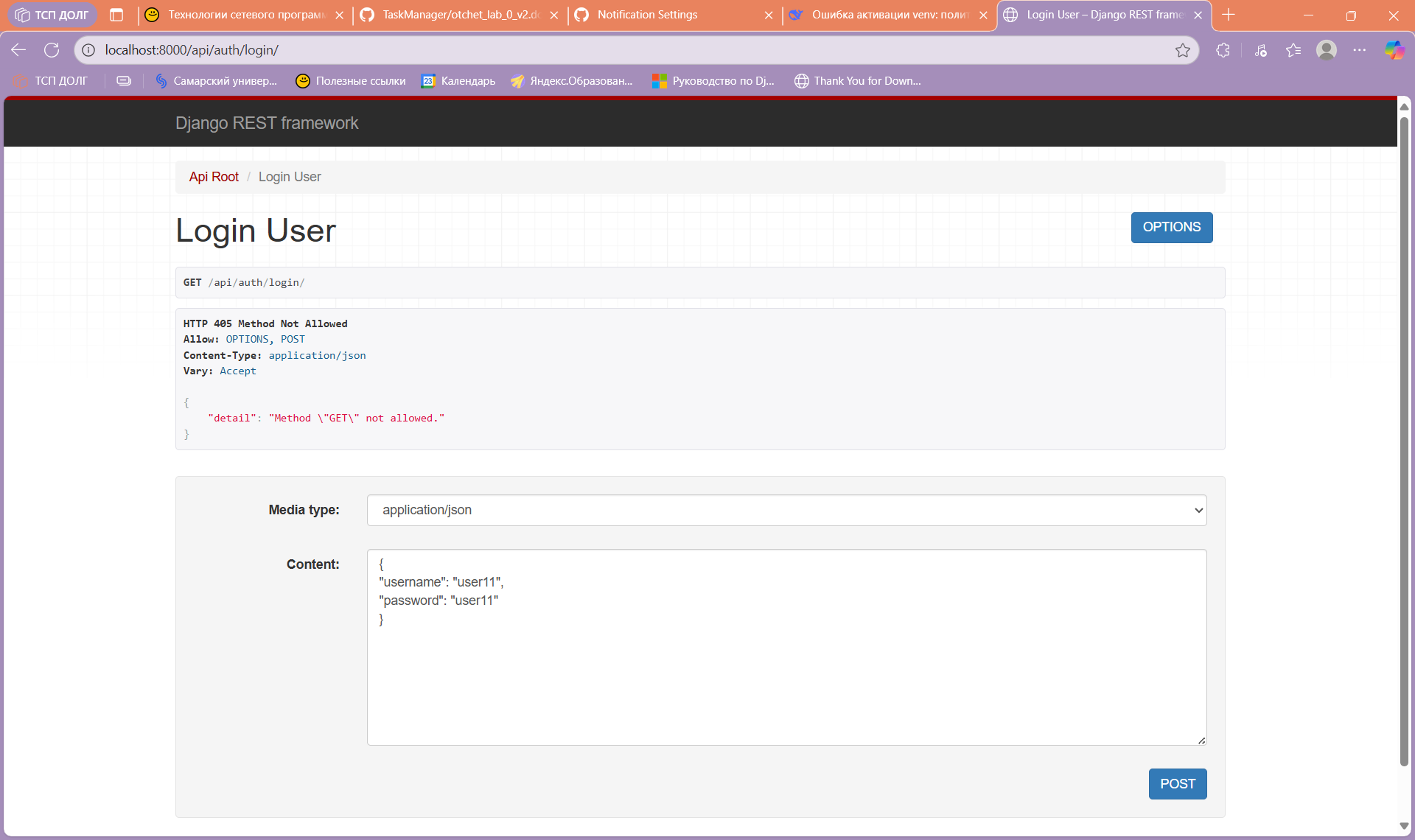


Рисунок 11 – Запрос на аутентификацию пользователя user11

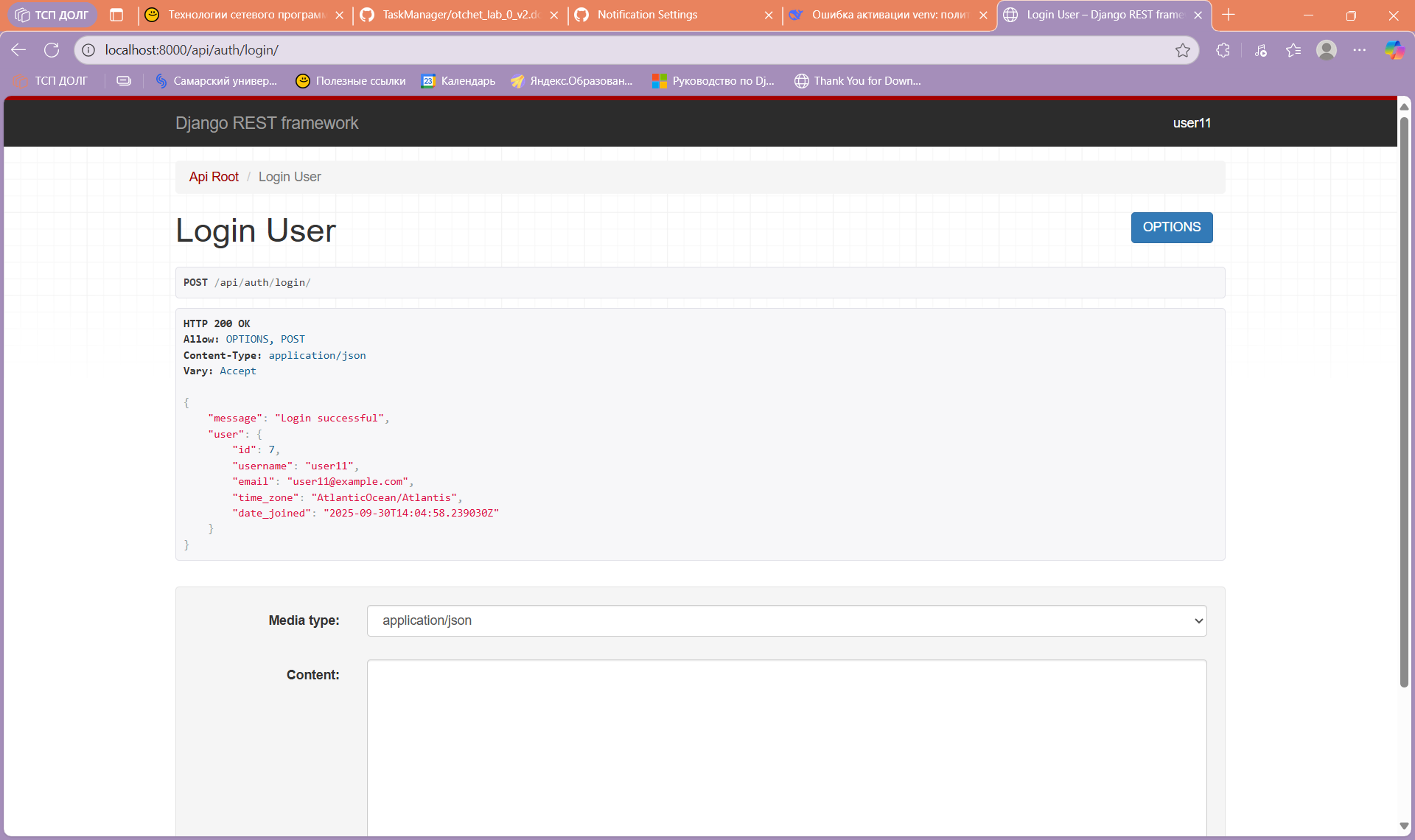


Рисунок 12 – Ответ об успешной аутентификации пользователя user11

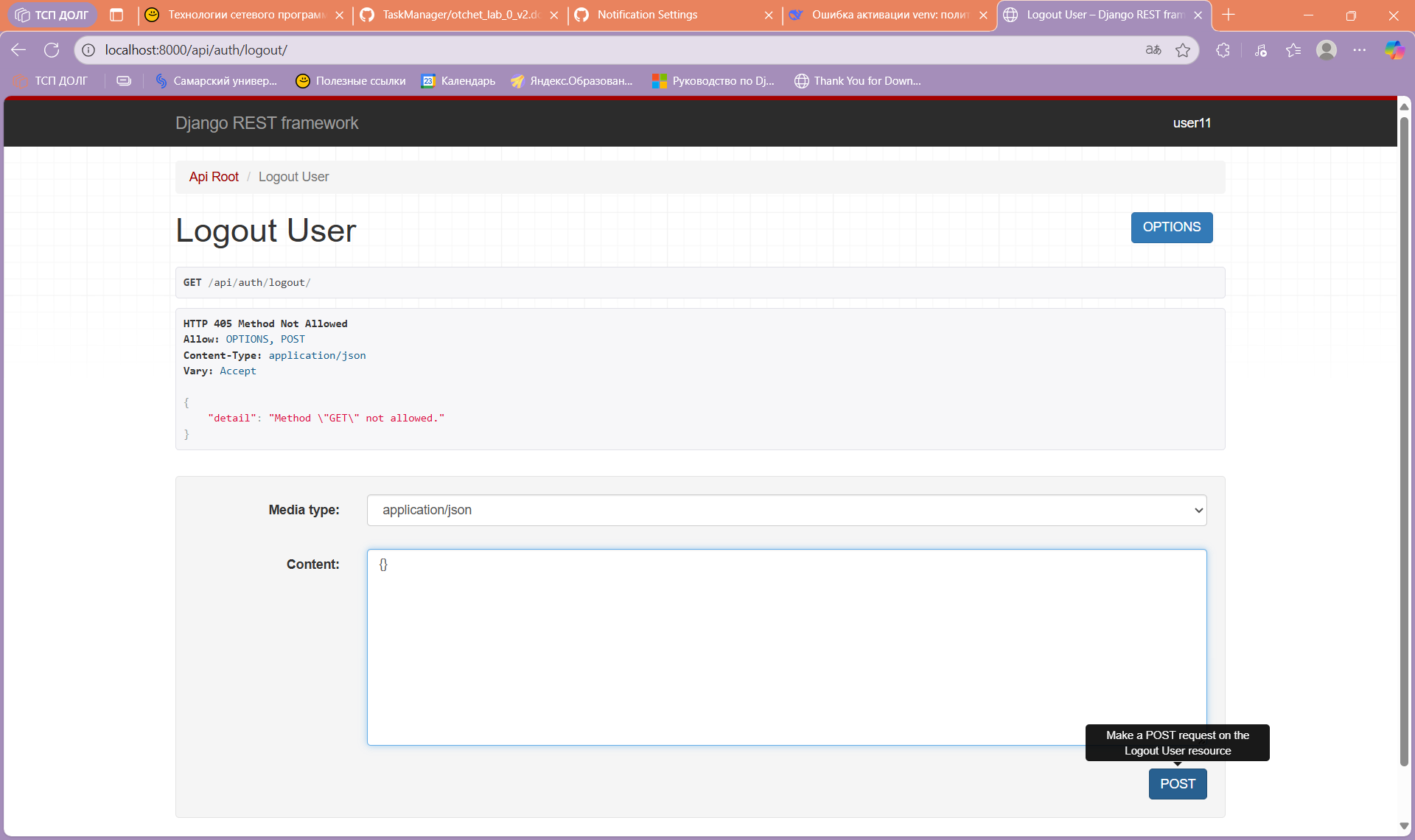


Рисунок 13 – Запрос на завершение сессии пользователя user11

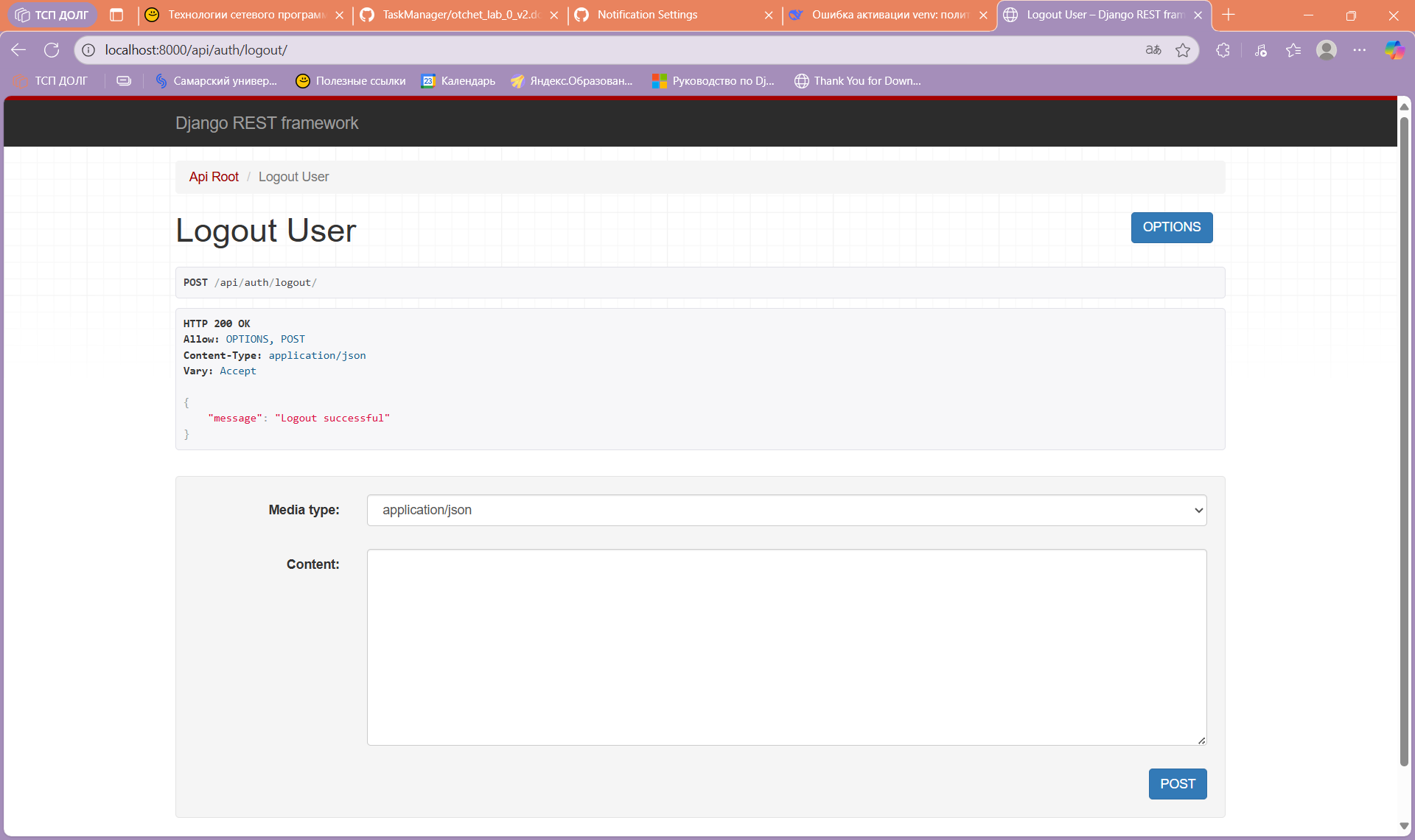


Рисунок 14 – Ответ об успешном завершении сессии пользователя user11