НИУ ИТМО

Факультет программной инженерии и компьютерных технологий

Отчет по лабораторной работе №2

по дисциплине Бизнес-логика программных систем

|  |  |
| --- | --- |
| Студент группы № P33151 | Шипулин Павел Андреевич |
| Преподаватель | Кривоносов Егор Дмитриевич |

Санкт-Петербург

2024

# Задание (вариант 1111)

Доработать приложение из лабораторной работы #1, реализовав в нём управление транзакциями и разграничение доступа к операциям бизнес-логики в соответствии с заданной политикой доступа.

Управление транзакциями необходимо реализовать следующим образом:

1. Переработать согласованные с преподавателем прецеденты (или по согласованию с ним разработать новые), объединив взаимозависимые операции в рамках транзакций.
2. Управление транзакциями необходимо реализовать с помощью Spring JTA.
3. В реализованных (или модифицированных) прецедентах необходимо использовать декларативное управление транзакциями.
4. В качестве менеджера транзакций необходимо использовать Java EE JTA, предварительно преобразовав приложение в war, развёртываемый на сервере приложений WildFly.

Разграничение доступа к операциям необходимо реализовать следующим образом:

1. Разработать, специфицировать и согласовать с преподавателем набор привилегий, в соответствии с которыми будет разграничиваться доступ к операциям.
2. Специфицировать и согласовать с преподавателем набор ролей, осуществляющих доступ к операциям бизнес-логики приложения.
3. Реализовать разработанную модель разграничений доступа к операциям бизнес-логики на базе Spring Security + JAAS. Информацию об учётных записах пользователей необходимо сохранять в файле XML, для аутентификации использовать JWT.

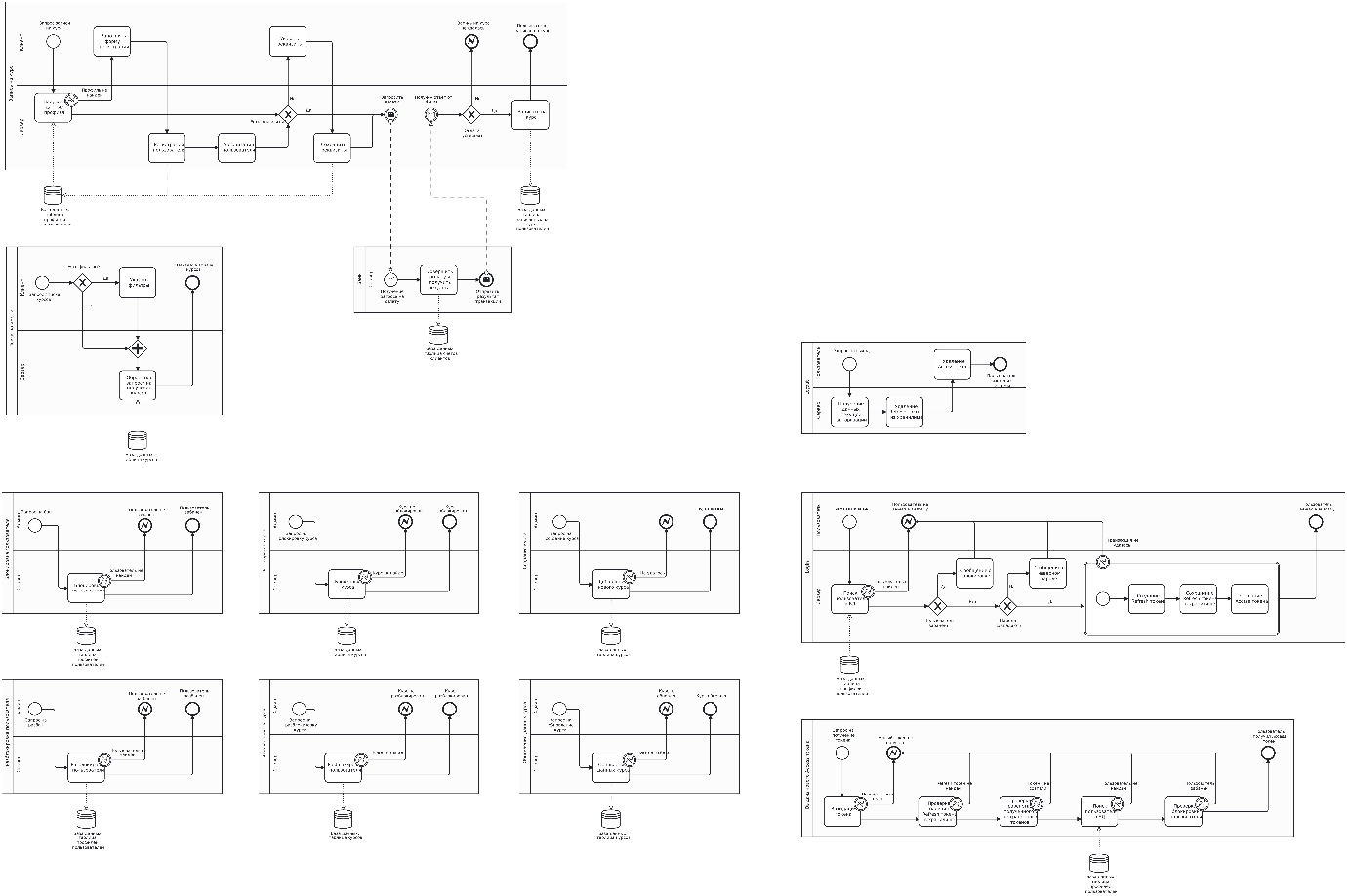
Правила выполнения работы:

1. Все изменения, внесённые в реализуемый бизнес-процесс, должны быть учтены в описывающей его модели, REST API и наборе скриптов для тестирования публичных интерфейсов модуля.
2. Доработанное приложение необходимо развернуть на сервере helios.

Содержание отчёта:

1. Текст задания.
2. Модель потока управления для автоматизируемого бизнес-процесса.
3. Спецификация пользовательских привилегий и ролей, реализованных в приложении.
4. UML-диаграммы классов и пакетов разработанного приложения.
5. Спецификация REST API для всех публичных интерфейсов разработанного приложения.
6. Исходный код системы или ссылка на репозиторий с исходным кодом.
7. Выводы по работе.

# Модель потока управления



<https://github.com/PashcalE2/BLPS/blob/main/BLPS_lab2/report/blps_lab2.svg>

# Роли

## Клиент

1. Выйти из аккаунта.
2. Может записываться на курсы.
3. Может привязать банковскую карту (так как это часть “условности”, эта карта обязана быть в “базе данных банка”).

## Администратор

1. Все привилегии клиента.
2. Может забанить / разбанить пользователя.
3. Может создавать / обновлять курсы.
4. (Искусственно) Может подгрузить пользователей из базы данных в .xml файл (требуется, так как многие сущности в БД связаны с сущностью User).

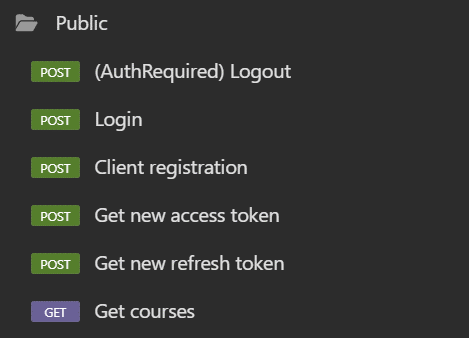
# UML-диаграммы классов и пакетов

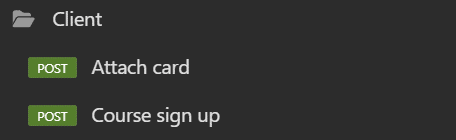
Изображение выглядит как снимок экрана, пространство, линия

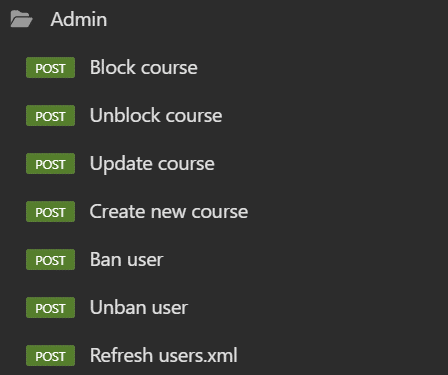
Автоматически созданное описание

<https://github.com/PashcalE2/BLPS/blob/main/BLPS_lab2/report/blps_lab2.png>

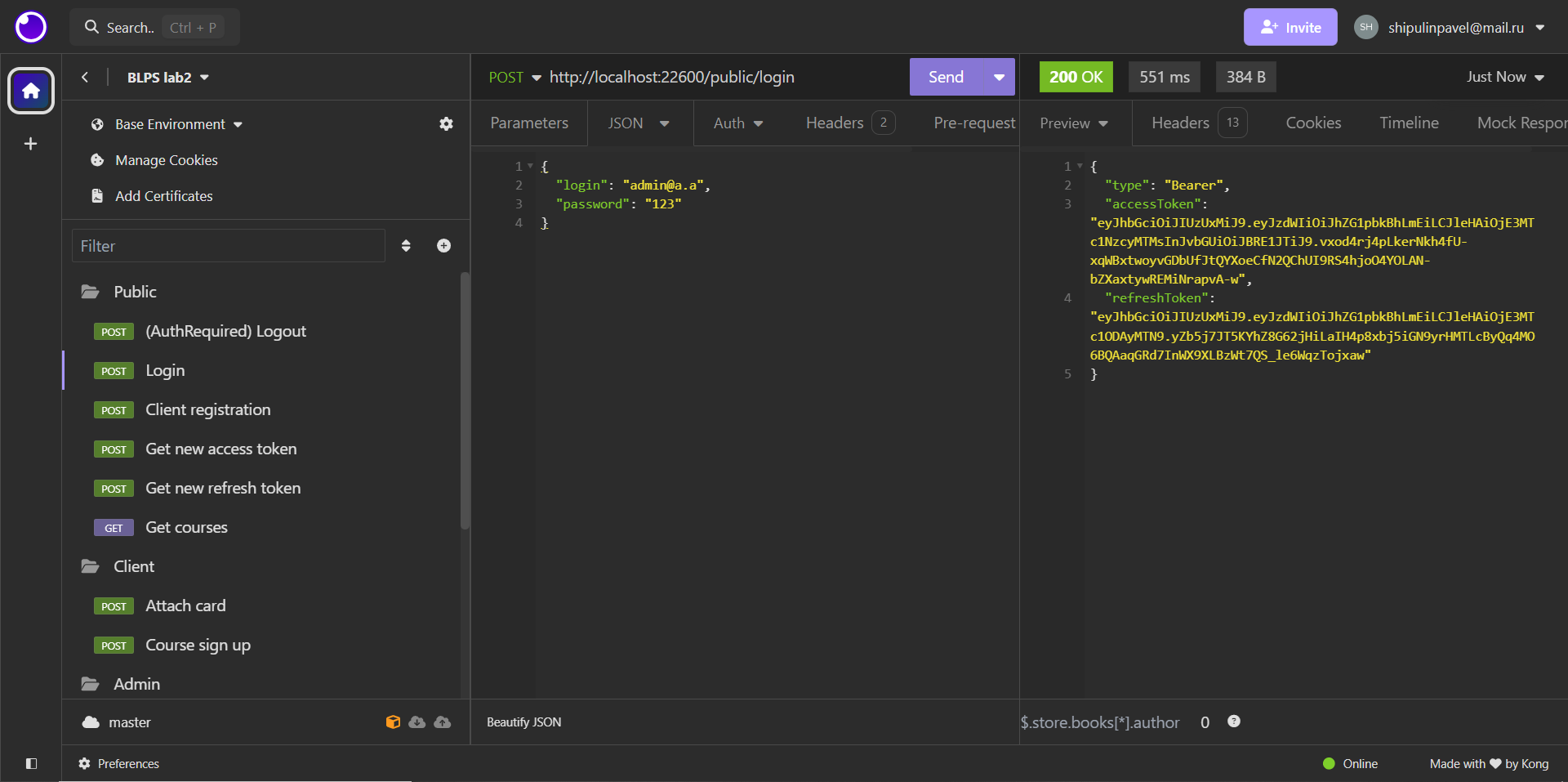
# Спецификация REST API

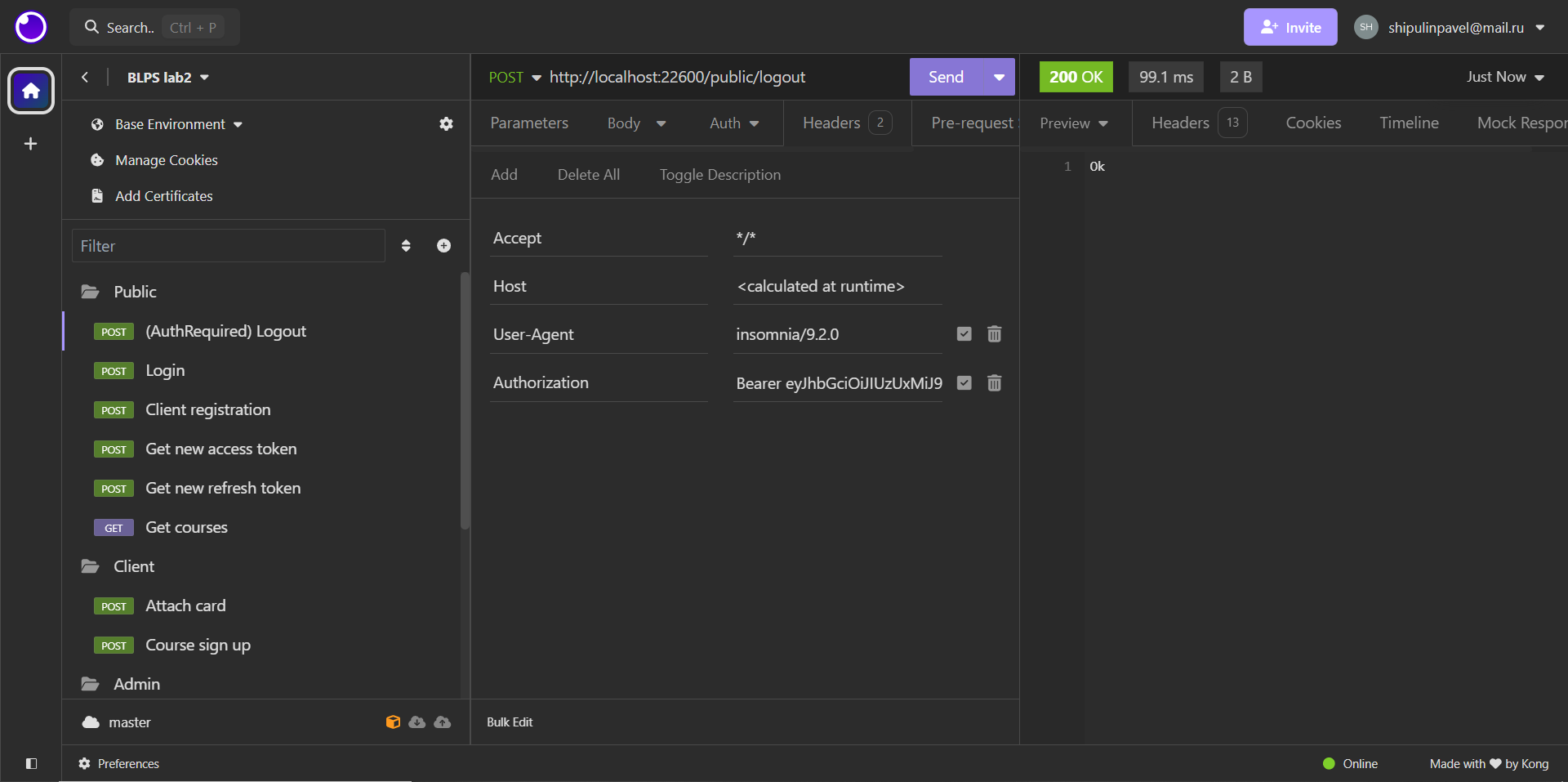


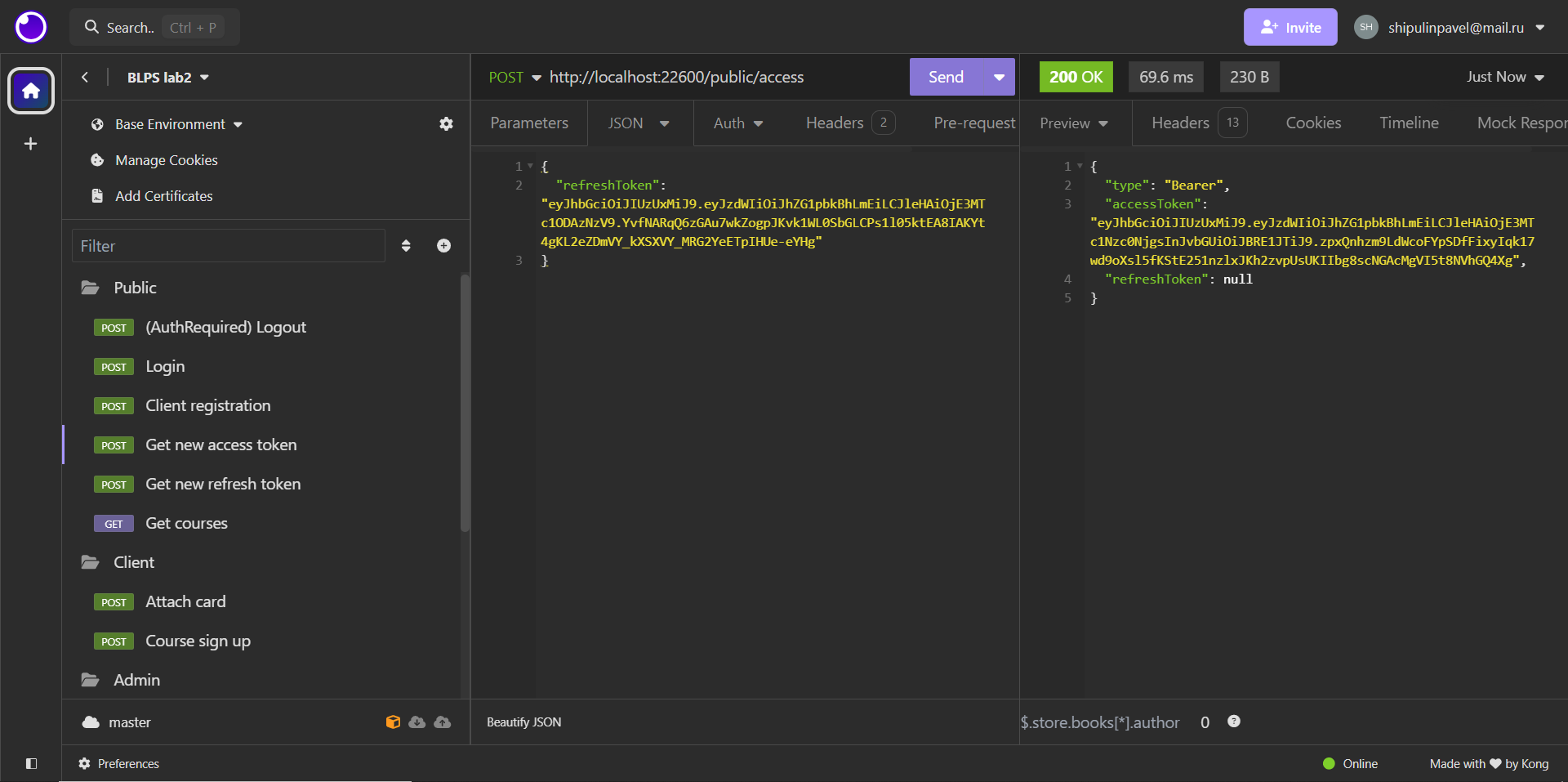


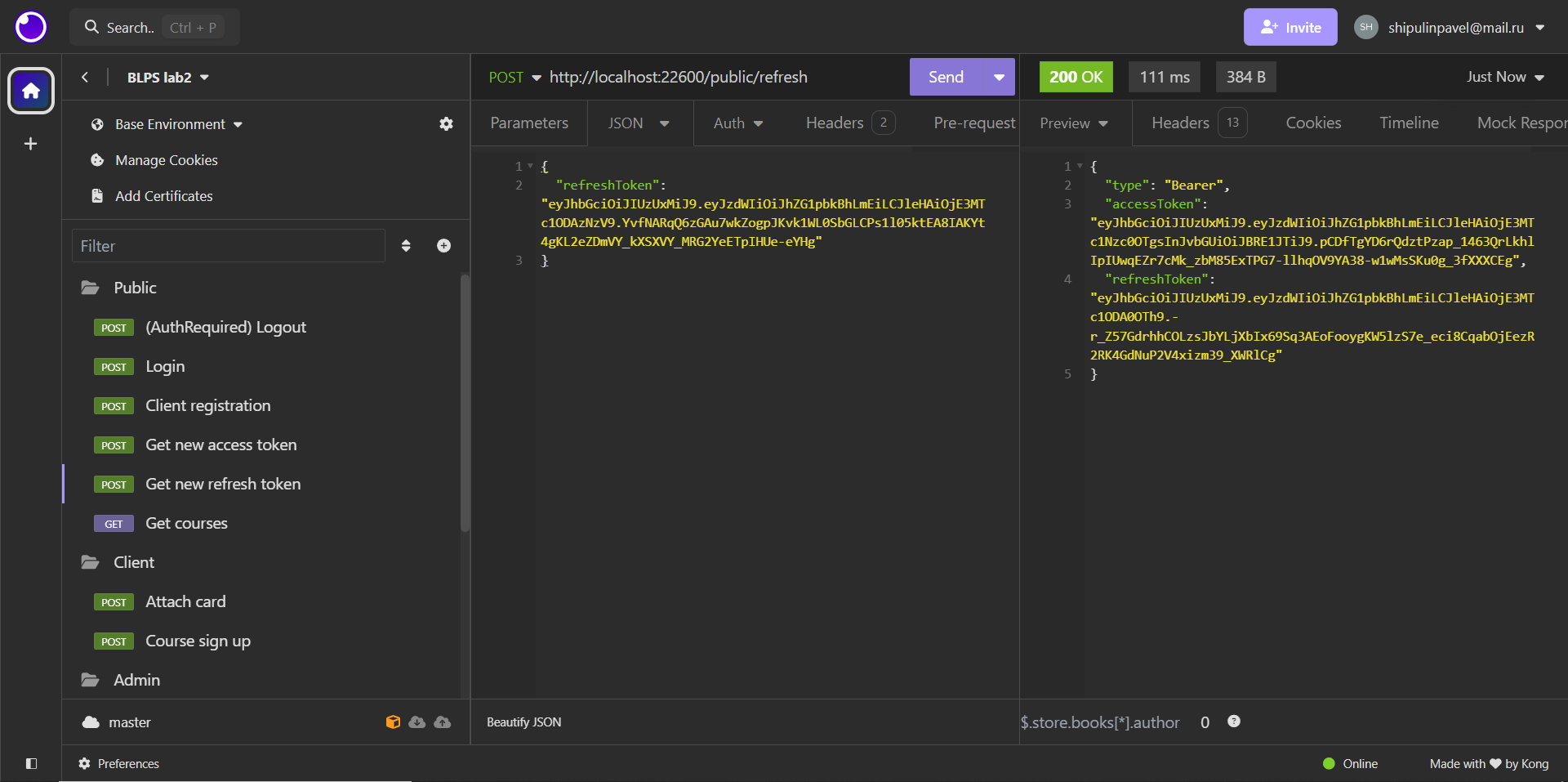


## Публичные интерфейсы





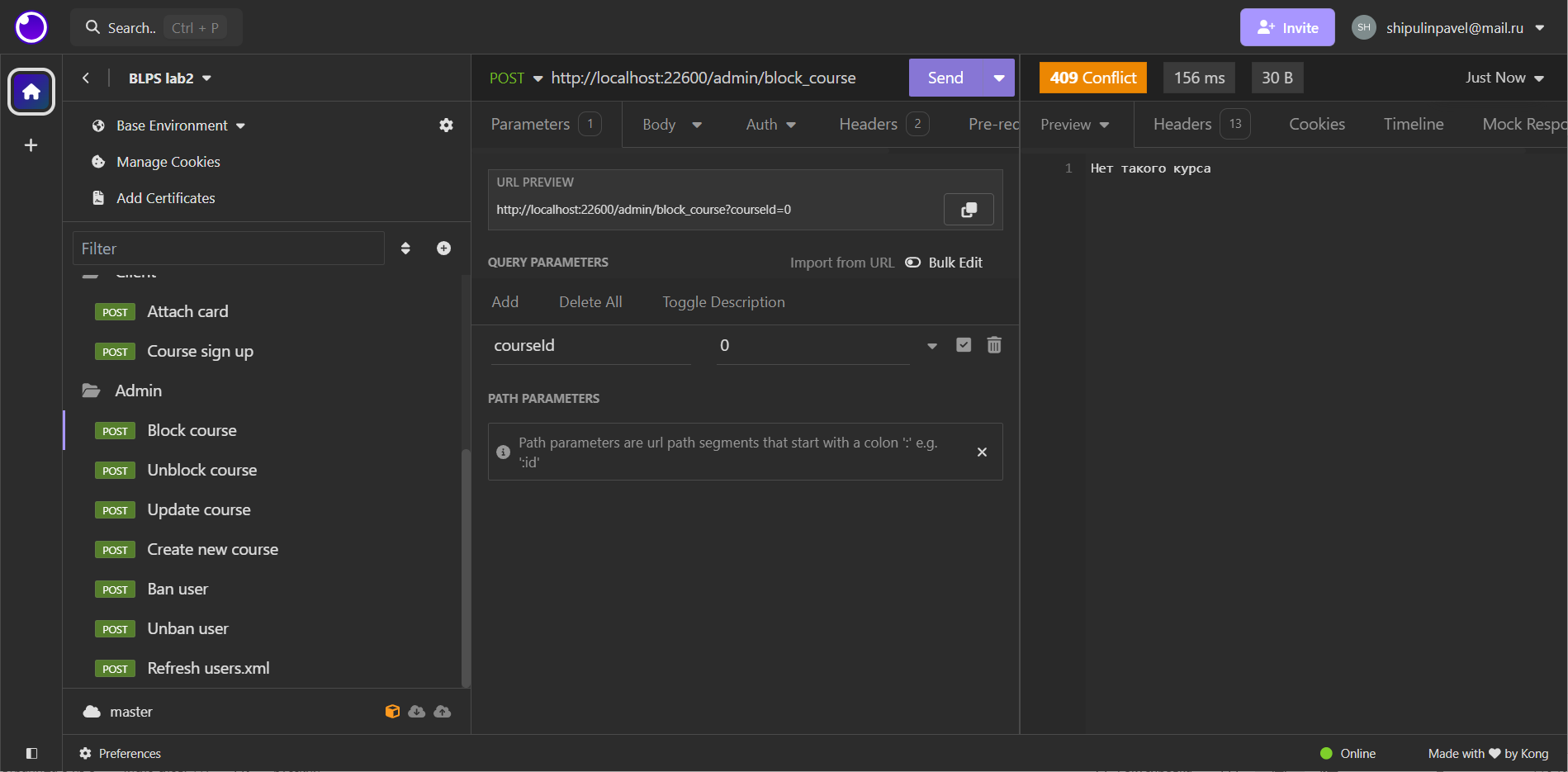


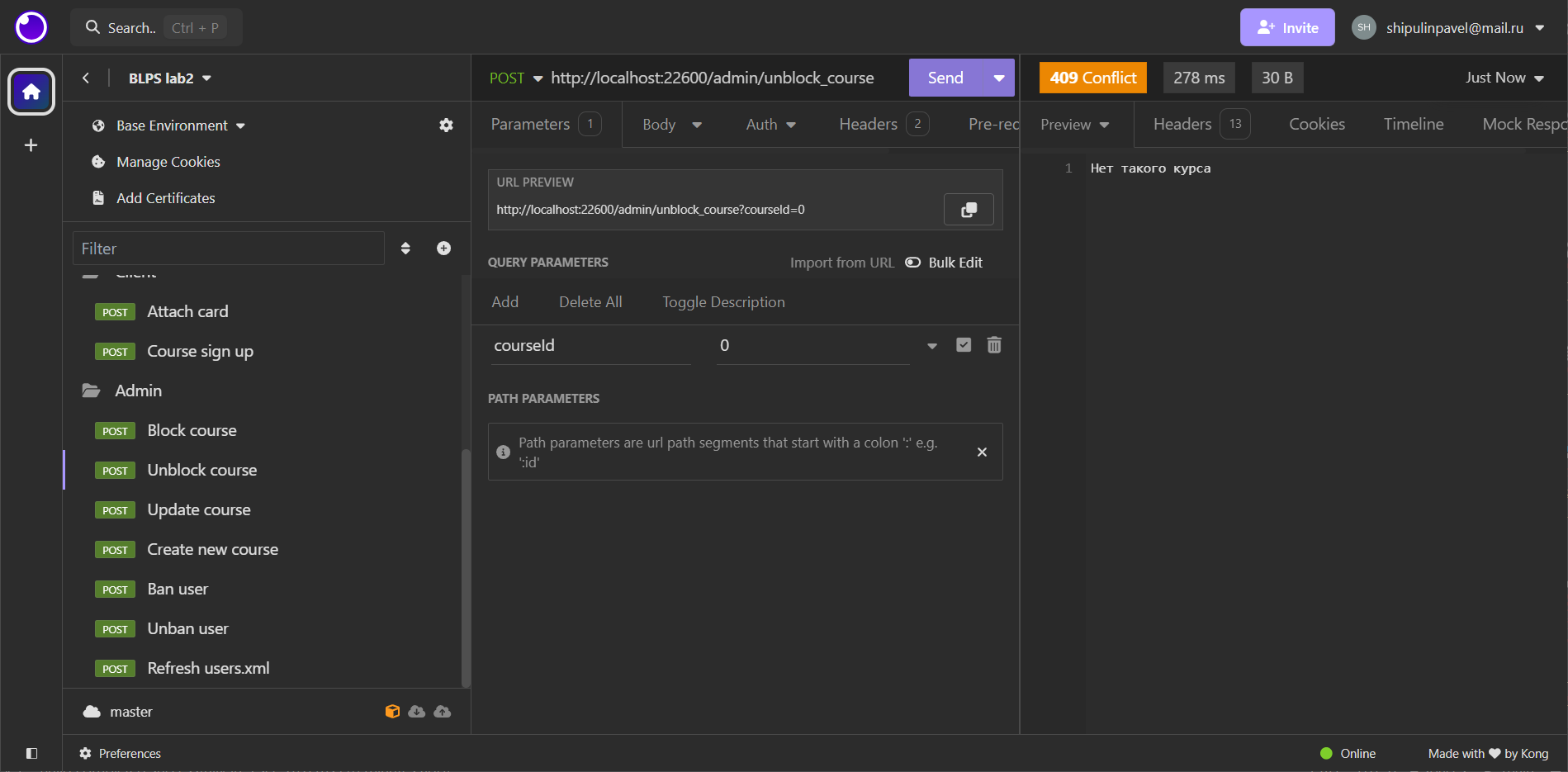


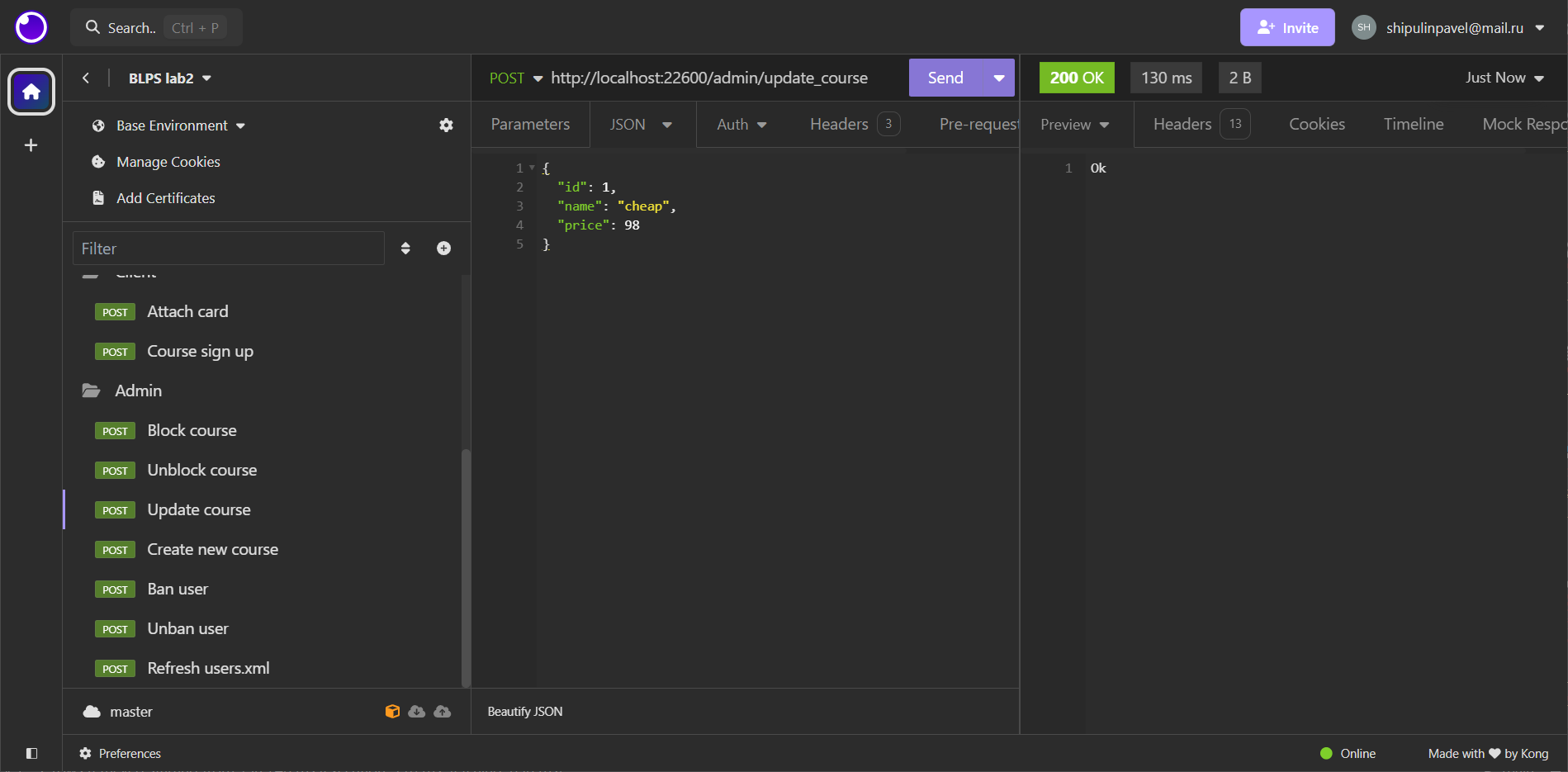
## Интерфейсы для клиентов

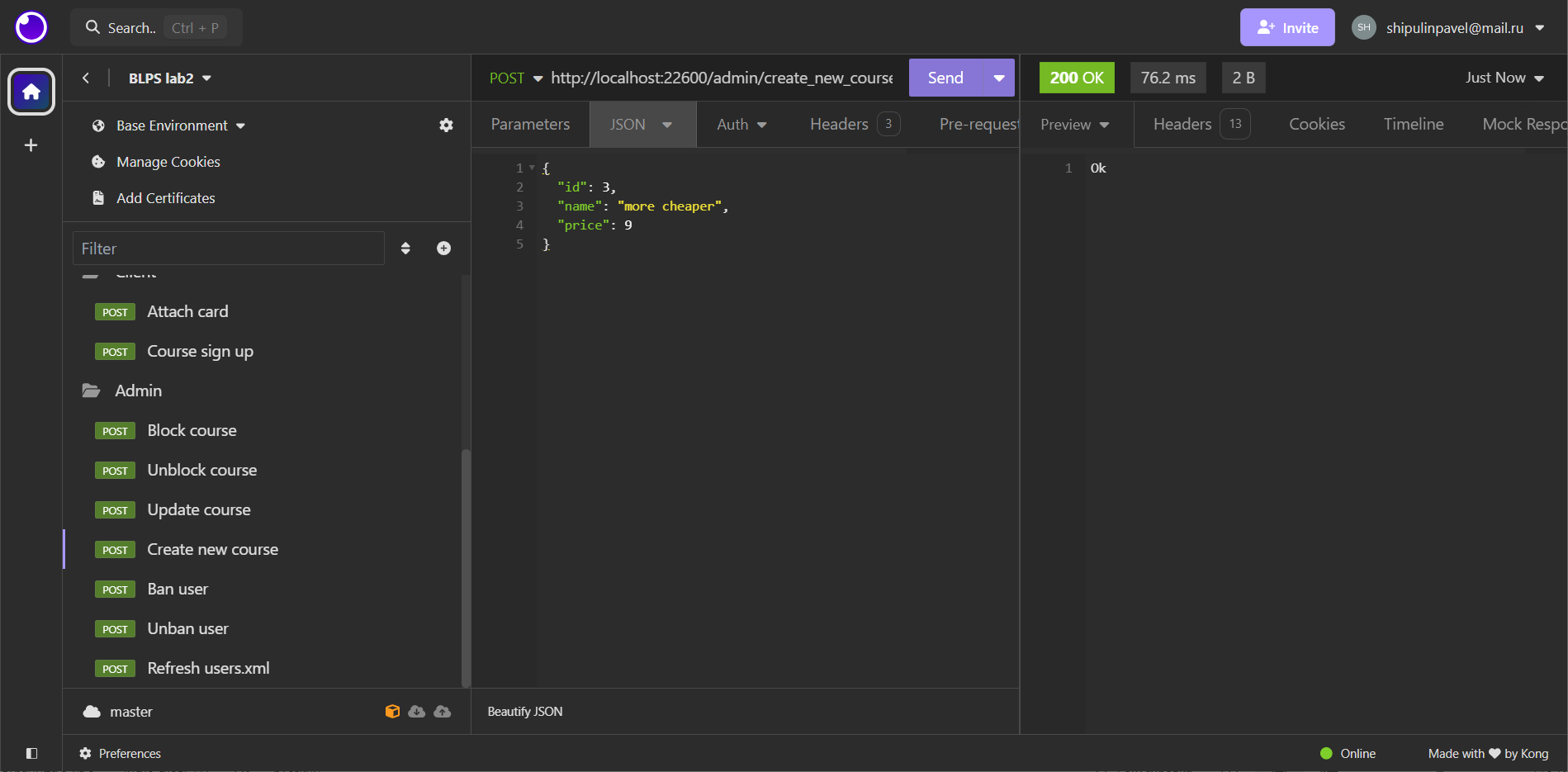
(Новых не добавилось, см. Отчет по работе №1)

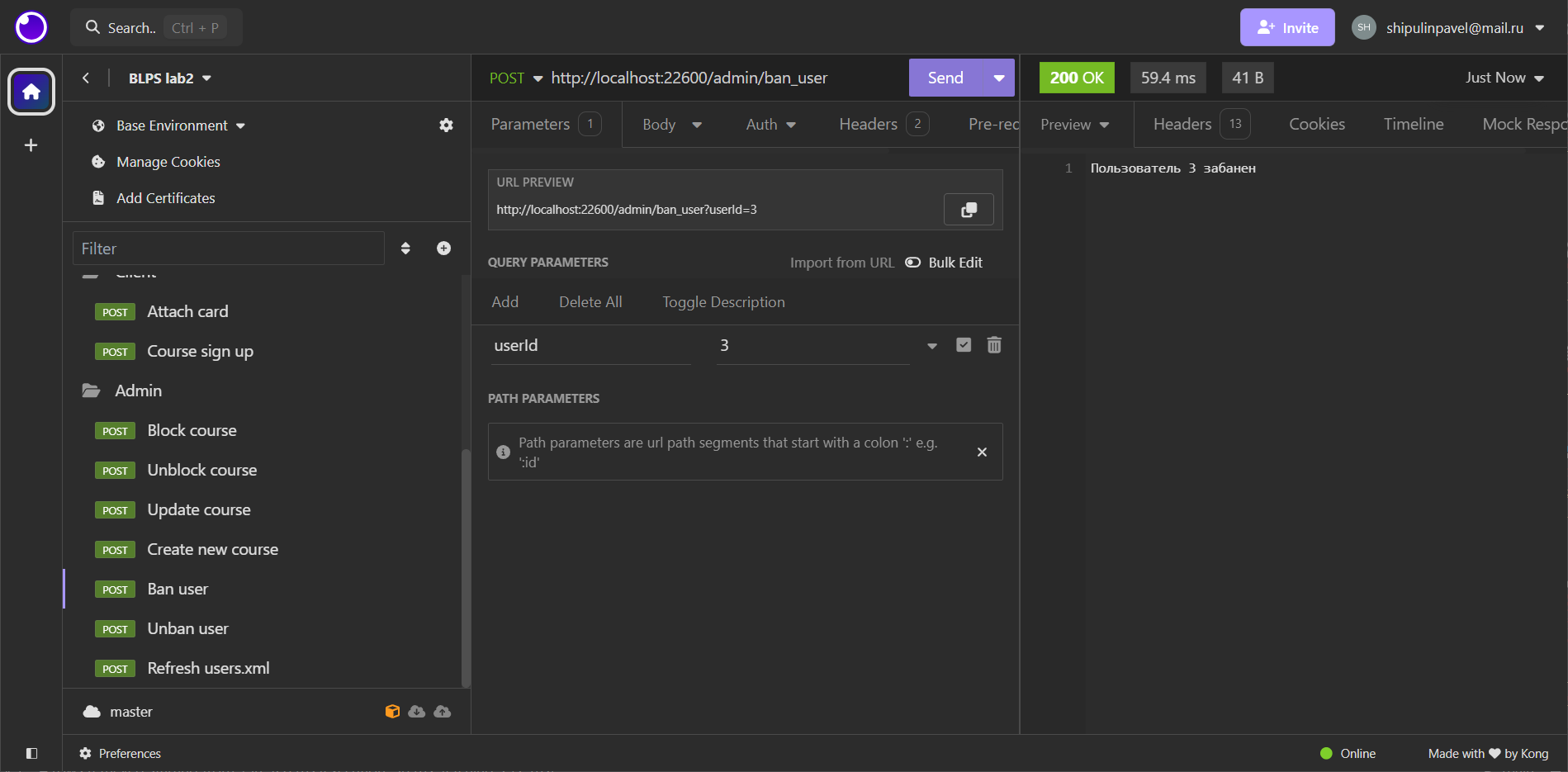
## Интерфейсы для админов

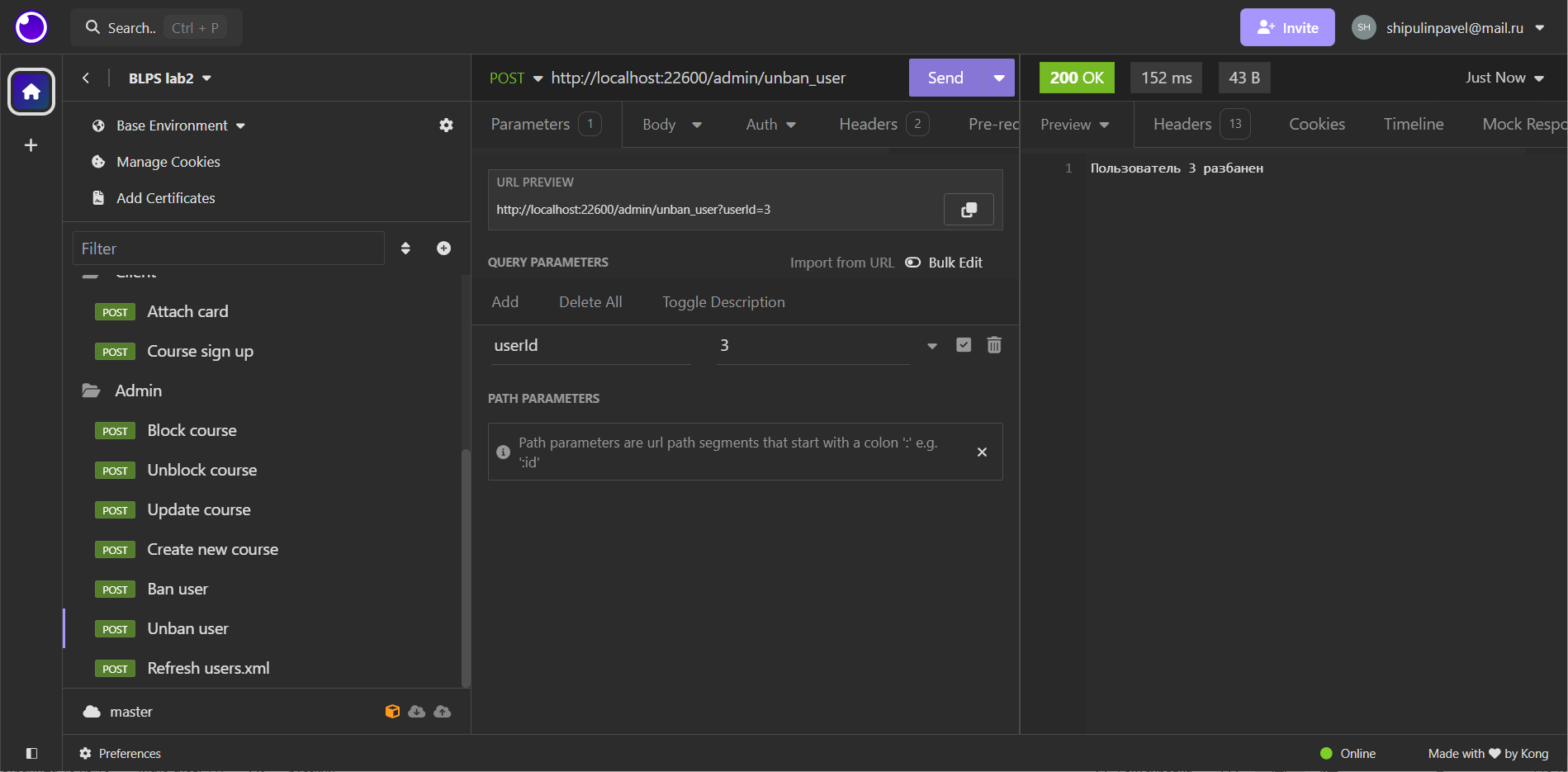


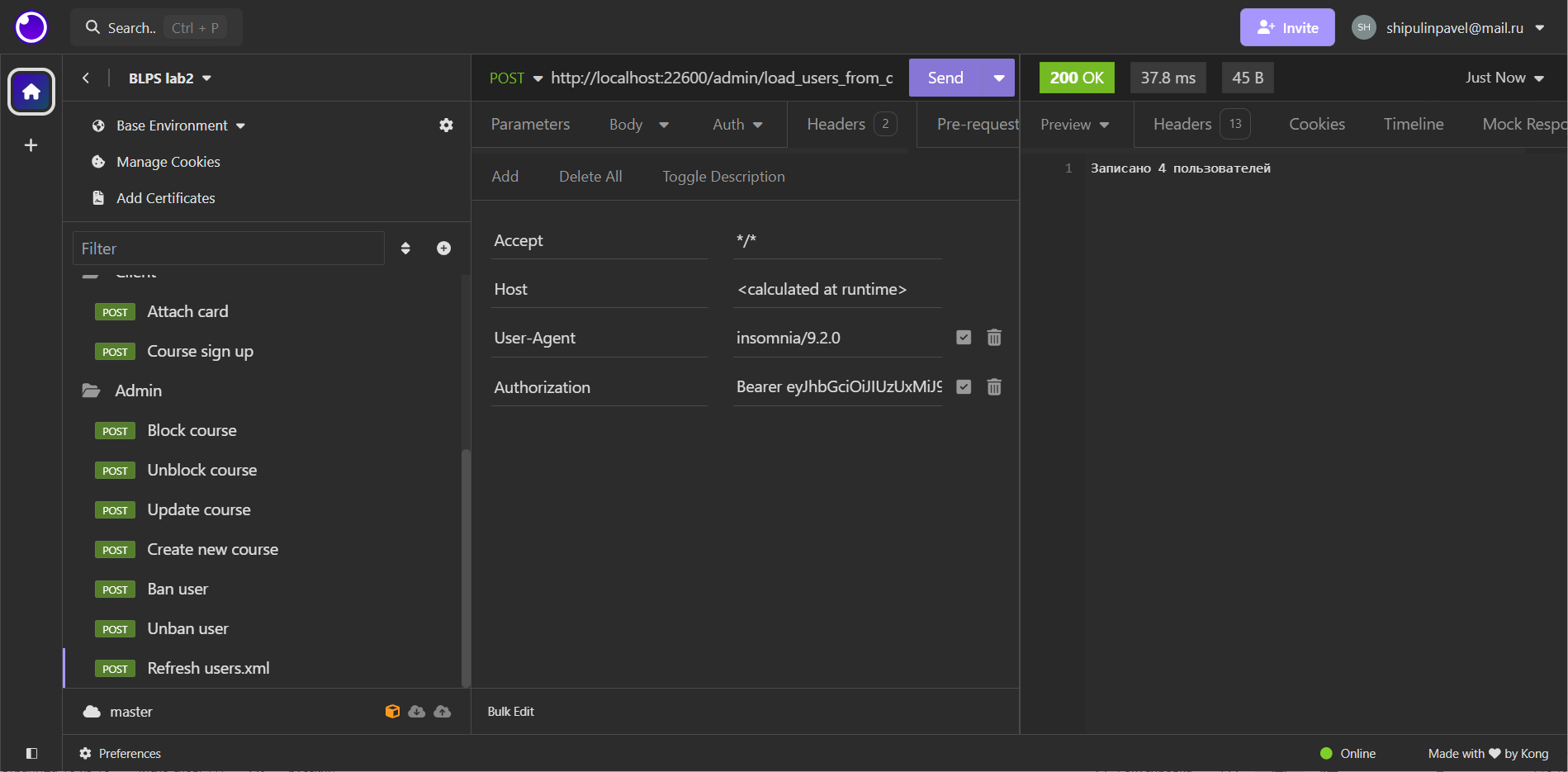












# Исходный код

<https://github.com/PashcalE2/BLPS/tree/main/BLPS_lab2>

# Выводы

Использовал декларативное управление транзакциями в требующих этого прецендентах.

Научился организовывать модели разграничения доступа на базе Spring Security. (Не) Интегрировал JAAS по официальной документации Spring. Реализовал JWT токены (refresh и access) для аутентификации пользователей после первичной проверки логина и пароля.