Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
Факультет инфокоммуникационных технологий

ОТЧЕТ

по Лабораторной работе № 2

Специальность:
09.03.03 Мобильные и сетевые технологии

Проверил:
Добряков Д. И. _____
Дата: «__» _____ 202__г.
Оценка _____

Выполнил:
студенты группы K33401
Ковалев В. М.

Санкт-Петербург
2023

ЦЕЛЬ РАБОТЫ

Реализовать свое RESTful API.

ВЫПОЛНЕНИЕ

Мое приложение представляет собой базу данных игр Steam (~8 тысяч записей), по которой можно делать поиск по категориям, жанрам, разработчикам и сортировкой по цене. Сервис доступен только зарегистрированным пользователям. Игры можно добавлять в избранное. Приложение разделено на микросервисы.

Схема базы данных auth-сервиса

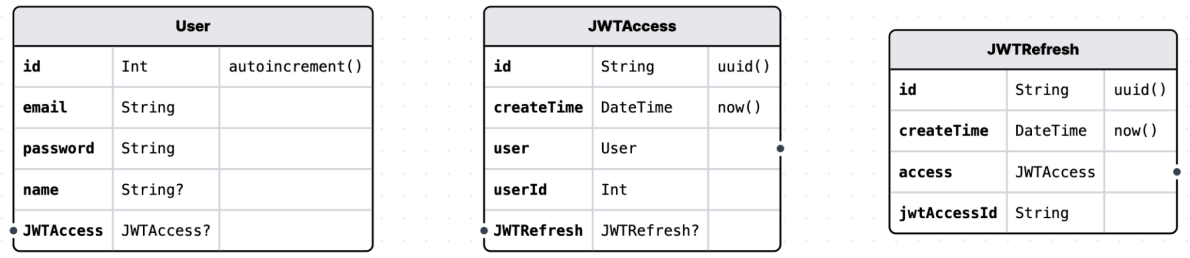
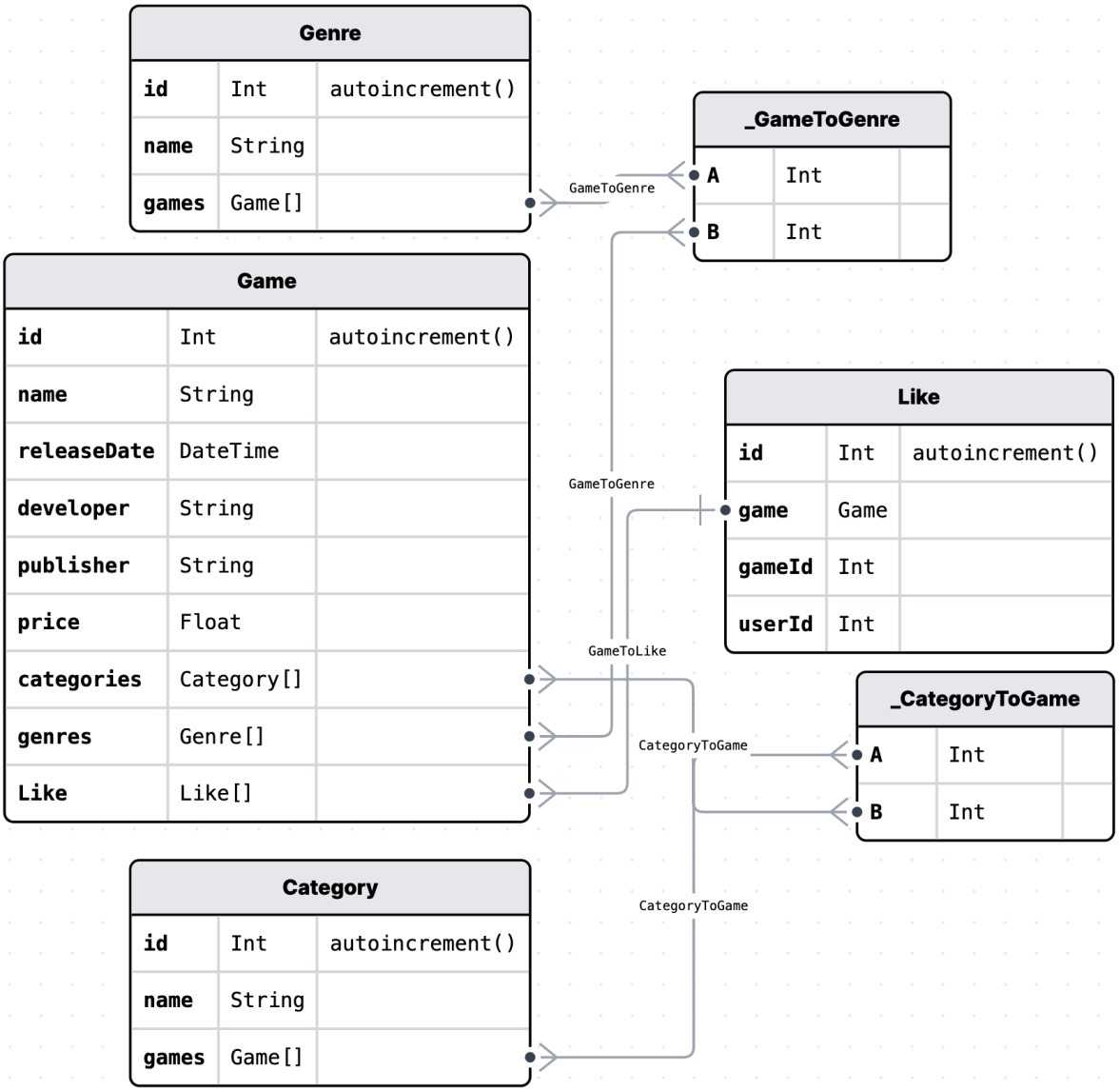


Схема базы данных games-сервиса



Файл GamesController.ts:

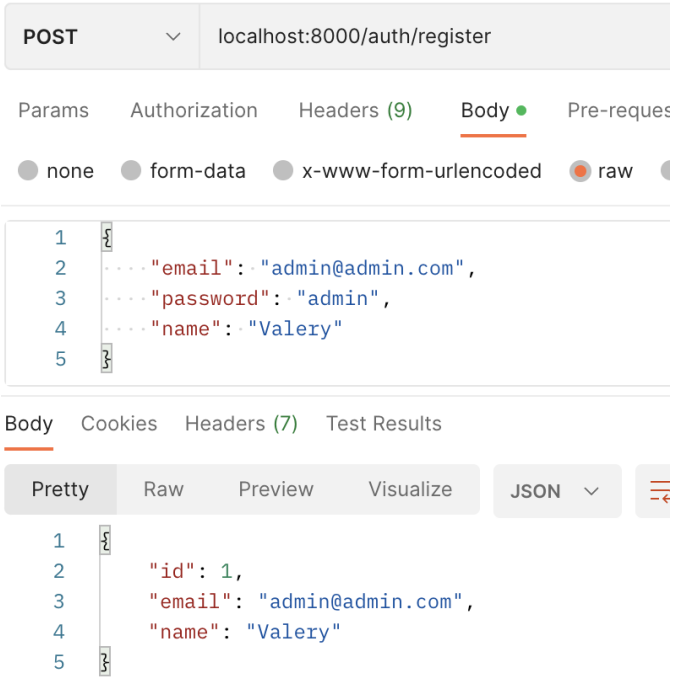
```
class GamesController {
    private gamesService: GamesService

    constructor() {
        this.gamesService = new GamesService()
    }

    getAll = async (req: Request, res: Response) => {
        let {offset, count, developer, publisher, sortByPrice} = req.query
        offset = offset ? offset : "0"
        count = count ? count : "10"
        sortByPrice = ["desc", "asc"].includes(String(sortByPrice)) ? String(sortByPrice) : undefined
        developer = developer ? String(developer) : undefined
        publisher = publisher ? String(publisher) : undefined
        const {
            total,
            result
        } = await this.gamesService.getAll(Number(count), Number(offset), developer, publisher, sortByPrice)
        res.json( body: {total, offset, count, result})
    }
}

export default GamesController;
```

Файл LikesController.ts

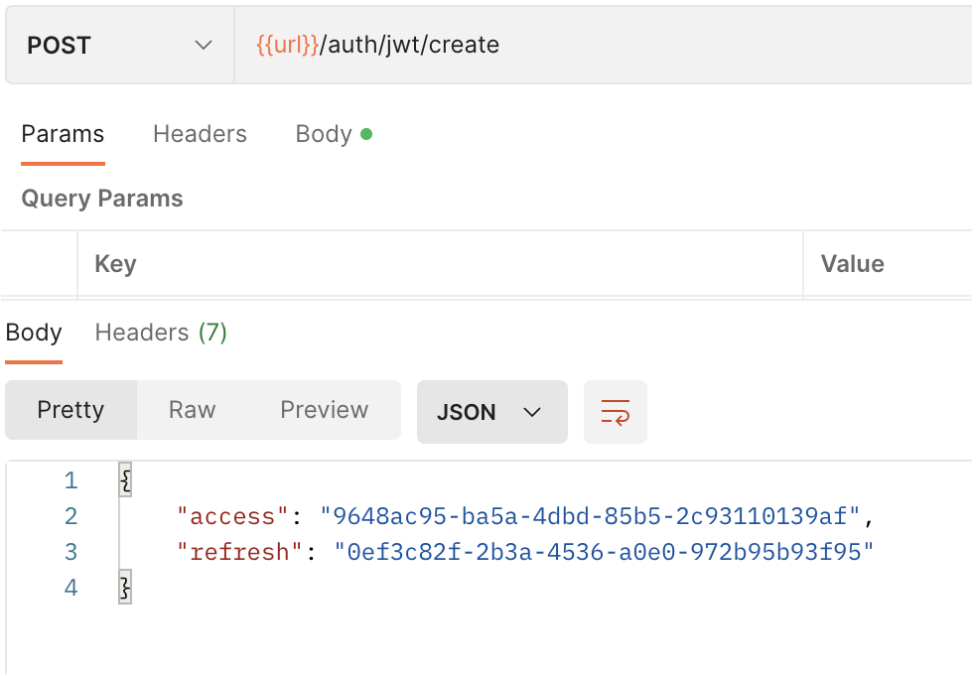
```
class LikesController {
    private LikesService: LikesService

    constructor() {
        this.LikesService = new LikesService()
    }

    get = async (req: Request, res: Response) => {
        try {
            const {"user-id": userId} = req.headers
            if (userId) {
                return res.json(await this.LikesService.get(Number(userId)))
            }
            return res.status( code: 412).json( body: {error: "User ID not in headers"})
        } catch (e: any) {
            return res.status( code: 404).json( body: {error: e.message})
        }
    }

    post = async (req: Request, res: Response) => {
        try {
            const {"user-id": userId} = req.headers
            const {gameId} = req.body
            if (userId) {
                return res.json(await this.LikesService.post(Number(gameId), Number(userId)))
            }
            return res.status( code: 412).json( body: {error: "User ID not in headers"})
        } catch (e: any) {
            return res.status( code: 404).json( body: {error: e.message})
        }
    }

    delete = async (req: Request, res: Response) => {
        try {
            const {"user-id": userId} = req.headers
            const {id: likeId} = req.params
            if (userId) {
                return res.json(await this.LikesService.delete(Number(likeId)))
            }
            return res.status( code: 412).json( body: {error: "User ID not in headers"})
        } catch (e: any) {
            return res.status( code: 404).json( body: {error: e.message})
        }

    }
}

export default LikesController;
```

ДЕМОНСТРАЦИЯ

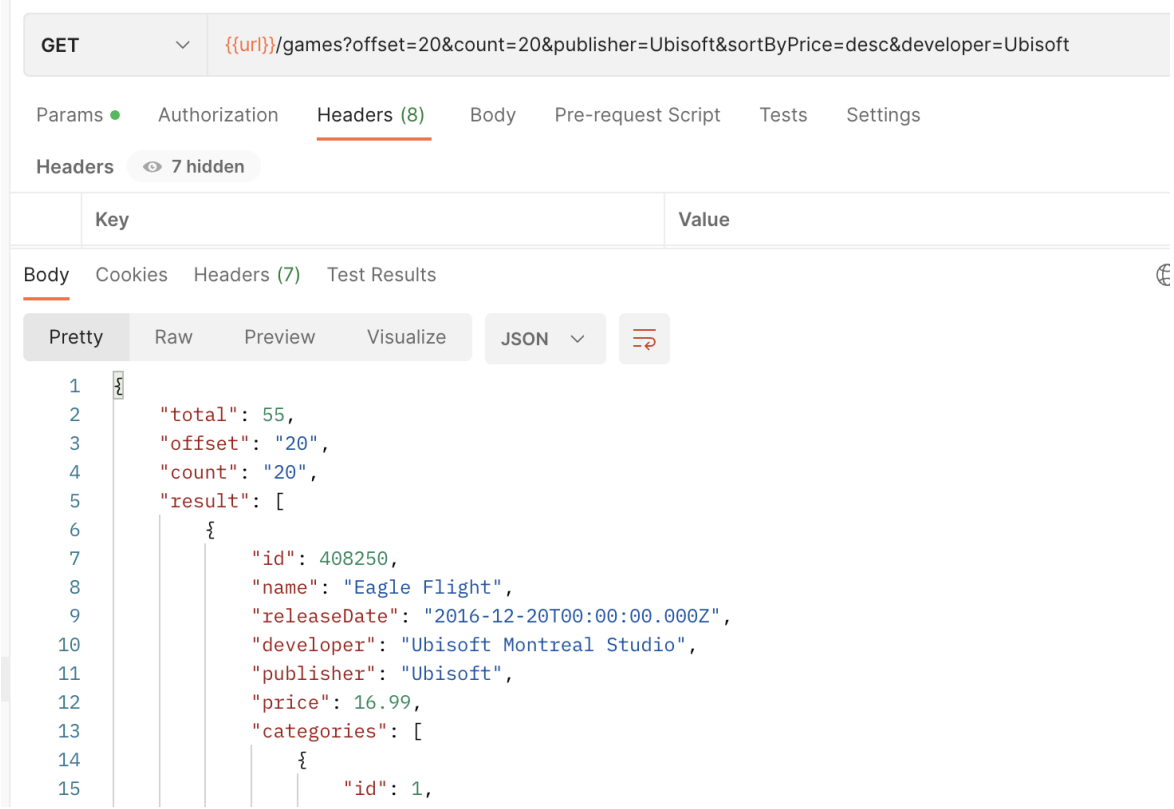
Регистрация:



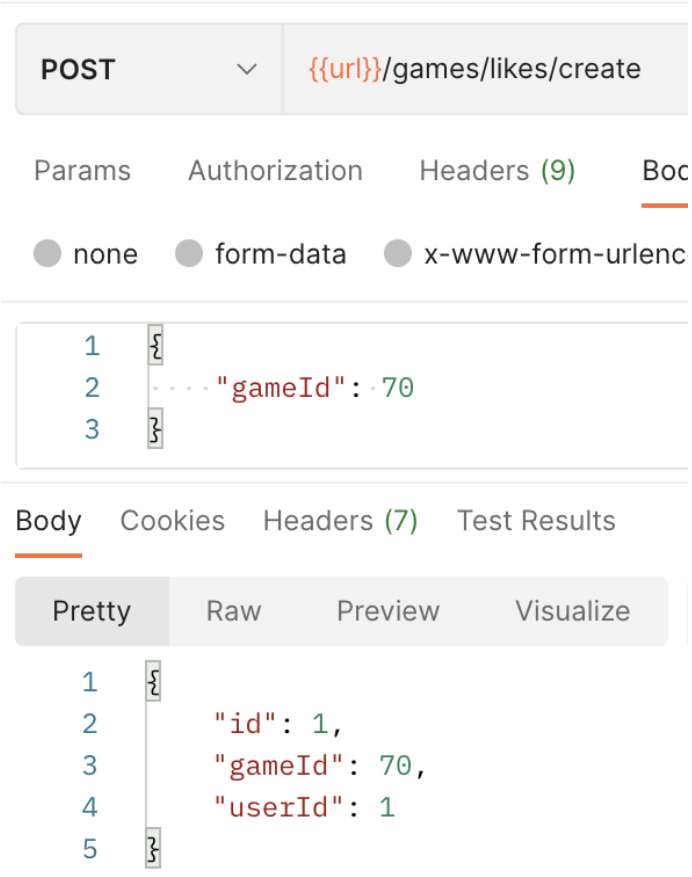
Получение токена:



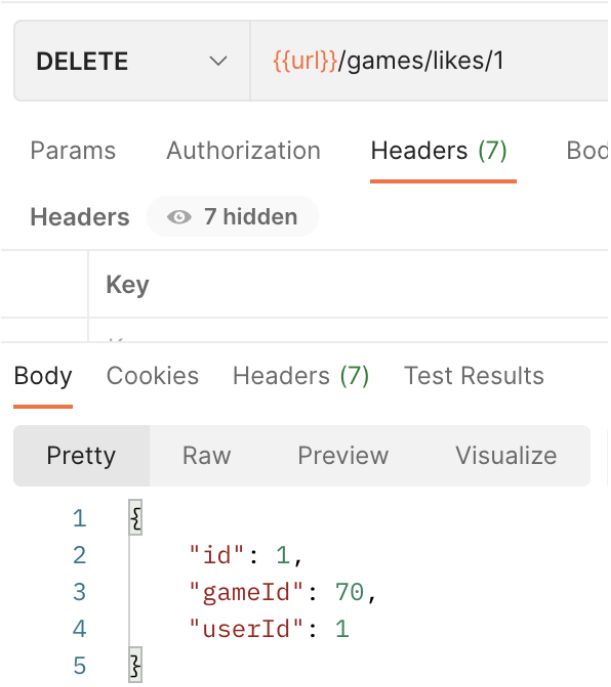
Получение списка игр (доступна пагинация):



Поставить лайк игре:



Удалить лайк:



Полная документация с примерами доступна по ссылке

ВЫВОД

В результате выполнения лабораторной работы я получил RESTfull API.