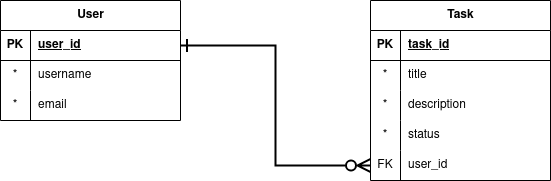
Лабораторная работа №4

Rest api сервиса личного списка задач (Todo service)

ER диаграммы БД:



Крайне простая схема БД

**User:**

**user\_id** - строка формата UUID, первичный ключ

**username** - логин пользователя - строка длины 36, обязательна для заполнения, не NULL

**email** - электронный адрес - строка формата <имя пользователя>@<домен> длины 64, обязательна для заполнения, не NULL

**Task:**

**task\_id** - строка формата UUID, первичный ключ

**title** - тайтл задачи - строка длины 512, обязательна для заполнения, не NULL

**description** - описание задачи - строка длины 2048, обязательна для заполнения, не NULL, начальное значение = “”

**status** - статус задачи - строка длины 64, обязательна для заполнения, не NULL, начальное значение = “CREATED”

**user\_id** - id пользователя - строка формата UUID, внешний ключ, обязателен для заполнения

Используемая СУБД - H2 database

Название БД: todo.bd, файл создаётся в папке пользователя

User: su

Password: password

Есть UI консоль, доступная по пути <http://localhost:8080/h2-console>

Также можно посмотреть все запросы при помощи swagger по пути http://localhost:8080/swagger-ui.html

Описание структуры работы:

**aspect** - папка с классами, описывающими аспекты (AOP) (только один аспект - заменяет все lazy списки, которые не были проинициализированы, на пустые списки, чтобы не было ошибки **LazyInitializationException**)

**dto** - папка с классами, описывающими все транспортные сущности

**controller** - папка с классами, описывающие контроллеры (ControllerExceptionAdvice перехватывает важные ошибки в запросах и отправляет их пользователю в более удобном виде)

**exception** - папка с классами, описывающими кастомные ошибки

**model** - папка с сущностями БД

**repository** - папка с классами, описывающими слой доступа к СУБД

**service** - папка с классами, описывающие бизнес слой и слой сервиса, подразделяется на интерфейсы, оканчивающиеся на Service, и папку implementation, в которой реализуются эти интерфейсы

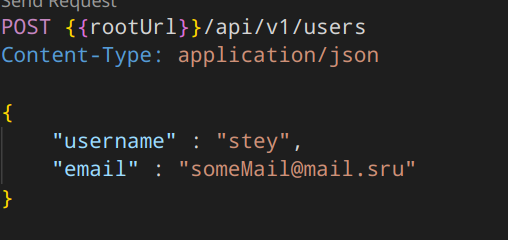
Все запросы описаны в файле requests.http

Запускалось при помощи плагина REST Client в Visual Studio Code

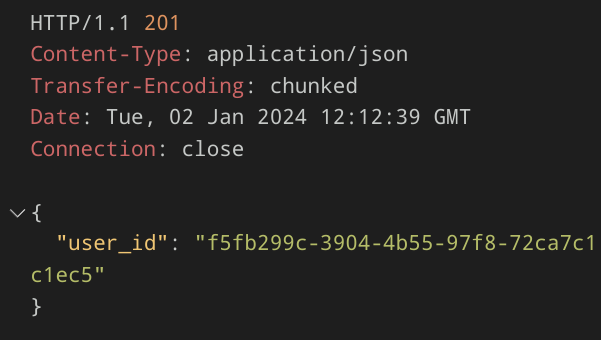
{{rootUrl}} - переменная хоста сервера с портом (Стандартно <http://localhost:8080>)

Запросы:

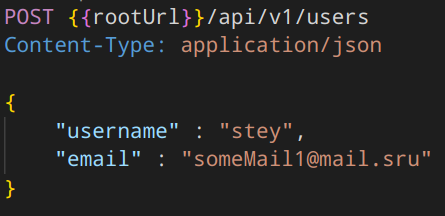
Запрос на добавление пользователя:



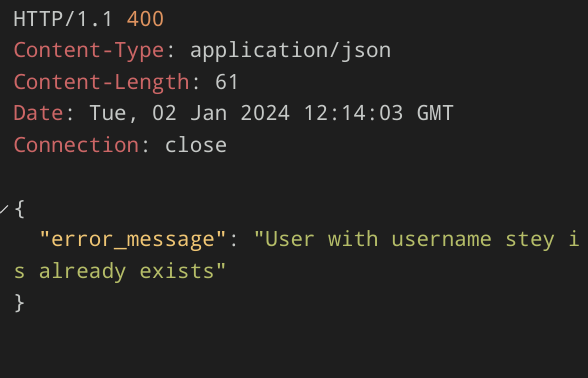
Ответ:



Запрос на добавление пользователя с существующим username:

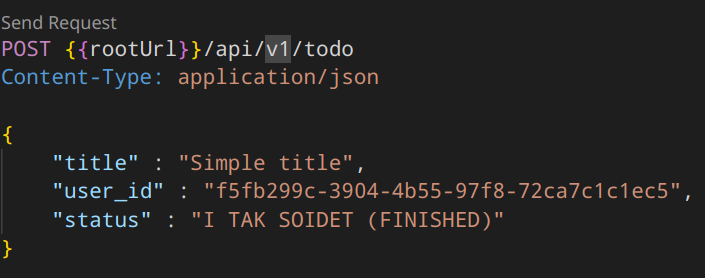


Ответ:

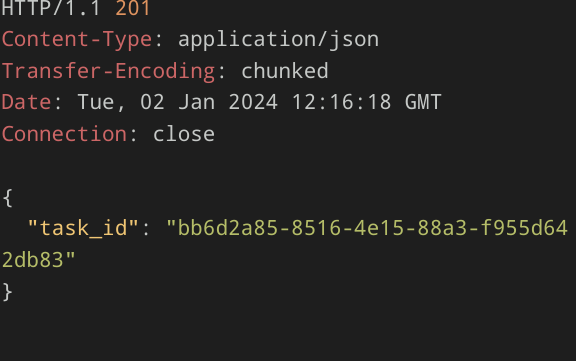


Аналогично будет и для email

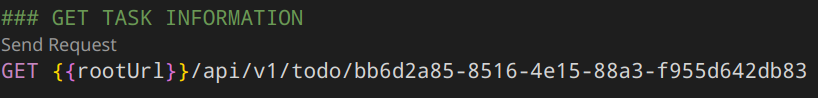
Запрос на создание задачи для пользователя:



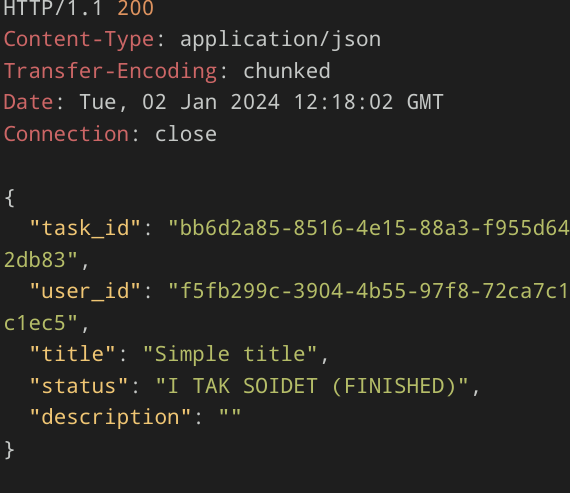
Ответ:



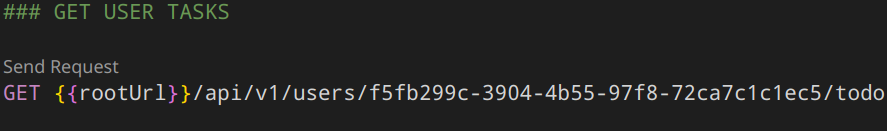
Запрос на получение информации о задаче:



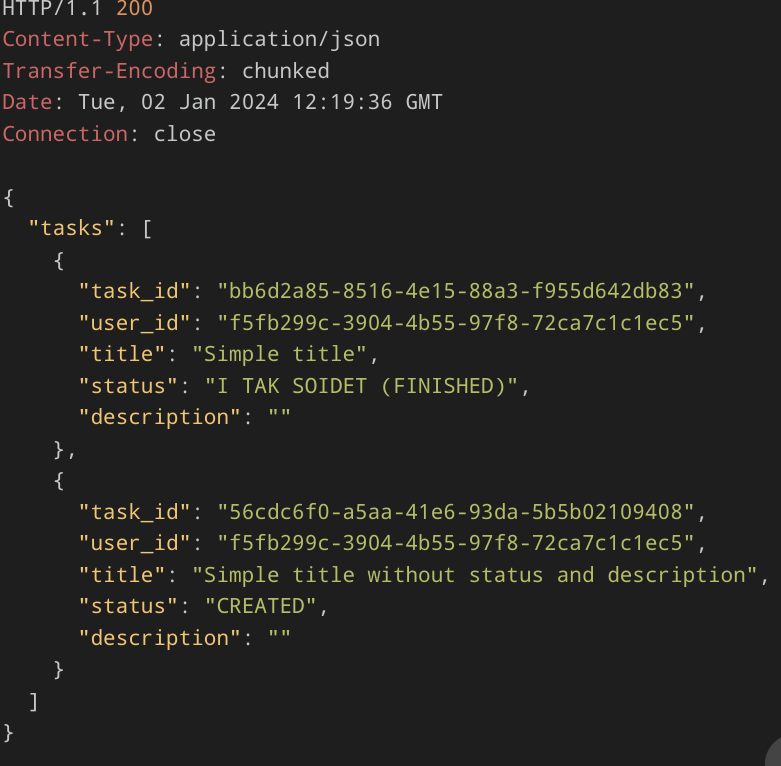
Ответ:



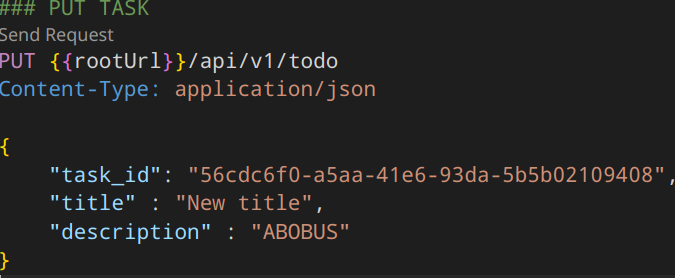
Запрос на получение задач пользователя:



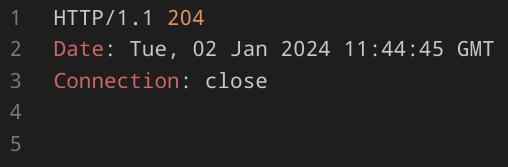
Ответ:



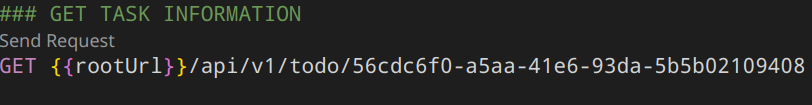
Запрос на обновление задачи:



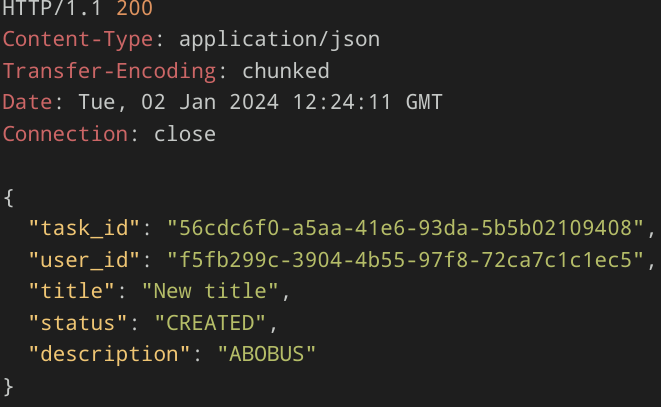
Ответ:



Запрос на получение информации изменённой задачи:



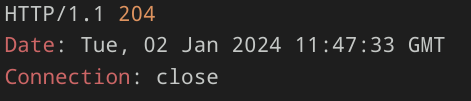
Ответ:



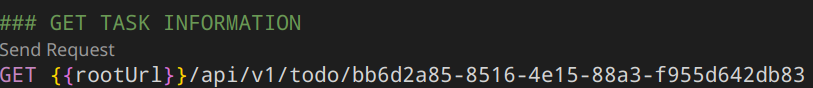
Запрос на удаление задачи:



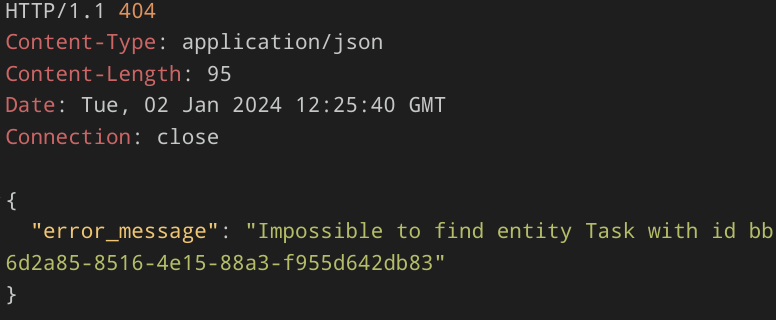
Ответ:



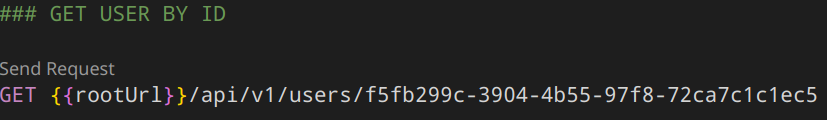
Запрос на получение информации об удалённой задачи:



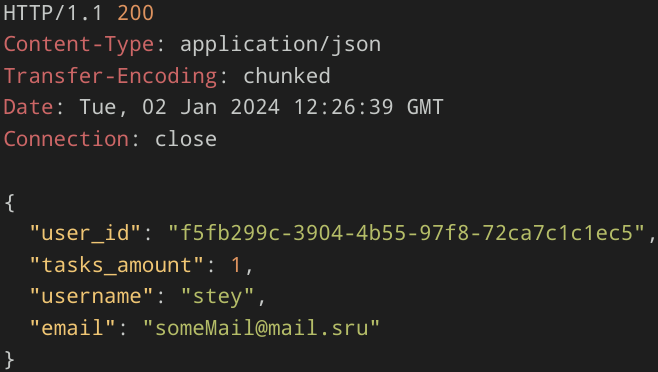
Ответ:



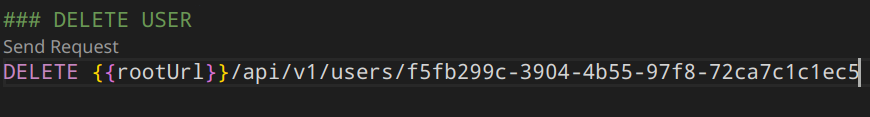
Запрос на получение информации о пользователе



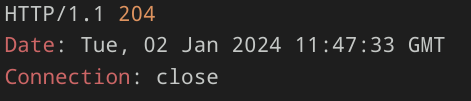
Ответ:



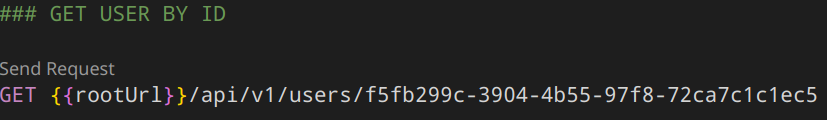
Запрос на удаление пользователя с задачами:



Ответ:



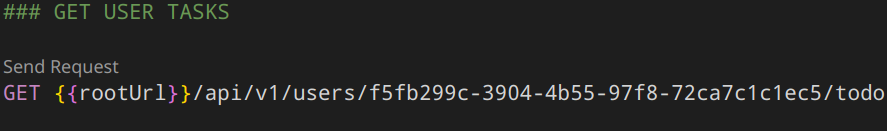
Запрос на получение информации об удалённом пользователе:



Ответ:



Запрос на получение задач удалённого пользователя:



Ответ:

