Это Spring Boot-приложение для управления транзакциями и пользователями. Приложение поддерживает операции создания, редактирования, удаления и просмотра транзакций, работу с пользователями, а также логирование изменений и простую интеграцию с Swagger UI для тестирования API.

**Как пользоваться**

**1. Запуск приложения**

* Убедитесь, что PostgreSQL запущен и создана база данных kata
* Отредактируйте application.properties при необходимости (логин/пароль БД, порт и т.д.)
* Запустите приложение как обычный Spring Boot проект:

mvn spring-boot:run

* После запуска Swagger будет доступен по адресу:
* http://localhost:8080/swagger-ui.html

**Регистрация пользователя:**

Post <http://localhost:8080/api/users/register>

**Пример:**

**Запрос:**

  "username": "gomic58",

  "password": "23",

  "email": "gomic58.com",

  "firstName": "Jdfgdfgdfgdfg",

  "lastName": "Doааe"

**Ответ:**

"id": 59,

    "username": "gomic58",

    "email": "gomic58.com",

    "firstName": "Jdfgdfgdfgdfg",

    "lastName": "Doааe"

Далее делаем запрос Post <http://localhost:8080/api/auth/login>

**Тело:**

"username": "gomic58",

"password": "23"

**Ответ:**

"token": "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJnb21pYzU4IiwiaWF0IjoxNzQ3NzE0NTk0LCJleHAiOjE3NDc4MDA5OTR9.2jPmU6qrrXTrIPucqX6DWI5N1QuJj8RjGEcGN4jEzhWZw4IhyeEDecJJem9eCNNPhNp\_OwteNzHN8Pyqndkqvg",

"type": "Bearer",

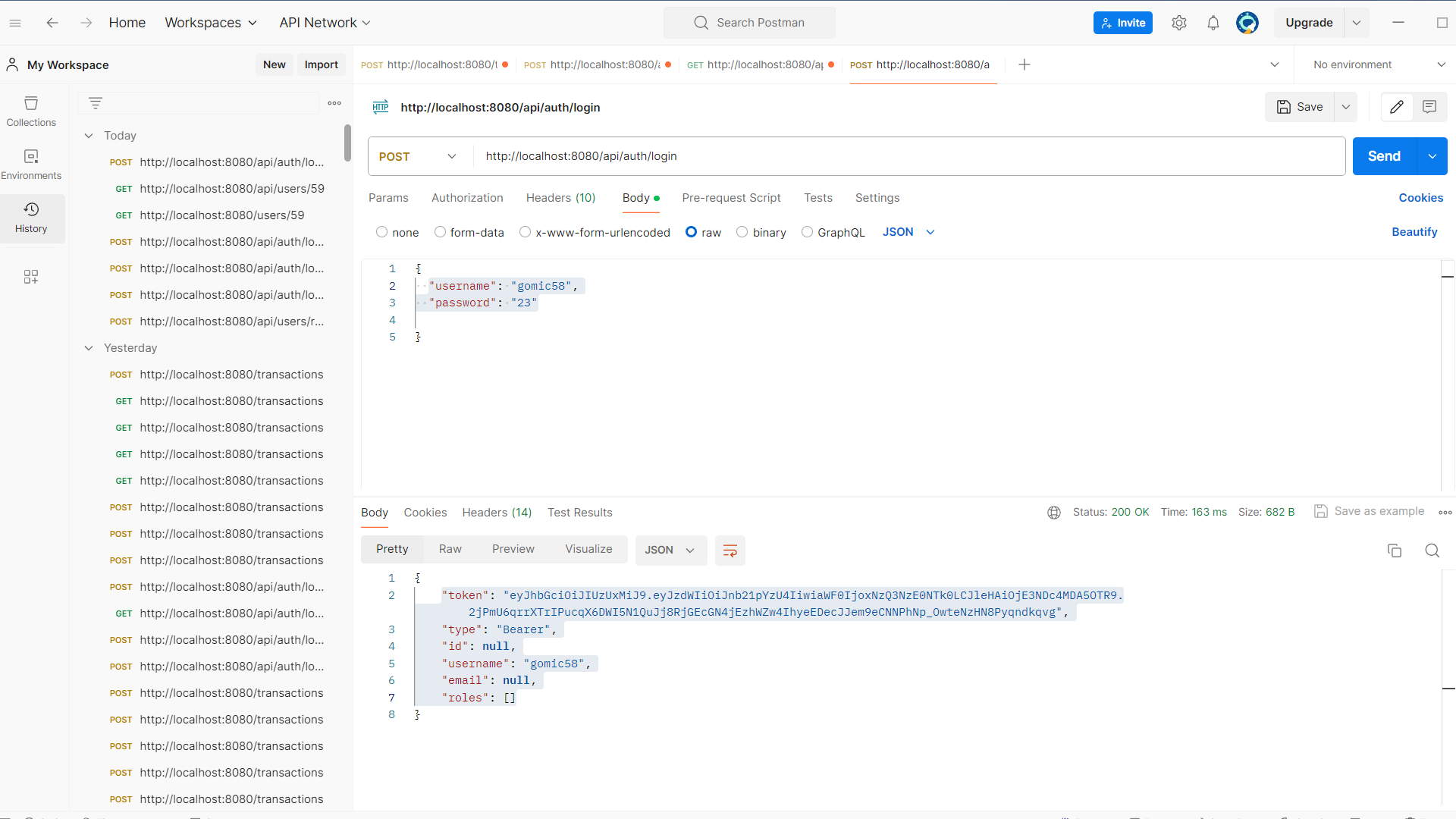
"id": null,

"username": "gomic58",

"email": null,

"roles": []

Далее копируем "token" из ответа:



И заходи в Headers. В **Key** прописываем **Authorization.**  В **Value** прописываем **Bearer** и токен.

Все, теперь можно работать с пользователем и транзакциями!

**Работа с пользователями**

**🔗 Основные эндпоинты пользователя:**

| **Метод** | **URL** | **Описание** | **Тело запроса / ответа** |
| --- | --- | --- | --- |
|  |  |  |  |
| GET | /api/users/{id} | Получить пользователя по id | - |
| PUT | /api/users/{id} | Обновить пользователя | UserRequest → UserResponse |
| DELETE | /api/users/{id} | Удалить пользователя | - |

**Работа с транзакциями**

**Основные эндпоинты транзакций:**

| **Метод** | **URL** | **Описание** | **Тело запроса / ответа** |
| --- | --- | --- | --- |
| POST | /api/transactions | Создать транзакцию | TransactionRequest |
| GET | /api/transactions/{id} | Получить транзакцию по id | - |
| PUT | /api/transactions/{id} | Обновить транзакцию | TransactionRequest |
| DELETE | /api/transactions/{id} | Удалить транзакцию (логическое удаление) | - |

**Пример тела запроса на создание (TransactionRequest):**

{

"userId": 59,

"personType": "Физическое лицо",

"transactionType": "Поступление",

"comment": "Премия",

"amount": 1000,

"status": "Подтвержденная",

"senderBank": "хер",

"senderAccount": "12348678",

"recipientBank": "кер",

"recipientInn": "123458789012",

"recipientAccount": "89654321",

"category": "Зарплата",

"recipientPhone": "+79991234567" }

**Ответ:**

"id": 24,

    "personType": "Физическое лицо",

    "transactionDate": [

        2025,

        5,

        20,

        7,

        31,

        2,

        55538200

    ],

"transactionType": "Поступление",

    "comment": "Премия",

    "amount": 1000,

    "status": "Подтвержденная",

    "senderBank": "хер",

    "senderAccount": "12348678",

    "recipientBank": "кер",

    "recipientInn": "123458789012",

    "recipientAccount": "89654321",

    "category": "Зарплата",

    "recipientPhone": "+79991234567",

    "createdAt": [

        2025,

        5,

        20,

        7,

        31,

        2,

        99387800

    ],

    "updatedAt": null

**А теперь по строкам в телах запроса POST /api/transactions:**

В **PersonType** можно указать только **"Физическое лицо"** или **"Юридическое лицо".**

В **CategoryType** можно выбрать следующее: Зарплата, Дивиденды, Проценты по вкладам,

Аренда, Подарок, Прочие доходы - **Это категория Поступления( "transactionType": "Поступление",)**

Продукты, Жилье, Транспорт, Развлечения, Здравоохранение, Образование,

Одежда, Коммунальные услуги, Выплата долга, Сбережения, Благотворительность,

Прочие расходы – **Это категория Списание (**     "transactionType": "Списание",

**)**

В **TransactionStatus** можно указать: Новая, Подтвержденная, В обработке, Отменена,

Платеж выполнен, Платеж удален, Возврат

В **TransactionType** можно выбрать либо **Поступление** либо **Списание.**

**Примечания по статусам и операциям:**

Вносить изменения в транзакции можно когда "status" = Новая

Статусы транзакций, невозможные к удалению: подтвержденная, в обработке, отменена, платеж выполнен, возврат

**Редактирование данных:**

1. Не допущены к редактированию операции со статусами: подтвержденная, в обработке, отменена, платеж выполнен, платеж удален, возврат

2. Параметры для редактирования: тип лица, дата, комментарий к операции, сумма операции, статус операции, банк отправителя, банк получателя, инн, категории, номер телефона

**Дашборды**

* **Веб-интерфейс дашбордов пока не реализован**
* Данные для дашбордов можно получать через API, фильтруя транзакции по статусу, дате, категории и т.д.
* Для визуализации и анализа используйте сторонние BI-инструменты или экспортируйте данные вручную.

**Логирование**

* Логи пишутся в файл logs/kata.log
* Изменения транзакций логируются в отдельной таблице (через IntegrationLog)

**Фильтрация транзакций**

**Пример запроса: Получить все транзакции со статусом NEW за определённый период и категорией SERVICE**

* **Метод:** GET
* **URL:**
* http://localhost:8080/api/transactions?status=NEW&category=SERVICE&fromDate=2024-01-01&toDate=2024-06-30
* **Описание параметров:**
  + status — статус транзакции (NEW, DELETED, COMPLETED и т.д.)
  + category — категория транзакции (SERVICE, PRODUCT, и т.п.)
  + fromDate — начальная дата (в формате YYYY-MM-DD)
  + toDate — конечная дата (в формате YYYY-MM-DD)

**Пример PATCH-запроса в Postman**

* **Метод:** PATCH
* **URL:**
* http://localhost:8080/api/transactions/{id}

где {id} — идентификатор нужной транзакции.

* **Тело запроса (JSON):**  
  Укажите только те поля, которые хотите изменить. Например, меняем комментарий и категорию:
* {
* "comment": "Изменённый комментарий",
* "category": "PRODUCT"

}

**Как отправить PATCH-запрос в Postman:**

1. Создайте новый запрос, выберите метод PATCH.
2. Введите URL вида http://localhost:8080/api/transactions/123 (где 123 — id транзакции).
3. В Body выберите raw и формат JSON, введите нужные поля.
4. Нажмите "Send".

**Дашборды:**

1. Динамика по количеству транзакций

GET/api/dashboard/period

1. Динамика по типу транзакции

GET/api/dashboard/type-period

3. Сравнение поступивших и потраченных

GET/api/dashboard/debit-credit

4. Количество проведённых и отменённых транзакций

GET/api/dashboard/status

1. Статистика по банкам

GET /api/dashboard/banks

6. Статистика по категориям расходов и поступлений

GET/api/dashboard/categories