

1. Регистрация пользователя

- **Endpoint:** POST /api/registration
- **Описание:** Регистрация нового пользователя.
- **Логика работы:**
 - Пользователь отправляет данные (email, телефон, пароль).
 - Система проверяет, что email и телефон уникальны (не используются другим пользователем).
 - Если данные корректны, система создает нового пользователя в базе данных.
 - Пользователь получает подтверждение успешной регистрации.
- **Связь с БД:**
 - Запрос к таблице users для проверки уникальности email и телефона.
 - Вставка данных в таблицу users для создания нового пользователя.
- **Валидация:**
 - Проверка, что email и телефон уникальны.
 - Проверка, что пароль соответствует требованиям (минимум 8 символов).
 - Проверка формата email и телефона.
- **Параметры запроса (JSON):**

```
{  
  "email": "user@example.com",  
  "phone": "+79123456789",  
  "password": "password123"  
}
```
- **Ответ (успех):**

```
{  
  "status": "success",  
  "message": "User registered successfully",  
  "userId": 1  
}
```
- **Ответ (ошибка):**

```
{  
  "status": "error",  
  "message": "Email already exists"  
}
```

2. Авторизация пользователя

- **Endpoint:** POST /api/login
- **Описание:** Вход пользователя в систему.
- **Логика работы:**
 - Пользователь отправляет телефон и пароль.
 - Система проверяет, существует ли пользователь с таким телефоном.
 - Если пользователь существует, система проверяет, совпадает ли пароль.
 - Если данные верны, система генерирует токен (JWT) для авторизации.
 - Пользователь получает токен для доступа к защищенным ресурсам.
- **Связь с БД:**
 - Запрос к таблице users для поиска пользователя по телефону.
 - Проверка пароля (сравнение хэшированного пароля в БД).
- **Валидация:**
 - Проверка, что телефон существует в системе.
 - Проверка, что пароль совпадает с сохраненным в БД.
- **Параметры запроса (JSON):**

```
{  "phone": "+79123456789",  "password": "password123"}
```
- **Ответ (успех)**

```
{  "status": "success",  "message": "Login successful",  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."}
```
- **Ответ (ошибка)**

```
{  "status": "error",  "message": "Invalid credentials"}
```

3. Получение списка фильмов

- **Endpoint:** GET /api/movies
- **Описание:** Получение списка доступных фильмов.
- **Ответ (успех):**

```
{
```

```
"status": "success",
"movies": [
  {
    "id": 1,
    "title": "Inception",
    "genre": "Sci-Fi",
    "description": "A thief who steals corporate secrets...",
    "showtimes": [
      {
        "id": 101,
        "time": "2023-10-15T18:00:00",
        "availableSeats": 50
      }
    ]
  }
]
```

4. Бронирование билета

- **Endpoint:** POST /api/book
- **Описание:** Бронирование билета на выбранный сеанс.
- **Параметры запроса (JSON):**

```
{
  "userId": 1,
  "showtimeId": 101,
  "seats": ["A1", "A2"]
}
```

- **Ответ (успех):**

```
{
  "status": "success",
  "message": "Tickets booked successfully",
  "bookingId": 456
}
```

- **Ответ (ошибка):**

```
{
  "status": "error",
  "message": "Seats are already taken"
}
```

5. Отправка отзыва

- **Endpoint:** POST /api/feedback
- **Описание:** Отправка отзыва о фильме.
- **Параметры запроса (JSON):**

```
{  
  "userId": 1,  
  "movieId": 1,  
  "rating": 5,  
  "comment": "Amazing movie!"  
}
```

- **Ответ (успех):**

```
{  
  "status": "success",  
  "message": "Feedback submitted successfully"  
}
```