**API Documentation**

**Base URL**

http://localhost:8080

**User Management API**

**1. Get User**

1. **Endpoint**: /users/{id}
2. **Method**: GET
3. **Description**: Получает информацию о пользователе по его ID.
4. **Path Parameters**:
   1. id (integer): ID пользователя, который нужно получить.
5. **Responses**:
   1. 200 OK: Успешный ответ с данными пользователя.
   2. 404 Not Found: Пользователь с указанным ID не найден.
   3. 400 Bad Request: Неверный формат ID.

**Example Request**

GET /users/1 HTTP/1.1

Host: localhost:8080

**Example Response**

{

"id": 1,

"name": "John Doe"

}

**2. Update User**

1. **Endpoint**: /users/{id}
2. **Method**: PUT
3. **Description**: Обновляет информацию о пользователе по его ID.
4. **Path Parameters**:
   1. id (integer): ID пользователя, которого нужно обновить.
5. **Request Body**:
   1. name (string): Новое имя пользователя.
6. **Responses**:
   1. 200 OK: Успешное обновление пользователя.
   2. 404 Not Found: Пользователь с указанным ID не найден.
   3. 400 Bad Request: Неверный формат данных.

**Example Request**

PUT /users/1 HTTP/1.1

Host: localhost:8080

Content-Type: application/json

{

"name": "Jane Doe"

}

**Example Response**

HTTP/1.1 200 OK

**3. Delete User**

1. **Endpoint**: /users/{id}
2. **Method**: DELETE
3. **Description**: Удаляет пользователя по его ID.
4. **Path Parameters**:
   1. id (integer): ID пользователя, которого нужно удалить.
5. **Responses**:
   1. 200 OK: Успешное удаление пользователя.
   2. 404 Not Found: Пользователь с указанным ID не найден.
   3. 400 Bad Request: Неверный формат ID.

**Example Request**

DELETE /users/1 HTTP/1.1

Host: localhost:8080

**Example Response**

HTTP/1.1 200 OK

**Error Responses**

В случае ошибки API возвращает JSON-объект с описанием ошибки.

**Example Error Response**

{

"error": "User not found"

}