Josh Talk

Task Management API Documentation

This API allows for creating tasks, assigning tasks to users, and retrieving tasks assigned to a specific user.

Base URL:

- **Development Environment:** `http://127.0.0.1:8000/`

API Endpoints

1. Create a Task

- **Endpoint:** `POST /tasks/`
- **Description:** Creates a new task with a name, description, task type, and status.

Request Body:

Plain Text

```
{
    "name": "Excel Reporting",
    "description": "This is a Excel Reporting task.",
    "task_type": "General",
    "status": "pending",
    "assigned_user_ids": [1]    or [1,2,3](multiple Users)
}
```

Response Body:

- **Success (201 Created):**

Plain Text

```
{
    "id": 3,
    "name": "Excel Reporting",
    "description": "This is a Excel Reporting task.",
    "created_at": "2025-03-25T06:06:41.791634Z",
    "task_type": "General",
    "completed_at": null,
    "status": "pending",
```

- **Failure (400 Bad Request)**:

Plain Text

```
{
    "error": "Invalid input"
}
```

2. Retrieve Tasks Assigned to a User

- **Endpoint**: `GET /users/{user_id}/tasks/`
- **Description**: Retrieves all tasks assigned to a specific user.

Request Body:

- **No request body required.** Only the `user_id` is passed in the URL.

Response Body:

- **Success (200 OK)**:

Plain Text

```
[
    {
        "id": 1,
        "name": "Report completion",
        "description": "Demo Task to complete Report",
        "created_at": "2025-03-24T12:07:46.116435Z",
        "task_type": "Documentation",
        "completed_at": "2025-03-24T12:07:41Z",
```

- **Failure (404 Not Found)**:

Plain Text

```
{
    "detail": "No User matches the given query."
}
```

---

3. List All Tasks

- **Endpoint**: `GET /tasks/`

- **Description**: Retrieve a list of all tasks.

Request Body:

- **No request body required.**

Response Body:

- **Success (200 OK)**:

Plain Text

```
[
    {
        "id": 1,
        "name": "Report completion",
        "description": "Demo Task to complete Report",
        "created_at": "2025-03-24T12:07:46.116435Z",
        "task_type": "Documentation",
        "completed_at": "2025-03-24T12:07:41Z",
```

---

4. Retrieve a Specific Task

- **Endpoint**: `GET /tasks/{task_id}/`
- **Description**: Retrieve a specific task by its ID.

Request Body:

- **No request body required.**

Response Body:

- **Success (200 OK)**:

Plain Text

```
{
    "id": 1,
    "name": "Report completion",
    "description": "Demo Task to complete Report",
    "created_at": "2025-03-24T12:07:46.116435Z",
    "task_type": "Documentation",
    "completed_at": "2025-03-24T12:07:41Z",
    "status": "pending",
```

- **Failure (404 Not Found)**:

Plain Text

```
{
    "detail": "No Tasks matches the given query."
}
```

---

5. List All Users

- **Endpoint**: `GET /users/`
- **Description**: List all users.

Request Body:

- **No request body required.**

Response Body:

- **Success (200 OK)**:

Plain Text

```
[
    {
        "id": 1,
        "name": "User 1",
        "email": "user1@example.com",
        "mobile": "1234567890"
    },
    {
```

---

6. Retrieve a Specific User

- **Endpoint**: `GET /users/{user_id}/`
- **Description**: Retrieve a specific user by ID.

Request Body:

- **No request body required.**

Response Body:

- **Success (200 OK)**:

Plain Text

```
{
    "id": 1,
    "name": "Pankaj Kain",
    "email": "pankaj123@gmail.com",
    "mobile": "1234567891"
}
```

- **Failure (404 Not Found)**:

Plain Text

```
{
    "detail": "No User matches the given query."
}
```

Authentication

This API does not include any authentication mechanism by default. You can add authentication in your Django settings, such as **Token Authentication** or **Session Authentication**, depending on your needs.