

Postman API Documentation is described below: -

To document the API using Postman, we will create a collection of requests for each endpoint and include details about the expected input and output, along with example requests and responses.

Collection: Task Management API

1.Folder: User Authentication

- **Request: User Registration**

- **Method:** POST
- **URL:** /api/register/
- **Description:** Registers a new user with a username and password.
- **Headers:** Content-Type: application/json
- **Body (JSON):-**

```
{  
  
  "username": "user1",  
  
  "password": "password123"  
}
```

- **Response (200 OK) :**

```
{  
  
  "refresh": "string",  
  
  "access": "string"  
}
```

2. Folder: Task Management

Request: Create Task

- **Method:** POST
- **URL:** /api/tasks/
- **Description:** Creates a new task for the authenticated user.
- **Headers:**
 - Content-Type: application/json
 - Authorization: Bearer <access_token>
- **Body (JSON)**

```
{
```

```
"title": "New Task",

"description": "Task description",

"status": "todo",

"priority": "high",

"due_date": "2024-07-31"

}
```

- **Response** (201 Created)

```
{

  "id": 1,

  "title": "New Task",

  "description": "Task description",

  "status": "todo",

  "priority": "high",

  "due_date": "2024-07-31",

  "created_at": "2024-01-01T00:00:00Z",

  "updated_at": "2024-01-01T00:00:00Z",

  "user": 1

}
```

Request: Get Tasks

- **Method:** GET
- **URL:** /api/tasks/
- **Description:** Retrieves a list of tasks for the authenticated user. Supports filtering by status, priority, and due date.
- **Headers:** Authorization: Bearer <access_token>
- **Query Parameters** (Optional):
 - status: Filter by task status (e.g., todo, in_progress, done)
 - priority: Filter by task priority (e.g., low, medium, high)
 - due_date: Filter by due date (e.g., 2024-07-31)

- **Response (200 OK)**

```
[  
  
  {  
  
    "id": 1,  
  
    "title": "New Task",  
  
    "description": "Task description",  
  
    "status": "todo",  
  
    "priority": "high",  
  
    "due_date": "2024-07-31",  
  
    "created_at": "2024-01-01T00:00:00Z",  
  
    "updated_at": "2024-01-01T00:00:00Z",  
  
    "user": 1  
  
  }  
  
]
```

- **Request: Update Task**

- **Method:** PUT
- **URL:** /api/tasks/{taskId}/
- **Description:** Updates an existing task for the authenticated user.
- **Headers:**
 - Content-Type: application/json
 - Authorization: Bearer <access_token>

- **Body (JSON)**

```
{  
  
  "title": "Updated Task",  
  
  "description": "Updated description",  
  
  "status": "in_progress",  
  
  "priority": "medium",  
  
  "due_date": "2024-07-31"
```

```
}
```

- **Response (200 OK)**

```
{
```

```
  "id": 1,
```

```
  "title": "Updated Task",
```

```
  "description": "Updated description",
```

```
  "status": "in_progress",
```

```
  "priority": "medium",
```

```
  "due_date": "2024-07-31",
```

```
  "created_at": "2024-01-01T00:00:00Z",
```

```
  "updated_at": "2024-01-02T00:00:00Z",
```

```
  "user": 1
```

```
}
```

Request: Delete Task

- **Method:** DELETE
- **URL:** /api/tasks/{taskId}/
- **Description:** Deletes a task for the authenticated user.
- **Headers:** Authorization: Bearer <access_token>
- **Response (204 No Content):** No content returned on successful deletion.

How to Import This Documentation into Postman?

1. **Open Postman:** Launch Postman and navigate to the "Collections" tab.
2. **Create a New Collection:** Click the "New" button, select "Collection", and name it "Task Management API".
3. **Add Requests to the Collection:**
 - For each endpoint described above, click "Add Request" under the collection.
 - Set the request method, URL, headers, and body as described.
 - Include example requests and responses in the "Example" section for each request.
4. **Save the Collection:** After adding all requests, save the collection.
5. **Export the Collection:**
6. Click the three dots next to the collection name and select "Export".
 - Choose "Collection v2.1 (recommended)" format and save the file.
7. **Share the Collection:**

- Share the collection with your team or stakeholders by importing the exported file or by generating a Postman collection link.

By following these steps, you can effectively document and share your API using Postman, allowing for easy testing and collaboration.