KanMind API - Postman Testing Guide

Base URL: `http://127.0.0.1:8000`

Setup

1. Start the server: `python manage.py runserver`
2. In Postman, create a variable `base_url` with value `http://127.0.0.1:8000`
3. Register a user (see below) and copy the token from the response
4. For all protected endpoints, add this header:
   - Key: `Authorization`
   - Value: `Token <your-token>`

---

1. Authentication (no token needed)

1.1 Register

```
POST {{base_url}}/api/registration/
```

Body (JSON):

```json
{
  "fullname": "John Doe",
  "email": "john@example.com",
  "password": "secure123",
  "repeated_password": "secure123"
}
```

Expected: `201 Created` with token, user_id, email, fullname

1.2 Login

```
POST {{base_url}}/api/login/
```

Body (JSON):

```json
{
  "email": "john@example.com",
  "password": "secure123"
}
```

Expected: `200 OK` with token, user_id, email, fullname

---

2. User Endpoints (token required)

2.1 Logout

```
POST {{base_url}}/api/logout/
```

No body needed. Expected: `200 OK`

2.2 Profile

```
GET {{base_url}}/api/profile/
```

Expected: `200 OK` with user_id, username, email, fullname, date_joined

2.3 Current User (Me)

```
GET {{base_url}}/api/users/me/
```

Expected: `200 OK` with id, name, email

2.4 Email Check

```
GET {{base_url}}/api/email-check/?email=john@example.com
```

Expected: `200 OK` with id, email, fullname (or `404` if not found)

2.5 List All Users

```
GET {{base_url}}/api/users/
```

Expected: `200 OK` with list of users (id, username, email, fullname)

---

3. Boards (token required)

3.1 List Boards

```
GET {{base_url}}/api/boards/
```

Expected: `200 OK` with list of boards you own or are a member of

3.2 Create Board

```
POST {{base_url}}/api/boards/
```

Body (JSON):

```json
{
  "title": "My Project",
  "members": [2, 3]
}
```

Expected: `201 Created` with board data

3.3 Get Single Board

```
GET {{base_url}}/api/boards/1/
```

Expected: `200 OK` with board details, tasks and members

3.4 Update Board

```
PATCH {{base_url}}/api/boards/1/
```

Body (JSON):

```json
{
  "title": "Updated Title",
  "members": [2, 3, 4]
}
```

Expected: `200 OK` with updated board data

3.5 Delete Board

```
DELETE {{base_url}}/api/boards/1/
```

Expected: `204 No Content`

---

4. Tasks (token required)

4.1 List Tasks

```
GET {{base_url}}/api/tasks/
```

Expected: `200 OK` with list of tasks from your boards

4.2 Create Task

```
POST {{base_url}}/api/tasks/
```

Body (JSON):

```json
{
  "board": 1,
  "title": "Implement feature X",
  "description": "Build the new login page",
  "status": "to-do",
  "priority": "high",
  "assignee_id": 2,
  "reviewer_id": 3,
  "due_date": "2026-03-15"
}
```

Status options: `to-do`, `in-progress`, `review`, `done` Priority options: `low`, `medium`, `high`

Expected: `201 Created` with task data

4.3 Get Single Task

```
GET {{base_url}}/api/tasks/1/
```

Expected: `200 OK` with task details

4.4 Update Task

```
PATCH {{base_url}}/api/tasks/1/
```

Body (JSON):

```json
{
  "status": "in-progress",
  "priority": "high"
}
```

Expected: `200 OK` with updated task data

4.5 Delete Task

```
DELETE {{base_url}}/api/tasks/1/
```

Expected: `204 No Content`

4.6 Tasks Assigned to Me

```
GET {{base_url}}/api/tasks/assigned-to-me/
```

Expected: `200 OK` with list of tasks assigned to you

4.7 Tasks in Review

```
GET {{base_url}}/api/tasks/reviewing/
```

Expected: `200 OK` with list of tasks with status `review`

---

5. Comments (token required)

5.1 List All Comments

```
GET {{base_url}}/api/comments/
```

Expected: `200 OK` with list of comments from your boards

5.2 Create Comment

```
POST {{base_url}}/api/comments/
```

Body (JSON):

```json
{
  "task": 1,
  "content": "This looks good!"
}
```

Expected: `201 Created` with comment data

5.3 List Comments for a Task

```
GET {{base_url}}/api/tasks/1/comments/
```

Expected: `200 OK` with list of comments for that task

5.4 Create Comment on a Task

```
POST {{base_url}}/api/tasks/1/comments/
```

Body (JSON):

```json
{
  "content": "Please check the styling"
}
```

Expected: `201 Created` with comment data

5.5 Delete Comment

```
DELETE {{base_url}}/api/tasks/1/comments/3/
```

Expected: `204 No Content` (only the comment author can delete)

Quick Test Flow

1. **Register** a user -> copy the token
2. **Create a board** with the token
3. **Create a task** on that board
4. **Create a comment** on that task
5. **List the board** and check that tasks and comments are included
6. **Update the task** status to `in-progress`
7. **Check assigned-to-me** and **reviewing** endpoints
8. **Delete** the comment, then the task, then the board
9. **Logout**

Error Responses

Status	Meaning
400	Bad request (missing or invalid fields)
401	Not authenticated (missing or invalid token)
403	Forbidden (not the owner)
404	Not found
500	Server error