To-Do API Documentation

# Base URL

All endpoints are prefixed with /api/. If your server is running locally, the full base URL would be http://127.0.0.1:8000/api/.

# Authentication Endpoints

## 1. Register a New User

Endpoint: POST /api/register/

Description: Registers a new user with a username, email, and password.

Request Body:

{  
 "username": "testuser",  
 "email": "testuser@example.com",  
 "password": "yourpassword"  
}

Response:

Success (201 Created):

{  
 "id": 1,  
 "username": "testuser",  
 "email": "testuser@example.com"  
}

Failure (400 Bad Request):

{  
 "username": ["This field is required."],  
 "email": ["This field is required."],  
 "password": ["This field is required."]  
}

## 2. Login and Obtain JWT Tokens

Endpoint: POST /api/login/

Description: Authenticates the user and returns JWT tokens (access and refresh).

Request Body:

{  
 "username": "testuser",  
 "password": "yourpassword"  
}

Response:

Success (200 OK):

{  
 "refresh": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...",  
 "access": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...",  
 "user": {  
 "id": 1,  
 "username": "testuser",  
 "email": "testuser@example.com"  
 }  
}

Failure (401 Unauthorized):

{  
 "detail": "No active account found with the given credentials"  
}

## 3. Refresh JWT Access Token

Endpoint: POST /api/token/refresh/

Description: Refreshes the access token using the refresh token.

Request Body:

{  
 "refresh": "your\_refresh\_token"  
}

Response:

Success (200 OK):

{  
 "access": "new\_access\_token"  
}

Failure (401 Unauthorized):

{  
 "detail": "Token is invalid or expired"  
}

# To-Do Endpoints

## 1. Create a To-Do Item

Endpoint: POST /api/todos/

Description: Creates a new to-do item for the authenticated user.

Headers:

Authorization: Bearer <access\_token>

Request Body:

{  
 "title": "New Task",  
 "description": "This is a new task."  
}

Response:

Success (201 Created):

{  
 "id": 1,  
 "title": "New Task",  
 "description": "This is a new task.",  
 "completed": false,  
 "created\_at": "2024-09-01T12:00:00Z",  
 "updated\_at": "2024-09-01T12:00:00Z"  
}

Failure (400 Bad Request):

{  
 "title": ["This field is required."]  
}

## 2. List All To-Do Items

Endpoint: GET /api/todos/

Description: Retrieves a list of all to-do items created by the authenticated user.

Headers:

Authorization: Bearer <access\_token>

Response:

Success (200 OK):

[  
 {  
 "id": 1,  
 "title": "New Task",  
 "description": "This is a new task.",  
 "completed": false,  
 "created\_at": "2024-09-01T12:00:00Z",  
 "updated\_at": "2024-09-01T12:00:00Z"  
 },  
 {  
 "id": 2,  
 "title": "Another Task",  
 "description": "This is another task.",  
 "completed": false,  
 "created\_at": "2024-09-01T13:00:00Z",  
 "updated\_at": "2024-09-01T13:00:00Z"  
 }  
]

Failure (401 Unauthorized):

{  
 "detail": "Authentication credentials were not provided."  
}

## 3. Retrieve a Single To-Do Item

Endpoint: GET /api/todos/<id>/

Description: Retrieves the details of a specific to-do item by ID.

Headers:

Authorization: Bearer <access\_token>

Response:

Success (200 OK):

{  
 "id": 1,  
 "title": "New Task",  
 "description": "This is a new task.",  
 "completed": false,  
 "created\_at": "2024-09-01T12:00:00Z",  
 "updated\_at": "2024-09-01T12:00:00Z"  
}

Failure (404 Not Found):

{  
 "detail": "Not found."  
}

## 4. Update a To-Do Item

Endpoint: PUT /api/todos/<id>/

Description: Updates the details of an existing to-do item by ID.

Headers:

Authorization: Bearer <access\_token>

Request Body:

{  
 "title": "Updated Task",  
 "description": "This task has been updated.",  
 "completed": true  
}

Response:

Success (200 OK):

{  
 "id": 1,  
 "title": "Updated Task",  
 "description": "This task has been updated.",  
 "completed": true,  
 "created\_at": "2024-09-01T12:00:00Z",  
 "updated\_at": "2024-09-01T14:00:00Z"  
}

Failure (400 Bad Request):

{  
 "title": ["This field is required."]  
}

## 5. Delete a To-Do Item

Endpoint: DELETE /api/todos/<id>/

Description: Deletes an existing to-do item by ID.

Headers:

Authorization: Bearer <access\_token>

Response:

Success (204 No Content): No content in response; indicates successful deletion.

Failure (404 Not Found):

{  
 "detail": "Not found."  
}