

UST Task Manager

Project: Task Tracker API(Phase 1)

Tech Stack: FastAPI + Python

Security: JWT authentication (HS256)

Data Storage: In-Memory Lists Only (No DB)

Goal: Build a secure backend API that manages tasks.

Step1: login

✓ Scenario 1: Successful Login

Input

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

Output

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1IWIIsInR5cCI6IkpXVCJ9.eyJ3ZdWlI0IjyYWh1bCIsImV4cCI6MTc2NDA0OTg0IiwiaWF0Ij0iMjAyMTEFToWw28tOMhaa42ggoU1B6jFBIZx IHDXu8",
  "token_type": "bearer"
}
```

POST

/login Login

Parameters

No parameters

Request body required

Edit Value | Schema

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


Code

Details

200

Response body

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ3ZdWlI0IjyYWh1bCIsImV4cCI6MTc2NDA0OTg0IiwiaWF0Ij0iMjAyMTEFToWw28tOMhaa42ggoU1B6jFBIZx IHDXu8",
  "token_type": "bearer"
}
```

 Download

✖ Scenario 2: Login Failure

Input

```
{
  "username": "rahul",
  "password": "wrongpass"
}
```

Output

```
{"detail": "Invalid username or password"}
```

POST	/login Login
Parameters	
No parameters	
Request body <small>required</small>	
Edit Value Schema	
<pre>{ "username": "rahul", "password": "wrongpass" }</pre>	
401 <small>Undocumented</small>	Error: Unauthorized
Response body	
<pre>{ "detail": "Incorrect username or password" }</pre>	

✓ Scenario 3: Create Task

Input

```
{
  "title": "Book train tickets",
  "description": "Book IRCTC tickets to Chennai"
}
```

Output

```
{
  "id": 1,
  "title": "Book train tickets",

```

Swagger

```
"description": "Book IRCTC tickets to Chennai",
"completed": false
}
```

POST /tasks Create

Parameters

No parameters

Request body required

Edit Value | Schema

```
{
  "title": "Book train tickets",
  "description": "Book IRCTC tickets to Chennai"
}
```

200

Response body

```
{
  "id": 2,
  "title": "Book train tickets",
  "description": "Book IRCTC tickets to Chennai",
  "completed": false
}
```

✖ Scenario 4: Create Task with Missing Title

Input

```
{
  "description": "Something"
}
```

Output

```
{
  "detail": [
    {
      "loc": ["body", "title"],
      "msg": "field required",
      "type": "value_error.missing"
    }
  ]
}
```

POST `/tasks` Create

Parameters

No parameters

Request body required

[Edit Value](#) | [Schema](#)

```
{
  "description": "Something"
}
```

Media type

application/json ▼

[Example Value](#) | [Schema](#)

```
{
  "detail": [
    {
      "loc": [
        "string",
        0
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

✓ Scenario 5: Update Task

Input

```
{
  "title": "Pay Water Bill",
  "description": "Pay BWSSB bill",
}
```

manager

```
  "completed": true
}
```

Output

```
{
  "id": 1,
  "title": "Pay Water Bill",
  "description": "Pay BWSSB bill",
  "completed": true
}
```

Parameters

Name	Description
------	-------------

task_id * required	
integer	<input type="text" value="2"/>
(path)	

Request body required

[Edit Value](#) | [Schema](#)

```
{
  "title": "Pay Water Bill",
  "description": "Pay BWSSB bill",
  "completed": true
}
```

200

Response body

```
{
  "id": 2,
  "title": "Pay Water Bill",
  "description": "Pay BWSSB bill",
  "completed": true
}
```

Response headers

✖ Scenario 6: Get Non-Existing Task

Response

```
{ "detail": "Task not found" }
```

GET

/tasks/{task_id} Search Byid

Parameters

Name	Description
task_id * required integer (path)	3

Code	Details
<div>404</div> <div>Undocumented</div>	<div>Error: Not Found</div> <div>Response body</div> <div><pre>{ "detail": "Task Not Found"}</pre></div>

✖ Scenario 7: Delete Non-Existing Task

Response

```
{"detail": "Task not found"}
```

DELETE `/tasks/{task_id}` Delete

Parameters

Name	Description
------	-------------

task_id * required integer (path)	<input type="text" value="3"/>
--	--------------------------------

404

Undocumented

Error: Not Found

Response body

```
{  
  "detail": "Task Not Found"  
}
```

Scenario 8: Display all tasks

GET	/tasks	Display
Parameters		
No parameters		
Execute		

200

Response body

```
[
  {
    "id": 2,
    "title": "Pay Water Bill",
    "description": "Pay BWSSB bill",
    "completed": true
  }
]
```

Response headers