

Secure file-sharing system between two different types of users

Submitted By
Dilip Kumar

Contents

1. Introduction	3
2. Requirements.....	4
2.1 Softwares	4
2.2Libraries.....	4
3. APIS	5
3.1 Users API	5
Signup API	5
3.2 File API.....	5
File upload API	5
File Download URL generation API	5
File download.....	6
List all Uploaded file.....	6
4. Screenshots of APIs.....	6
API Authorization	7
USERS API.....	7
SIGNUP API.....	8
User Activation API	9
Login API.....	10
Get user details API.....	11
Profile API.....	12
FILE APIs	13
Uploadfile API	13
Generate file download URL API.....	14
File Download API	15
List Uploaded files API.....	16

1. Introduction

The developed application is built using the fastapi and mysql database. The all api's are secured with jwt token authorization.

It have two type of apis , user APIs an File APIs. Files are uploaded by ops user only an ops user can only upload files of type docx, xlsx, pptx. Other user can see list of files and can download those files.

Only login user can see files and able to download the files.

2. Requirements

2.1 Softwares

- MYSQL
- Python

2.2 Libraries

- FASTAPI
- UVICORN
- SHUTIL
- FASTAPI-EMAIL
- JOSE
- TYPING

MYSQL database is used to store the users details. Python is required to run the FASTAPI project. FASTAPI framework is used for developing the APIs. UVICORN is used to run the project. SHUTIL is used to store the uploaded files. JOSE is used to create the JWT token. TYPING is used for the various things.

To run the project Command is used:

```
Uvicorn app:app --reload
```

--reload is used for tracking the changes and update the same on server running the application.

Production side --reload is not used.

3. APIS

There are two types of API's used

1. Users API

This is responsible for the user's operation. It includes the various user's API's such as

- Signup.
- Login
- Profile
- See User Details
- User activation

2. File API

This is responsible for the file upload and download operations it includes the API's as

- File upload
- File download
- File Download URL generation
- List all Uploaded file

3.1 Users API

This APIs performs all user's operation such as login and signup or listing user or profile data.

All apis are secured only user login can access.

Signup API

it performs the signup using the data of user like name, email and password.

It creates the activation token and sends the email for with activation url.

User activation API

It takes the token from the activation URL and activates the account.

Login API

It takes the username and password and generates the JWT token for the successful login.

Profile API

It sends back the current users' details.

User Details

It takes the user Id and send the data of the given userid.

3.2 File API

File upload API

This api uploads the file to the server and makes entry to the database by whom it is uploaded and filename.

File Download URL generation API

It takes then file name and generates the file download URL.

File download

It takes the file upload Id and sends the file when invoked.

List all Uploaded file

It sends the all-uploaded files list name.

4. Screenshots of APIs

FastAPI 0.1.0 OAS3
/openapi.json

[Authorize](#)

Signup

POST /signup Signup

Activation

GET /activate/{confirmation_code} Activation

login Authentication

POST /login Login

User

GET /{id} Get User Detail

GET /profile/ Get Profile

File

POST /uploadfile/ Create Upload File

GET /generate_url/{filename} Generate Uri

GET /downloadfile/{id} Download File

GET /files/ Get File List

Schemas

Body_create_upload_file_uploadfile__post >

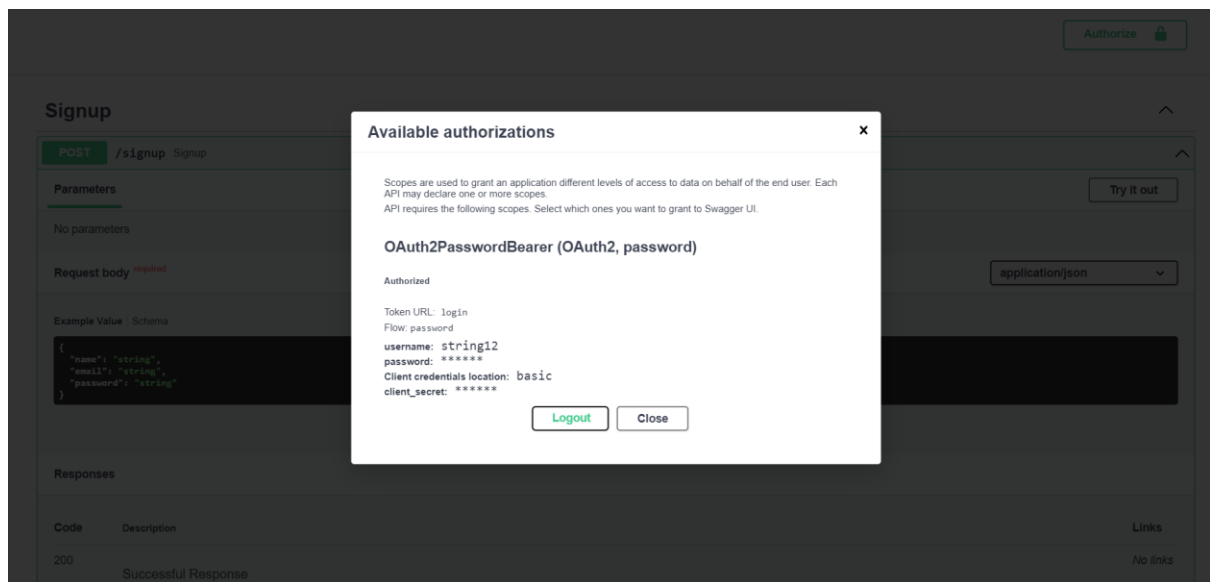
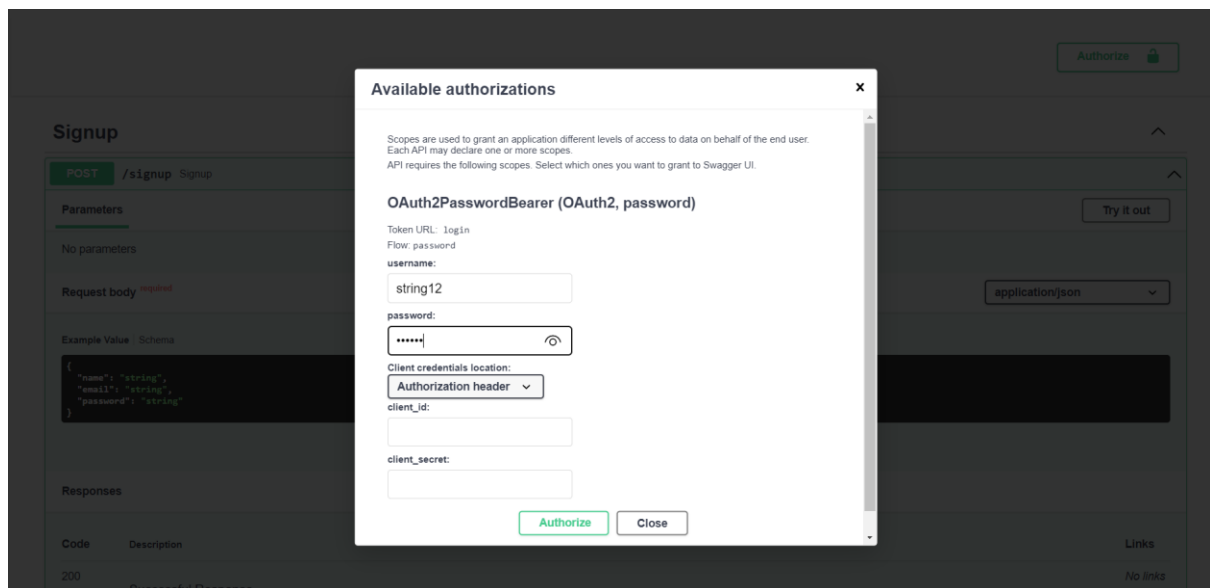
Body_login_login_post >

HTTPValidationError >

User >

ValidationError >

API Authorization



USERS API

Signup			^
POST	/signup	Signup	▼
Activation			^
GET	/activate/{confirmation_code}	Activation	▼
login Authentication			^
POST	/login	Login	▼
User			^
GET	/id	Get User Detail	▼ 🔒
GET	/profile	Get Profile	▼ 🔒

SIGNUP API

Signup

POST

/signup Signup

Parameters

No parameters

Request body required

application/json

```
{
  "name": "abc xyz",
  "email": "abc@abc.com",
  "password": "password"
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8000/signup' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "name": "abc xyz",
    "email": "abc@abc.com",
    "password": "password"
  }'
```

Request URL

http://localhost:8000/signup

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "message": "activation mail sent" }</pre></div><div><div>Response headers</div><div><pre>content-length: 34 content-type: application/json date: Mon, 11 Apr 2022 05:24:50 GMT server: uvicorn</pre></div></div></div>

Code	Description	Links
200	Successful Response	No links
	<div><div>Media type</div><div>application/json</div><div>Controls Accept header</div><div><div>Example Value</div><div>Schema</div></div><div><pre>"string"</pre></div></div>	
422	Validation Error	No links
	<div><div>Media type</div><div>application/json</div><div>Example Value</div><div>Schema</div><div><pre>{ "detail": [{ "loc": ["string"], "msg": "string", "type": "string" }] }</pre></div></div>	

User Activation API

Activation

GET

/activate/{confirmation_code} Activation

Cancel

Parameters

Name	Description
confirmation_code required	4cc420a4-4bdb-4785-b50e-cf3289b9f1c3
string (path)	

ExecuteClear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:8000/activate/4cc420a4-4bdb-4785-b50e-cf3289b9f1c3' \
-H 'accept: application/json'
```

Request URL

```
http://localhost:8000/activate/4cc420a4-4bdb-4785-b50e-cf3289b9f1c3
```

Server response

Code	Details
200	<div><div>Response body<pre>{ "message": "Account Activation Successfull" }</pre></div><div>Response headers<pre>content-length: 44 content-type: application/json date: Mon, 11 Apr 2022 05:27:23 GMT server: uvicorn</pre></div></div>

Responses

Code	Description	Links
200	Successful Response	No links
	<div>Media type<div>application/json</div>Controls Accept header:Example Value Schema<pre>"string"</pre></div>	
422	Validation Error	No links
	<div>Media type<div>application/json</div>Example Value Schema<pre>{ "detail": [{ "loc": ["string"], "msg": "string", "type": "string" }] }</pre></div>	

\wedge

No links

Get user details API

User

GET

/[id] Get User Detail

Parameters

Cancel

Name	Description
id required	
integer	13
(path)	

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/13' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpzZW50L3M4IiwiaWQiOiJ1b3R5b250bmc9PSIsImk1Ijo6LCJ1c2VydmI6IjZlZS1lbnR5cyIsIm44IjE0OTY1MTQ5NW0uPP8-ZGQpImRMTS78jQE1AAR-Cu4cM3M-4ah66w'
```

Request URL

http://localhost:8080/13

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "id": 13, "name": "abc xyz", "email": "abc@abc.com" }</pre></div><div>Response headers</div><div><pre>content-length: 48 content-type: application/json date: Mon, 11 Apr 2022 05:32:22 GMT server: uvicorn</pre></div></div>

Responses

Code	Description	Links
200	Successful Response	No links

Media type

application/json

Controls Accept header

Example Value | Schema

```
"string"
```

422

Validation Error

No links

Media type

application/json

Example Value | Schema

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

Profile API

GET

/profile/

Get Profile

⌵

🔒

Parameters

Cancel

No parameters

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8000/profile/' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IWRXV3R9.ey2zdmU1O1JzdHJpbmcxfl1aIn1k1je0LC31c2VydmUzS16In0ucyIsInV4cCI6IHR5b0TY1NTQ5NW0_PPf8-ZG0JgpImwPT578jQE1AARrCu4cbi3Pr-4ahG6w'
```

Request URL

```
http://localhost:8000/profile/
```

Server response

Code

Details

200

Response body

```
{
  "id": 1,
  "username": "string12",
  "usertype": "ops"
}
```

Response headers

```
content-length: 47
content-type: application/json
date: Mon, 11 Apr 2022 05:33:23 GMT
server: uvicorn
```

Responses

Code

Description

Links

200

Successful Response

Media type

application/json

Controls Accept header.

Example Value | Schema

```
"string"
```

No links

FILE APIs

File

Method	Endpoint	Description	Expand	Lock
POST	/uploadfile/	Create Upload File	⌵	🔒
GET	/generate_url/{filename}	Generate Url	⌵	🔒
GET	/downloadfile/{id}	Download File	⌵	🔒
GET	/files/	Get File List	⌵	🔒

Uploadfile API

File

POST

/uploadfile/

Create Upload File

Parameters

No parameters

Request body required

multipart/form-data

files required

array

Choose File

resume.docx

-

Add string item

Execute

Clear

Responses

Curl

curl -X 'POST' \n'http://localhost:8000/uploadfile/' \n-H 'accept: application/json' \n-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IWRkcCkiLCJ1c2VydmVhZSI6Im9ucyIsInV4cCI6MTY0OTY1NTQ5NW0_PPB-ZGMJgpImm+KTS78jQE1AAR-Cw4cbW3M-4ahG6w' \n-H 'Content-Type: multipart/form-data' \n-F 'files@resume.docx;type=application/vnd.openxmlformats-officedocument.wordprocessingml.document'

Request URL

http://localhost:8000/uploadfile/

Server response

Code

Details

200

Response body

{\n "filename": "resume",\n "type": "docx"\n}

Response headers

content-length: 35\ncontent-type: application/json\ndate: Mon, 11 Apr 2022 05:36:56 GMT\nserver: uvicorn

Responses

Code

Description

Links

200

Successful Response

No links

Media type

application/json

Controls Accept header.

Example Value | Schema

"string"

422

Validation Error

No links

Media type

application/json

Example Value | Schema

{\n "detail": {\n {\n "loc": [\n "string"\n],\n "msg": "string",\n "type": "string"\n }\n }\n}

Generate file download URL API

GET

/generate_url/{filename} Generate Url

Parameters

Cancel

Name	Description
filename * required <small>(path)</small>	<input type="text" value="resume.docx"/>

Execute

Clear

Responses

Curl

```
curl -X 'GET' \  
  'http://localhost:8000/generate_url/resume.docx' \  
  -H 'accept: application/json' \  
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJpdjQpbmcvMiIsInklIjo0LCl1c2VydHlwZSI6Im9wcyIsInV4cCkiOiJMTYB0Y1NTQ5Mm0_PPB-ZlGfOgpImmRTS78jQE14ARrCu4cbk3M+4ahG6w'
```

Request URL

http://localhost:8000/generate_url/resume.docx

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "download_url": "http://localhost:8000/downloadfile/5", "message": "Success" }</pre></div><div><div>Response headers</div><pre>content-length: 75 content-type: application/json date: Mon, 11 Apr 2022 05:38:13 GMT server: uvicorn</pre></div></div>

Responses

Code	Description	Links
200	<div>Successful Response</div> <div><div>Media type</div><div>application/json</div><div>Controls Accept header.</div><div>Example Value Schema</div><div>"string"</div></div>	No links
422	<div>Validation Error</div> <div><div>Media type</div><div>application/json</div><div>Example Value Schema</div><div><pre>{ "detail": [{ "loc": ["string"], "msg": "string", "type": "string" }] }</pre></div></div>	No links

File Download API

/downloadfile/{id} Download File

Parameters

Name	Description
id * required <small>(path)</small>	<input type="text" value="5"/>

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:8000/downloadfile/5' \
-H 'accept: application/json' \
-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJlbnVudHkiLCJmYWduZyJlbWVkaSIsImkiOiJwYXlydXIkbGVhdDNIcnRScGUoLjvcHUULCJleHAJOjE2NDkzMNTYwOTI1LnRqALeTCj7EelbhahNEKPL6ChSrOBsNQyd3huZnPkBA'
```

Request URL

```
http://localhost:8000/downloadfile/5
```

Server response

Code	Details
200	<div>Response bodyDownload file</div> <div>Response headers<pre>content-disposition: attachment; filename="resume.docx" content-length: 3965 content-type: application/octet-stream date: Mon, 11 Apr 2022 05:39:36 GMT etag: "f04dcdf145956c4c2a94c71b6881178d" last-modified: Mon, 11 Apr 2022 05:36:57 GMT server: uvicorn</pre></div>

Responses

Code	Description	Links
200	Successful Response	No links
422	Validation Error	No links

List Uploaded files API

GET

/files/

Get File List

Parameters

No parameters

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:8000/files/' \
-H 'accept: application/json' \
-M 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJkYWVhZjJlLnVkbWVudXI6bnR5dXNlcnc8SzcwLjE3cmMMLCJleHAIOjE2NDk2NTY7OTY19.trgALsTCj7EeHbaHEEMl6CH5rOBsNQyd3hwPbKBN'
```

Request URL

```
http://localhost:8000/files/
```

Server response

Code	Details
200	<div><div>Response body<pre>{ "files": ["assign.docx", "Fill a few decades agl.docx", "Fill a few decades ago.docx", "Cover Letter.docx", "resume.docx"] }</pre></div><div><div>Download</div></div></div> <div><div>Response headers<pre>content-length: 119 content-type: application/json date: Mon, 11 Apr 2022 05:41:01 GMT server: uvicorn</pre></div></div>

Responses

Code	Description	Links
200	Successful Response	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

"string"