Atlan Backend Internship Task -

API calls :-

1). POST /signin

Body (in JSON format) : username , password

Example :

POST method : <baseURL>/signin

Body :

"username": "Utkal",
"password": "utkal123"

Result :

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoiVXRrYWwiLCJhcGIfa2V5IjoiQUNCRUpTREtKR0tEUzEzMzJCSVVKRVNESyIsImlhdCI6MTU4MTg3NzczNCwiZXhwIjoxNzA1MzM3NjEwfQ.Gzm7fSnRGbAJXu6JmP1uwLQKmByDGah8-IjtmJ6qpFo

Description :

Gives auth_token which is required for every other request made.
Look for usersData.json for users credentials. There are 3 users currently.

---

2). POST /upload

Headers: auth_token

Body (in form-data format): file

Example :

POST method : <baseURL>/upload

Headers :

auth_token:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoiVXRrYWwiLCJhcGIfa2V5IjoiQUNCRUpTREtKR0tEUzEzMzJCSVVKRVNESyIsImlhdCI6MTU4MTg3NzczNCwiZXhwIjoxNzA1MzM3NjEwfQ.Gzm7fSnRGbAJXu6JmP1uwLQKmByDGah8-IjtmJ6qpFo

Body :

file : <selected_file>

Result :
Success message when upload finishes. Abort message when upload terminate is called.

Description :
Uploaded file stays in the Temp folder of the system until it completely uploads. After completion, it is copied to 'Uploads' folder in the root directory of the application.

---

3). DELETE /upload/terminate

Headers: auth_token

Example :
DELETE method : <baseURL>/upload/terminate
Headers :
auth_token:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoiVXRrYWwiLCJhcGIfa2V5IjoiQUNCRUpTREtKR0tEUzEzMzJCSVVKRVNESyIsImlhdCI6MTU4MTg3NzczNCwiZXhwIjoxNzA1MzM3NjEwfQ.Gzm7fSnRGbAJXu6JmP1uwLQKmByDGah8-IjtmJ6qpFo

Result :
Success message on successful termination. Otherwise, the reason for failure of termination is returned.

Description :
Terminates the file Upload by the user if he/she is currently uploading. Otherwise sends a message that they are not uploading anything currently.

---

4). POST /export

Headers: auth_token

Body (in JSON format) : "filename"

Example :

POST method : <baseURL>/export

Headers:

auth_token:

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoiVXRrYWwiLCJhcGIfa2V5IjoiQUNCRUpTREtKR0tEUzEzMzJCSVVKRVNESyIsImlhdCI6MTU4MTg3NzczNCwiZXhwIjoxNzA1MzM3NjEwfQ.Gzm7fSnRGbAJXu6JmP1uwLQKmByDGah8-IjtmJ6qpFo

Body :

"filename" : <file_name>

Result :

Success on complete file export. If terminated, a message is sent that the export is terminated.

Description :

Exports the requested file (if it exists in the "Uploads" folder).

---

5). POST /export/terminate

Headers: auth_token

Example :

POST method : <baseURL>/export/terminate

Headers :

auth_token :

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoiVXRrYWwiLCJhcGIfa2V5IjoiQUNCRUpTREtKR0tEUzEzMzJCSVVKRVNESyIsImlhdCI6MTU4MTg3NzczNCwiZXhwIjoxNzA1MzM3NjEwfQ.Gzm7fSnRGbAJXu6JmP1uwLQKmByDGah8-IjtmJ6qpFo

Result :

Success on successful export termination(If the user is exporting). Otherwise, a message is sent stating that the user is not currently exporting any file.

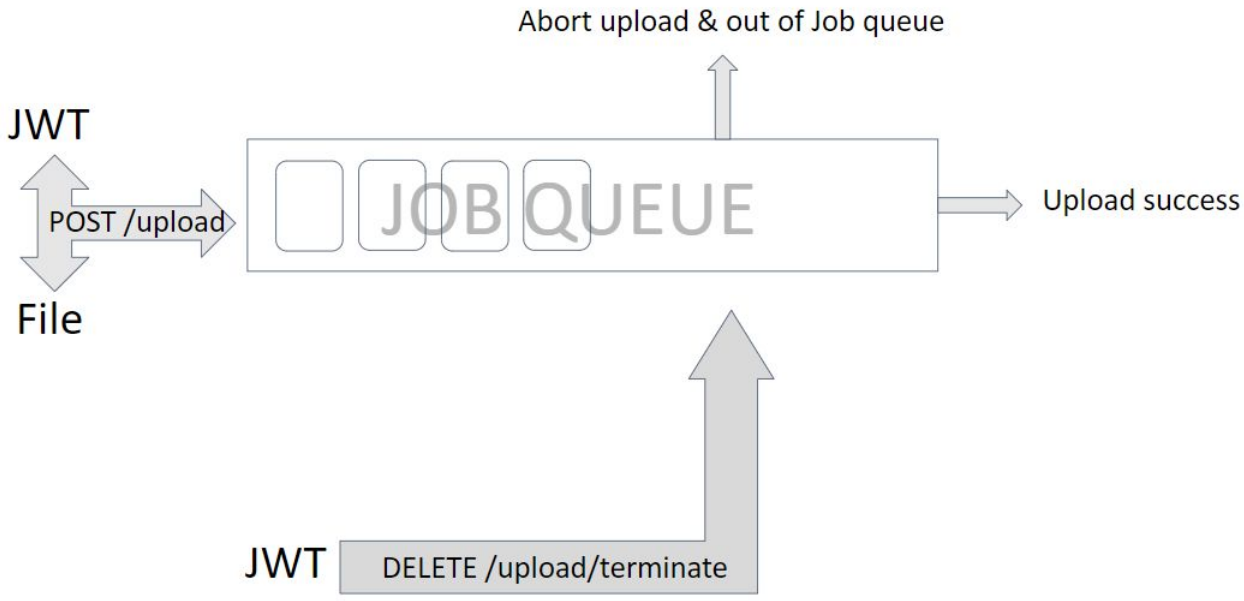
Description :
Terminates the file export that the user requested.

Approach :-

0). **Authentication** :

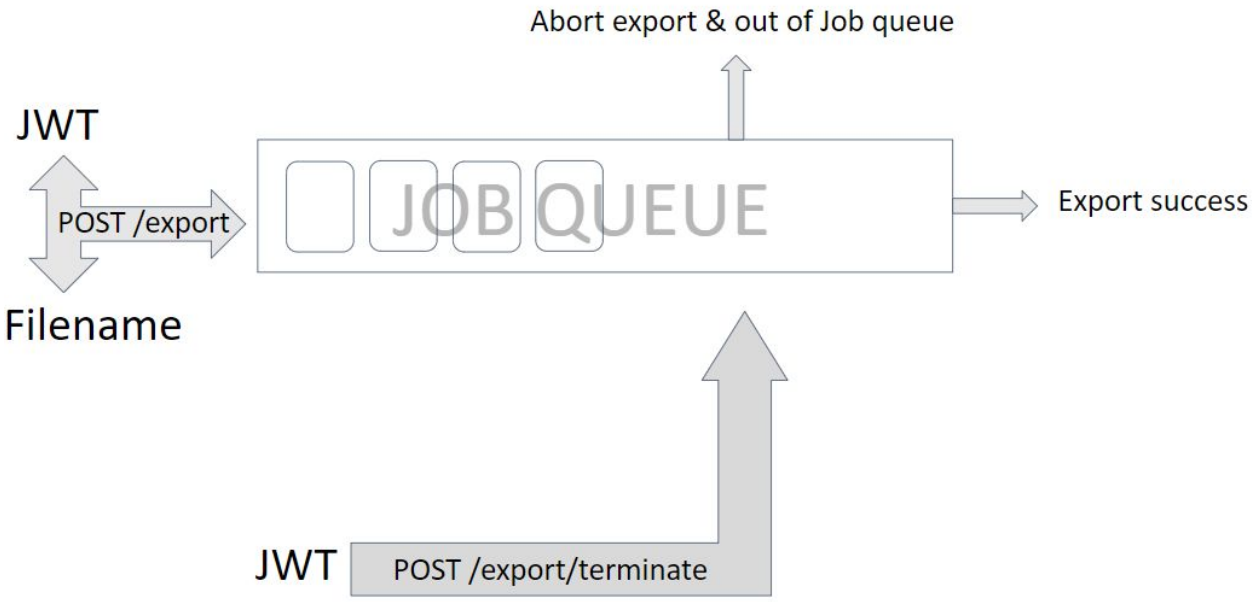
- Generate JWT after verifying the given credentials. (usersData.json contains the list of user credentials.)
- Place the auth_token in headers of every API calls 2,3,4 and 5.
- The user needs to have auth_token for verifying his authorization for any task.

1). **File Uploads** :



- Create a Job Queue that contains all the instances of forms that are currently being parsed.
- Job queue is an object of Key Value pairs, where Key = Username and Value = Instance of the parsing form.
- Whenever an upload event is triggered, we head straight to that instance and handle the event.
- When data is being uploaded, “data” event is triggered and the form is parsed normally. Finally, when whole data is parsed, store the file into ‘Uploads’ directory (initially it is in OS’s Temp directory).
- When there is a request to abort, “abort” event is triggered and the form is stopped from parsing further. Also, this instance is deleted from the Job queue, since it is terminated.

1). File Exports :



- Create a Job Queue that contains all dataStreams (read streams).
- Job queue is an object of Key-Value pairs. Key = Username and Value = dataStream
- Listen to events and handle them.

- When file ‘abort’ is triggered, destroy the stream and delete the dataStream instance in Job queue because it is no longer required.