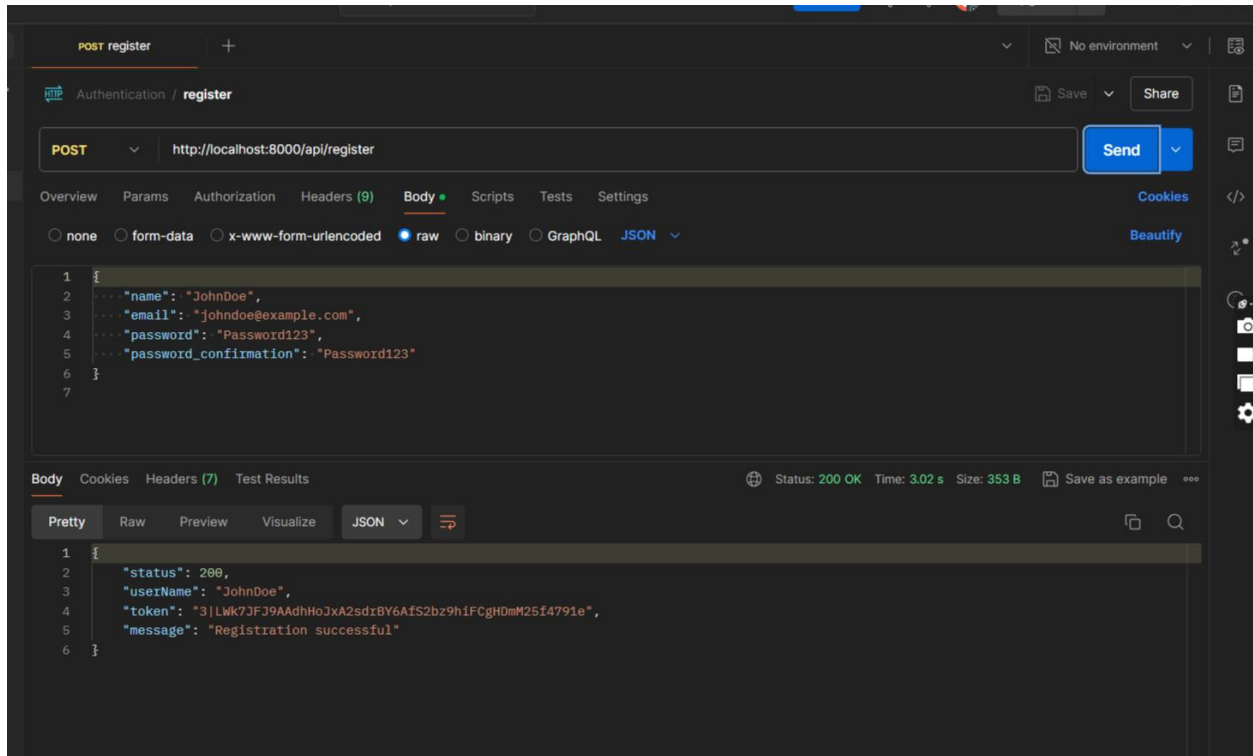


Api Documentation for InvadTasks Project

1- register end-point



2- login end-point

The image shows the Postman application interface. At the top, there's a search bar and navigation icons. Below that, the 'POST login' endpoint is selected, with the URL 'http://localhost:8000/api/login'. The request body is set to 'raw' and contains a JSON object with 'email' and 'password' fields. The response is displayed in the bottom panel, showing a 200 OK status with a JSON body containing 'status', 'userName', 'token', and 'message' fields.

Authentication / login

POST http://localhost:8000/api/login

Overview Params Authorization Headers (10) Body Scripts Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   ... "email": "johndoe@example.com",
3   ... "password": "Password123"
4 }
5
```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 792 ms Size: 352 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": 200,
3   "userName": "JohnDoe",
4   "token": "4|0xbkZ2M0YyFg5jHh1jI0QqS1NWqCITsDN1kbKkUTc59e902f",
5   "message": "Logged in Successfully"
6 }
```

3- retrieve categories;

The screenshot shows the Postman interface for a GET request to `http://localhost:8000/api/categories`. The request is successful with a status of 200 OK, a response time of 689 ms, and a size of 2.37 KB. The response body is displayed in JSON format, showing a list of two categories: "Home" and "Work".

Request Details:

- Method: GET
- URL: `http://localhost:8000/api/categories`
- Headers (9):
 - User-Agent: PostmanRuntime/7.40.0
 - Accept: */*
 - Accept-Encoding: gzip, deflate, br
 - Connection: keep-alive
 - Content-Type: application/json
 - Authorization: Bearer 4|0xbkZ2M0YyFg5jHhJlIQOqSrNWqCITsDN1kbK...

Response Body (JSON):

```
{
  "data": [
    {
      "id": 1,
      "name": "Home",
      "image": "https://st4.depositphotos.com/1009647/27204/i/450/depositphotos_272041202-stock-photo-modern-living-interior-design.jpg",
      "created_at": "2024-07-26T00:00:56.000000Z",
      "updated_at": "2024-07-26T00:00:56.000000Z"
    },
    {
      "id": 2,
      "name": "Work",
      "image": "https://wallpapers.com/images/featured/work-background-kxmiw0h0ugqy2eoa.jpg",
      "created_at": "2024-07-26T00:00:56.000000Z"
    }
  ]
}
```

4- Retrieve Tasks

The screenshot displays the Postman interface for a REST client. The top bar shows the request method as **GET** and the URL as `http://localhost:8000/api/tasks`. The **Send** button is visible. Below the URL bar, the **Headers** tab is selected, showing a table of request headers:

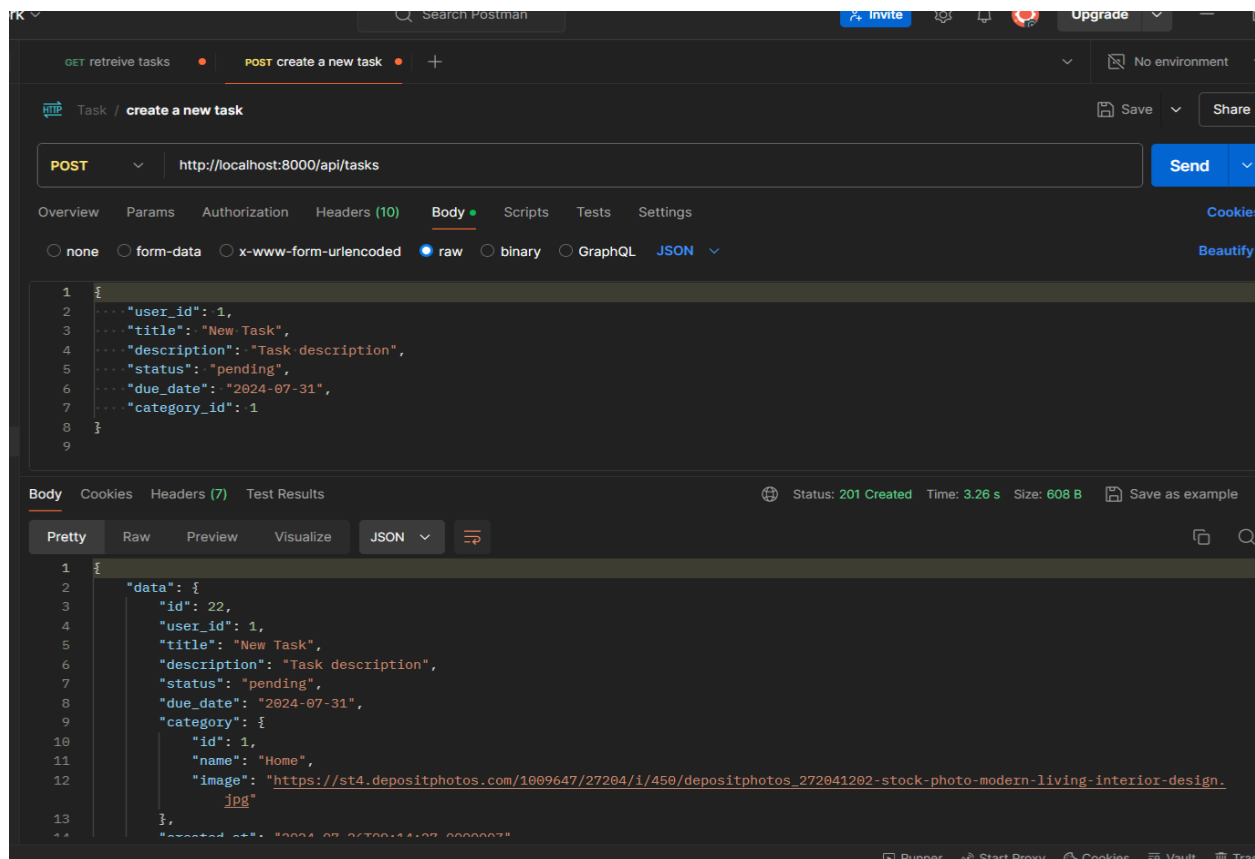
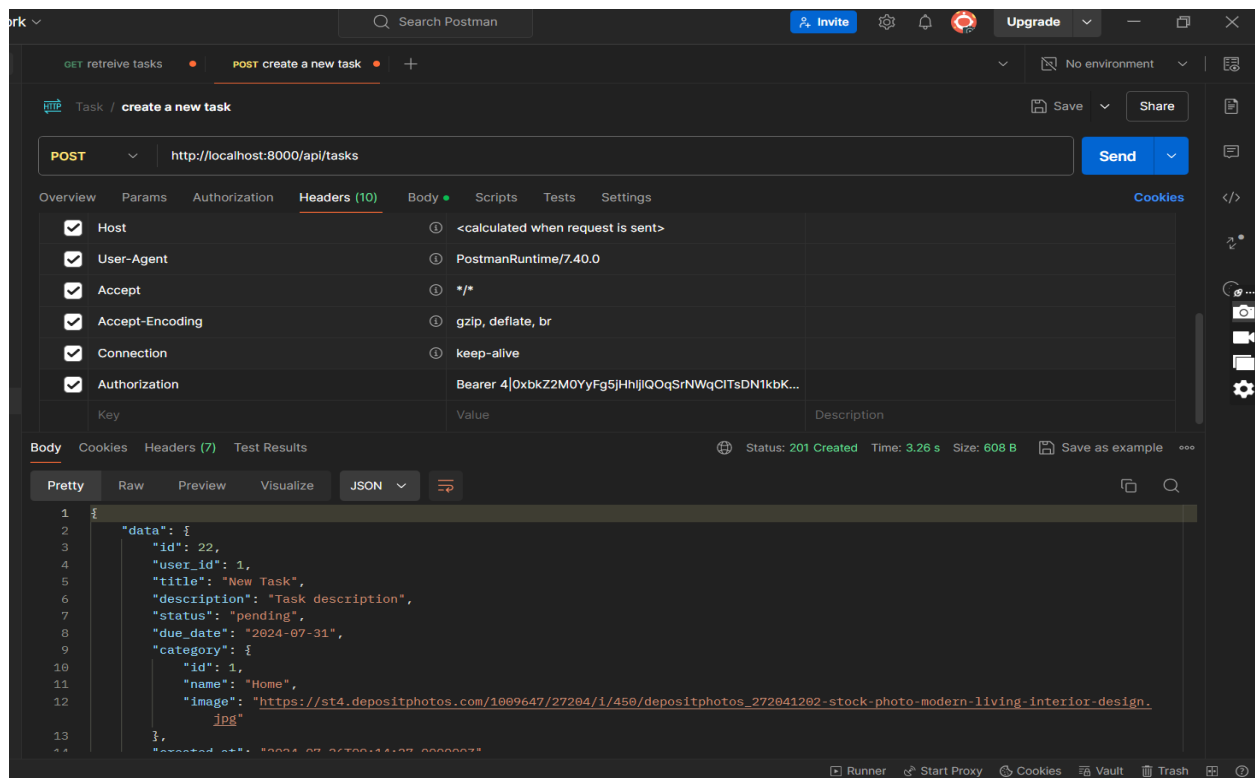
Key	Value	Description
<input checked="" type="checkbox"/> Host	<calculated when request is sent>	
<input checked="" type="checkbox"/> User-Agent	PostmanRuntime/7.40.0	
<input checked="" type="checkbox"/> Accept	/*/*	
<input checked="" type="checkbox"/> Accept-Encoding	gzip, deflate, br	
<input checked="" type="checkbox"/> Connection	keep-alive	
<input checked="" type="checkbox"/> Authorization	Bearer 4 0xbkZ2M0YyFg5jHhijlQOqSrNWqCfTsDN1kbK...	

Below the headers, the **Body** tab is selected, showing the response in **JSON** format. The response status is **200 OK**, with a time of **586 ms** and a size of **3.01 KB**. The JSON data is as follows:

```
1 {
2   "data": [
3     {
4       "id": 1,
5       "user_id": null,
6       "title": "Clean the house",
7       "description": "Do a thorough cleaning of the entire house",
8       "status": "pending",
9       "due_date": "2024-07-30",
10      "category": {
11        "id": 1,
12        "name": "Home",
13        "image": "https://st4.depositphotos.com/1069647/27204/1/450/
depositphotos_272041202-stock-photo-modern-living-interior-design.jpg"
14      }
15    }
16  ]
17 }
```

The bottom status bar includes icons for **Runner**, **Start Proxy**, **Cookies**, **Vault**, **Trash**, and a clock showing **12:12 PM**.

5-Create a new task :



Show Task :

The screenshot displays the Postman interface for a GET request to `http://localhost:8000/api/tasks/4`. The request is saved and has a status of 200 OK, with a response time of 2.55s and a size of 626 B.

Headers (8):

Key	Value	Description
Host	<calculated when request is sent>	
User-Agent	PostmanRuntime/7.40.0	
Accept	*/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	
Authorization	Bearer 4 0xbkZ2M0YyFg5jHhjl QOqSrNWqCITsDN1kbK...	

Body (JSON):

```
1 {
2   "data": {
3     "id": 4,
4     "user_id": 1,
5     "title": "Organize the garage",
6     "description": "Clean and organize the garage",
7     "status": "pending",
8     "due_date": "2024-08-03",
9     "category": {
10      "id": 1,
11      "name": "Home",
12      "image": "https://st4.depositphotos.com/1009647/27204/1/450/depositphotos_272041202-stock-photo-modern-living-interior-design.jpg"
13    }
14  },
15  "message": "Task created successfully"
16 }
```

Edit Task :

The screenshot shows the Postman interface for a task named 'update task'. The task is a PUT request to the URL `http://localhost:8000/api/tasks/4`. The headers tab is selected, showing 10 headers. The body tab is also visible, showing a JSON body.

Task: update task

Method: PUT

URL: `http://localhost:8000/api/tasks/4`

Headers (10):

Key	Value	Description
Host	<calculated when request is sent>	
User-Agent	PostmanRuntime/7.40.0	
Accept	/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	
Authorization	Bearer 4 0xbkZ2M0YyFg5jHhjlIQOqSrNWqCITsDN1kbK...	

Body:

```
1 {
2   "data": {
3     "id": 4,
4     "user_id": 1,
5     "title": "Updated Task4",
6     "description": "Updated description",
7     "status": "completed",
8     "due_date": "2024-08-01",
9     "category": {
10      "id": 1,
11      "name": "Home",
12      "image": "https://st4.depositphotos.com/1009647/27204/i/450/depositphotos_272041202-stock-photo-modern-living-interior-design.jpg"
13    }
14  },
15  "created_at": "2024-07-26T00:00:00.000000Z"
16 }
```

The screenshot shows the Postman interface for the same 'update task' task. The body tab is selected, showing the JSON body. The headers tab is also visible, showing the same 10 headers.

Task: update task

Method: PUT

URL: `http://localhost:8000/api/tasks/4`

Body:

```
1 {
2   "user_id": 1,
3   "title": "Updated Task4",
4   "description": "Updated description",
5   "status": "completed",
6   "due_date": "2024-08-01",
7   "category_id": 1
8 }
9
```

Headers (10):

Key	Value	Description
Host	<calculated when request is sent>	
User-Agent	PostmanRuntime/7.40.0	
Accept	/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	
Authorization	Bearer 4 0xbkZ2M0YyFg5jHhjlIQOqSrNWqCITsDN1kbK...	

Body:

```
1 {
2   "data": {
3     "id": 4,
4     "user_id": 1,
5     "title": "Updated Task4",
6     "description": "Updated description",
7     "status": "completed",
8     "due_date": "2024-08-01",
9     "category": {
10      "id": 1,
11      "name": "Home",
12      "image": "https://st4.depositphotos.com/1009647/27204/i/450/depositphotos_272041202-stock-photo-modern-living-interior-design.jpg"
13    }
14  },
15  "created_at": "2024-07-26T00:00:00.000000Z"
16 }
```

Delete Task

The screenshot displays the Postman interface for a DELETE request. The request is named "Delete Task" and is directed to the URL `http://localhost:8000/api/tasks/4`. The request headers are visible, including Host, User-Agent, Accept, Accept-Encoding, Connection, and Authorization (Bearer token). The response body is shown in JSON format, indicating a successful deletion with the message "Task deleted successfully".

Request Details:

- Method: DELETE
- URL: `http://localhost:8000/api/tasks/4`
- Headers (8):
 - Host: <calculated when request is sent>
 - User-Agent: PostmanRuntime/7.40.0
 - Accept: */*
 - Accept-Encoding: gzip, deflate, br
 - Connection: keep-alive
 - Authorization: Bearer 4[0xbkZ2M0YyFg5]HhijlQOqSrNWqCITsDN1kbK...

Response Details:

- Status: 200 OK
- Time: 567 ms
- Size: 260 B
- Body (JSON):

```
{  "message": "Task deleted successfully"}
```


Restore Task :

The image shows the Postman REST client interface. At the top, there's a search bar and navigation icons. Below that, a list of tasks is shown: GET show task, PUT update task, DEL Delete Task, and POST Restore Task (selected). The main area displays the details for the 'Restore Task' endpoint: `POST http://localhost:8000/api/tasks/4/restore`. The 'Headers' tab is active, showing a list of headers with checkboxes for each: Host, User-Agent, Accept, Accept-Encoding, Connection, and Authorization. The Authorization header is set to 'Bearer 4[0xbkZ2M0YyFg5]HhjlQOqSrNWqCITsDN1kbK...'. Below the headers, the 'Body' tab is active, showing a JSON response in 'Pretty' format. The response status is 200 OK, with a time of 539 ms and a size of 612 B. The JSON body contains a 'data' object with fields for id, user_id, title, description, status, due_date, and category.

Task: Restore Task

POST `http://localhost:8000/api/tasks/4/restore` Send

Overview Params Authorization Headers (9) Body Scripts Tests Settings Cookies

Key	Value	Description
<input checked="" type="checkbox"/> Host	<calculated when request is sent>	
<input checked="" type="checkbox"/> User-Agent	PostmanRuntime/7.40.0	
<input checked="" type="checkbox"/> Accept	*/*	
<input checked="" type="checkbox"/> Accept-Encoding	gzip, deflate, br	
<input checked="" type="checkbox"/> Connection	keep-alive	
<input checked="" type="checkbox"/> Authorization	Bearer 4[0xbkZ2M0YyFg5]HhjlQOqSrNWqCITsDN1kbK...	

Body Cookies Headers (7) Test Results Status: 200 OK Time: 539 ms Size: 612 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "data": {
3     "id": 4,
4     "user_id": 1,
5     "title": "Updated Task4",
6     "description": "Updated description",
7     "status": "completed",
8     "due_date": "2024-08-01",
9     "category": {
10      "id": 1,
11      "name": "Home",
12      "image": "https://st4.depositphotos.com/1669647/27204/1/456/depositphotos_272041202-stock-photo-modern-living-interior-design.jpg"
13    },
14   },
15   "updated_at": "2024-08-01T00:00:00.000000Z"
```

Runner Start Proxy Cookies Vault Trash

Logout :

The image shows the Postman application interface for a POST request to the `http://localhost:8000/api/logout` endpoint. The request is successful, returning a 200 OK status with a response body indicating successful logout.

Request Details:

- Method: POST
- URL: `http://localhost:8000/api/logout`
- Headers (11):
 - Accept: `*/*`
 - Accept-Encoding: `gzip, deflate, br`
 - Connection: `keep-alive`
 - Access-Token: `KWprgWbPz4mYPzavtelS1e1D21iKwIYvr7Aorqrh7qbCjl...`
 - Content-Type: `application/json`
 - Authorization: `Bearer 4|0xbkZ2M0YyFg5jHhijlQQoSrNWqCITsDN1kbK...`

Response Details:

- Status: 200 OK
- Time: 664 ms
- Size: 271 B
- Body (JSON):

```
{
  "status": 200,
  "message": "Logged out successfully"
}
```