Online Property Rental

User

User related endpoints

GET Home page

http://localhost:8080/property-rental/

Property Rental API

This API endpoint is used to retrieve the property rental web page.

Request

HTTP Request

`GET http://localhost:8080/property-rental/`

Response

- Status: 200
- Body:

```
html

<html>
          <body>
          <h2 class=&#x27;preserveHtml&#x27; class=&#x27;preserveHtml&#x27; class=&#x27;pr
          </body>
          </html>
```

POST Sign Up

http://localhost:8080/property-rental/user?action=signup

Property Rental User Signup

This endpoint is used to sign up a new user for the property rental system.

Request Body Parameters

- `fullname` (text): The full name of the user.
- `email` (text): The email address of the user.
- `hashedPassword` (text): The hashed password of the user.
- `DOB` (text): The date of birth of the user.
- `UserType` (text): The type of user.

Response

Upon successful user creation, the response will have a status code of 200 and include the details of the newly created user, including the `userId`, `fullname`, `email`, `DOB`, and `userType`.

PARAMS

action	signup

Body urlencoded

action	signup
fullname	Johnpaul
email	johnpaul@gmail.com
hashedPassword	emmanuel123
DOB	2002-06-18
UserType	admin

GET Get all users

http://localhost:8080/property-rental/user/secured/?action=getAllUsers

This endpoint makes an HTTP GET request to retrieve all users from the secured property rental system. The request should be sent to http://localhost:8080/property-rental/user/secured/?action=getAllUsers with an empty form-data request body.

Response

The response will have a status code of 200 and will contain an array of user objects. Each user object will have the following properties:

- userId (number): The unique identifier for the user.
- fullname (string): The full name of the user.
- email (string): The email address of the user.
- userType (string): The type of user.
- hashedPassword (string): The hashed password of the user.
- dob (number): The date of birth of the user in Unix timestamp format.

PARAMS

action	getAllUsers

Body formdata

email	john@gmail.com

GET Get User By Email

http://localhost:8080/property-rental/user/secured/?action=getUserByEmail&email=update@gmail.com

This endpoint makes an HTTP GET request to retrieve a user's information by their email. The request should include the 'action' parameter set to 'getUserByEmail' and the 'email' parameter set to the user's email address.

Request Parameters

- action: set to 'getUserByEmail'
- email: the user's email address

Response

Upon a successful request, the server returns a status code of 200 along with the user's information including userId, fullname, email, date of birth (DOB), and userType.

Example:

```json
{
```

```
    "userId": 1,
    "fullname": "update",
    "email": "update@gmail.com",
    "DOB": "1983-06-12",
    "userType": "customer"
}
```


PARAMS

action	getUserByEmail
email	update@gmail.com

Body urlencoded

email	anene@gmail.com

GET Get User By UserId

http://localhost:8080/property-rental/user/secured/?action=getByUserId&userId=1

This API endpoint makes an HTTP GET request to retrieve a user's information by their user ID. The request should include the action parameter set to 'getByUserId' and the userId parameter set to the specific user ID. The response will have a status code of 200 if the user is found, and it will include the user's details such as userId, fullname, email, date of birth, and userType.

PARAMS

action	getByUserId
userId	1

DELETE Delete User

http://localhost:8080/property-rental//user/secured/?action=delete&userId=11

This endpoint sends an HTTP DELETE request to http://localhost:8080/property-rental//user/secured/?action=delete&userId=11 to delete a user. The request does not have a request body.

Response

The request returns a status code of 200, indicating a successful deletion with the message "User has been deleted".

PARAMS

action	delete
userId	11

PUT Update User

http://localhost:8080/property-rental//user/secured/?action=update&userId=10&DOB=1983-06-12&userType=admin&fullname=update user two&email=user2u6pdate@gmail.com&hashedPassword=update1234

This endpoint allows updating user information with the specified user ID. The HTTP PUT request should be made to http://localhost:8080/property-rental//user/secured/?action=update&userId=10&DOB=1983-06-12&userType=admin&fullname=update user two&email=user2u6pdate@gmail.com&hashedPassword=update1234. The request does not require a request body.

The response to the request will have a status code of 200, indicating a successful update. The response message will confirm the details of the updated user, including the user ID, full name, email, date of birth, and user type.

PARAMS

action	update
userId	10
DOB	1983-06-12
userType	admin
fullname	update user two
email	user2u6pdate@gmail.com
hashedPassword	update1234

POST Login

http://localhost:8080/property-rental/login

Property Rental Login

This endpoint is used to log in to the property rental system.

Request Body

- email (text): The email of the user.
- password (text): The password of the user.

Response

Upon successful login, the response will have a status code of 200 and the message "Login Successful".

PARAMS

action

Body urlencoded

email	update@gmail.com
password	update1234

GET LogOut

http://localhost:8080/property-rental//logout

This endpoint is used to log out the user from the system.

When a GET request is made to {{root}}/logout, it logs out the user and returns a status code of 200 with the message "GoodBye".

House

House related endpoints

DELETE Delete house

http://localhost:8080/property-rental//house/secured/?action=delete&houseId=7

This endpoint is used to delete a secured house by sending an HTTP DELETE request to the specified URL. The request

This endpoint is used to delete a secured house by sending an HTTP DELETE request to the specified URL. The request should include the 'action' parameter set to 'delete' and the 'houseId' parameter set to the ID of the house to be deleted.

The request body for this endpoint has an undefined type and does not require any payload.

Response

Upon successful execution, the endpoint returns a status code of 200 with the message "House has been deleted".

PARAMS

action	delete
houseId	7

POST Create House

http://localhost:8080/property-rental//house/secured/?action=create

This endpoint allows you to create a secured house. When making a POST request to `{{root}}/house/secured/?action=create`, you should include the following parameters in the x-www-form-urlencoded request body:

- `userId`: (text) The ID of the user.
- `rent`: (text) The rent amount for the house.
- `numberOfRooms`: (text) The number of rooms in the house.
- `locality`: (text) The locality of the house.

Response

Upon successful creation, the endpoint will return a status code of 200 along with the details of the newly created house, including the `houseId`, `userId`, `rent`, `locality`, and `numberOfRooms`.

PARAMS

action	create
userId	1

Body urlencoded

userId	10
rent	700

numberOfRooms	2
locality	minna

PUT Update House

http://localhost:8080/property-rental//house/secured/?action=update&houseId=7&rent=1000&numberOfRooms=1&locality=minna

This HTTP PUT request is used to update the details of a secured house. The request should be made to {{root}}/house/secured/ with the action parameter set to "update" and the houseId, rent, numberOfRooms, and locality provided as query parameters.

The request does not require a request body.

The response to the request will have a status code of 200, indicating a successful update. The response body will confirm that the house has been updated, providing the updated details including houseId, userId, rent, locality, and numberOfRooms.

PARAMS

action	update
houseId	7
rent	1000
numberOfRooms	1
locality	minna

GET Get All house

http://localhost:8080/property-rental//house/?action=getAllHouse

This endpoint makes an HTTP GET request to retrieve all houses. The request does not require a request body.

The response to the request returns a status code of 200, along with an array of objects. Each object represents a house and includes the following properties:

- houseId (number): The unique identifier for the house.
- userId (number): The user ID associated with the house.
- rent (number): The rent amount for the house.

- locality (string): The locality where the house is located.
- numberOfRooms (number): The number of rooms in the house.

PARAMS

action	getAllHouse

GET Get house by houseId

http://localhost:8080/property-rental//house/?action=getHouseByHouseId&houseId=2

This endpoint makes an HTTP GET request to retrieve the details of a specific house by its houseId. The request should include the query parameter "houseId" with the ID of the house to be retrieved.

The response will have a status code of 200, and it will contain the details of the house with the specified houseId, including the userId, rent, locality, and the number of rooms.

For example, a successful response will look like:

Plain Text

```
This is the house with houseId: house{houseId=2, userId=1, rent=500, locality='abuja', numberO
```

PARAMS

action	getHouseByHouseId
houseId	2

GET Get House by userId

http://localhost:8080/property-rental//house/?action=getHouseByUserId&userId=1

This endpoint makes an HTTP GET request to retrieve the houses owned by a specific user based on the provided userId. The request should include the action parameter set to 'getHouseByUserId' and the userId parameter specifying the user's ID.

The response to the last call returned a 200 status, providing a list of houses owned by the user. The response includes details such as houseId, userId, rent, locality, and numberOfRooms for each house.

Example Response:

json


```
[
    {"houseId":2,"userId":1,"rent":500,"locality":"abuja","numberOfRooms":4},
    {"houseId":3,"userId":1,"rent":500,"locality":"abuja","numberOfRooms":4},
    {"houseId":4,"userId":1,"rent":500,"locality":"abuja","numberOfRooms":4},
    {"houseId":5,"userId":1,"rent":500,"locality":"abuja","numberOfRooms":4}
]
```


PARAMS

action	getHouseByUserId
userId	1

GET Get house by rent

http://localhost:8080/property-rental//house/?action=getHouseByRent&rent=500

This endpoint makes an HTTP GET request to retrieve houses based on the specified rent. It takes in a query parameter "action" with the value "getHouseByRent" and "rent" which represents the maximum rent value for the houses to be retrieved.

The last call to this request did not include a request body.

Response

The endpoint returns a status code of 200 along with an array of houses that match the specified rent. Each house object in the response includes the following properties:

- "houseId": The unique identifier for the house
- "userId": The user identifier associated with the house
- "rent": The rent amount for the house
- "locality": The locality where the house is located
- "numberOfRooms": The number of rooms in the house

PARAMS

action	getHouseByRent
rent	500

GET Get house by rooms

http://localhost:8080/property-rental//house/?action=getHouseByNumberOfRoom&numberOfRooms=2

This endpoint makes an HTTP GET request to retrieve houses based on the number of rooms specified. It uses the query parameter "numberOfRooms" to filter the results.

Request

- Method: GET
- URL: {{root}}/house/?action=getHouseByNumberOfRoom&numberOfRooms=2

Response

The response will have a status code of 200, indicating a successful request. It will contain an array of objects, each representing a house with the following properties:

- "houseId": The unique identifier for the house
- "userId": The user identifier associated with the house
- "rent": The rent amount for the house
- "locality": The locality where the house is located
- "numberOfRooms": The number of rooms in the house

Example response:

Plain Text


```
[
    {
        "houseId": 8,
        "userId": 10,
        "rent": 700,
        "locality": "minna",
        "numberOfRooms": 2
    }
]
```


PARAMS

action	getHouseByNumberOfRoom
numberOfRooms	2

GET Get house by locality

http://localhost:8080/property-rental//house/?action=getHouseByLocality&locality=abuja

This endpoint makes an HTTP GET request to retrieve houses based on the specified locality. It takes a query parameter "locality" to filter the houses based on the provided locality. The response will include an array of objects, each representing a house with details such as houseId, userId, rent, locality, and numberOfRooms.

For example, a request to get houses in the locality "abuja" would return a JSON array with details of the houses in that locality.

The last call to this endpoint with the specified parameters returned a 200 status code along with an array of houses in the specified locality.

PARAMS

action	getHouseByLocality
locality	abuja