Testing Documentation for RentalNinja

1. get-presigned-url

- Method: POST

- Endpoint: /get-presigned-url

- Test Objective: Verify that the API returns a valid presigned URL for uploading a file to AWS S3.

- Test Data: No request body required.

- Expected Result: A presigned URL is successfully returned.

- Screenshot:

A screenshot of a computer

Description automatically generated

2. get-specific-post

- Method: POST

- Endpoint: /get-post-detail

- Test Objective: Verify that the API correctly retrieves the details of a specific post.

- Test Data:

{

"postId": "873c1deb-6dd3-4be2-8c9c-2850eb5caf87"

}

- Expected Result: The correct post details are returned, including title, content, and metadata.

A screenshot of a computer

Description automatically generated

3. create-post

- Method: POST

- Endpoint: /upload-post

- Test Objective: Verify that the API successfully creates a new post with all required fields.

- Test Data:

{

"userId": "userId\_3002d4ff47b3",

"title": "testttttt",

"content": "13213123123213",

"contactInfo": "contactInfo\_68bd734c79a6",

"picUrls": "picUrls\_958272bf91fc",

"countryCode": "country\_fc3d13f854c1",

"stateCode": "state\_bac4a8e3fad4",

"cityCode": "city\_895a00e96818",

"price": 100,

"area": "area\_31b5a51ea565"

}

- Expected Result: The post is successfully created and a success message is returned.

- Screenshot: A screenshot of a computer

Description automatically generated

4. put-object

- Method: PUT

- Endpoint: AWS S3 Presigned URL

- Test Objective: Verify that the file is successfully uploaded to S3 using the provided presigned URL.

- Test Data: File upload (e.g., image or document).

- Expected Result: The file is successfully uploaded.

- Screenshot:

A screenshot of a computer

Description automatically generated

5. delete-post

- Method: POST

- Endpoint: /delete-post

- Test Objective: Verify that the API successfully deletes the specified post.

- Test Data:

{

"postId": "7158df56-26e8-4b21-8b45-82325ab7c914"

}

- Expected Result: The post is successfully deleted and a success message is returned.

- Screenshot:

A screenshot of a computer

Description automatically generated

6. get-post-list

- Method: POST

- Endpoint: /get-post-list

- Test Objective: Verify that the API retrieves a list of posts based on the provided search parameters.

- Test Data:

{

"stateCode": "",

"cityCode": "",

"keyword": "",

"pageNum": 0,

"pageSize": 2

}

- Expected Result: A list of posts is successfully retrieved.

- Screenshot:

A screenshot of a computer

Description automatically generated

7. add-collection

- Method: POST

- Endpoint: /add-collection

- Test Objective: Verify that the API correctly adds or removes a post from the user's collection.

- Test Data:

{

"postId": "226de9e3-85c6-4ab6-8e6c-152c7e3f2eff",

"isAdd": 1

}

- isAdd: 1 to add to collection, 0 to remove.

- Expected Result: The post is successfully added or removed from the collection.

- Screenshot:

A screenshot of a computer

Description automatically generated

8. get-collection-list

- Method: POST

- Endpoint: /get-collection-list

- Test Objective: Verify that the API retrieves the list of collected posts for the user.

- Test Data: No request body required.

- Expected Result: The collected posts are successfully retrieved.

- Screenshot:

A screenshot of a computer

Description automatically generated

9. get-my-list

- Method: POST

- Endpoint: /get-my-list

- Test Objective: Verify that the API retrieves the list of posts created by the user.

- Test Data:

{

"pageNum": 0,

"pageSize": 10

}

- Expected Result: A list of user-created posts is successfully retrieved.

- Screenshot:

A screenshot of a computer

Description automatically generated

10. update-post

- Method: POST

- Endpoint: /update-my-post

- Test Objective: Verify that the API successfully updates an existing post with new information.

- Test Data:

{

"postId": "873c1deb-6dd3-4be2-8c9c-2850eb5caf87",

"title": "testttttt",

"content": "13213123123213",

"contactInfo": "contactInfo\_68bd734c79a6",

"picUrls": "picUrls\_958272bf91fc",

"countryCode": "country\_fc3d13f854c1",

"stateCode": "state\_bac4a8e3fad4",

"cityCode": "city\_895a00e96818",

"price": 100,

"area": "area\_31b5a51ea565"

}

- Expected Result: The post is successfully updated.

- Actual Result:

{

"message": "Internal server error"

}

- Screenshot:A screenshot of a computer

Description automatically generated

**Exception Cases**

**1.Token Expired**

**Scenario**: The user's token has expired, and the API request is made with an invalid token.

**Expected Response**:

{

"message": "The incoming token has expired"

}

Affected Endpoints: All endpoints that require authentication.

Screenshot:

A screenshot of a computer

Description automatically generated

2.Server Error

Scenario: The server encounters an internal error while processing the request.

Expected Response:A screenshot of a computer

Description automatically generated