

Updated 10/6/2019

Note: Screenshots do not necessarily represent consecutive order of events!

POST:

<https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/assoctagsanduserinfo>

Description: Use this to associate a list of tags for a particular image of a user. This is basically adding an image, its tags for a user, and adds the user itself as the true owner of the image. It also updates the tags for an image, so whatever list of tags you pass for an image, all those tags will become the new tags for that image.

Screenshot of when user isn't yet in database (brand new user/owner):

The screenshot shows a Postman interface with a POST request to `https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/assoctagsanduserinfo`. The request body is a JSON object:

```
{
  "username": "sampleUser1@gmail.com",
  "imagename": "imageToBeStoredTestEvent1.png",
  "tags": [
    "tree",
    "fractal",
    "nature",
    "colorful",
    "postman event",
    "hidden postman event success"
  ]
}
```

The response status is 200 OK. The response body is a JSON object:

```
{
  "statusCode": 200,
  "headers": {
    "Access-Control-Allow-Origin": "*"
  },
  "body": "{\\\"userName\\\": \\\"sampleUser1@gmail.com\\\", \\\"categories\\\": \\\"[]\\\", \\\"imageName\\\": \\\"[\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\"]\\\", \\\"refToImage\\\": \\\"[\\\"https://s3.amazonaws.com/imageuploader-main-bucket/All_User_Images/sampleUser1%40gmail.com*imageToBeStoredTestEvent1.png\\\"]\\\", \\\"imgDictByTag\\\": \\\"[\\\"tree\\\": [\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\"], \\\"nature\\\": [\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\"], \\\"fractal\\\": [\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\"], \\\"colorful\\\": [\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\"], \\\"postman event\\\": [\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\"], \\\"hidden postman event success\\\": [\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\"]\\\", \\\"canView\\\": \\\"[\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\": [\\\"sampleUser1@gmail.com\\\"]\\\", \\\"imgDictByImage\\\": \\\"[\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\": [\\\"tree\\\", \\\"fractal\\\", \\\"nature\\\", \\\"colorful\\\", \\\"postman"
}
```

Screenshot of when you want to add a new image and tags:

POST https://c2ecjqc X POST https://c2ecjqoud4 POST https://c2ecjqoud4 GET https://c2ecjqoud4 GET https://c2ecjqoud4 [CONFLICT] GET https://c No Environment

POST https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/assotagsanduserinfo

Params Authorization Headers (1) Body Pre-request Script Tests Cookies Code Comments (0)

none form-data x-www-form-urlencoded raw binary JSON (application/json) Beautify

```
1 {
2
3   "username": "sampleUser1@gmail.com",
4   "imagename": "imageToBeStoredTestEvent2.png",
5   "tags": [
6     "tree",
7     "fractal",
8     "nature",
9     "colorful",
10    "postman event",
11    "2nd postman event"
12  ]
13 }
14 }
```

Body Cookies Headers (8) Test Results Status: 200 OK Time: 391 ms Size: 2.23 KB Save Download

Pretty Raw Preview JSON

```
1 {
2   "statusCode": 200,
3   "headers": {
4     "Access-Control-Allow-Origin": "*"
5   },
6   "body": "{\\\"username\\\": \\\"sampleUser1@gmail.com\\\", \\\"categories\\\": \\\"[]\\\", \\\"imageName\\\": \\\"[\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent2.png\\\", \\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\"]\\\", \\\"refToImage\\\": \\\"[\\\"https://s3.amazonaws.com/imageuploader-main-bucket/All_User_Images/sampleUser1%40gmail.com*imageToBeStoredTestEvent1.png\\\", \\\"https://s3.amazonaws.com/imageuploader-main-bucket/All_User_Images/sampleUser1%40gmail.com*imageToBeStoredTestEvent2.png\\\"]\\\", \\\"imgDictByTag\\\": \\\"[\\\"tree\\\": [\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\", \\\"sampleUser1@gmail.com*imageToBeStoredTestEvent2.png\\\"]\\\", \\\"nature\\\": [\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\", \\\"sampleUser1@gmail.com*imageToBeStoredTestEvent2.png\\\"]\\\", \\\"fractal\\\": [\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\", \\\"sampleUser1@gmail.com*imageToBeStoredTestEvent2.png\\\"]\\\", \\\"colorful\\\": [\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\", \\\"sampleUser1@gmail.com*imageToBeStoredTestEvent2.png\\\"]\\\", \\\"postman event\\\": [\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\", \\\"sampleUser1@gmail.com*imageToBeStoredTestEvent2.png\\\"]\\\", \\\"2nd postman event\\\": [\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\", \\\"sampleUser1@gmail.com*imageToBeStoredTestEvent2.png\\\"]\\\"}\\\"}"
```

Screenshot of updating tags for a current user's image (includes adding and deleting tags):

POST https://c2ecjqc X POST https://c2ecjqoud4 POST https://c2ecjqoud4 GET https://c2ecjqoud4 GET https://c2ecjqoud4 [CONFLICT] GET https://c No Environment

POST https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/assotagsanduserinfo

Params Authorization Headers (1) Body Pre-request Script Tests Cookies Code Comments (0)

none form-data x-www-form-urlencoded raw binary JSON (application/json) Beautify

```
1 {
2
3   "username": "sampleUser1@gmail.com",
4   "imagename": "imageToBeStoredTestEvent1.png",
5   "tags": [
6     "tree",
7     "fractal",
8     "canopy",
9     "colorful",
10    "postman event",
11    "event postman"
12  ]
13 }
14 }
```

Body Cookies Headers (8) Test Results Status: 200 OK Time: 278 ms Size: 2.35 KB Save Download

Pretty Raw Preview JSON

```
1 {
2   "statusCode": 200,
3   "headers": {
4     "Access-Control-Allow-Origin": "*"
5   },
6   "body": "{\\\"username\\\": \\\"sampleUser1@gmail.com\\\", \\\"categories\\\": \\\"[]\\\", \\\"imageName\\\": \\\"[\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent2.png\\\", \\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\"]\\\", \\\"refToImage\\\": \\\"[\\\"https://s3.amazonaws.com/imageuploader-main-bucket/All_User_Images/sampleUser1%40gmail.com*imageToBeStoredTestEvent1.png\\\", \\\"https://s3.amazonaws.com/imageuploader-main-bucket/All_User_Images/sampleUser1%40gmail.com*imageToBeStoredTestEvent2.png\\\"]\\\", \\\"imgDictByTag\\\": \\\"[\\\"tree\\\": [\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\", \\\"sampleUser1@gmail.com*imageToBeStoredTestEvent2.png\\\"]\\\", \\\"canopy\\\": [\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\", \\\"sampleUser1@gmail.com*imageToBeStoredTestEvent2.png\\\"]\\\", \\\"fractal\\\": [\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\", \\\"sampleUser1@gmail.com*imageToBeStoredTestEvent2.png\\\"]\\\", \\\"colorful\\\": [\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\", \\\"sampleUser1@gmail.com*imageToBeStoredTestEvent2.png\\\"]\\\", \\\"event postman\\\": [\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\", \\\"sampleUser1@gmail.com*imageToBeStoredTestEvent2.png\\\"]\\\", \\\"2nd postman event\\\": [\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\", \\\"sampleUser1@gmail.com*imageToBeStoredTestEvent2.png\\\"]\\\", \\\"canView\\\": \\\"[\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\", \\\"sampleUser1@gmail.com*imageToBeStoredTestEvent2.png\\\"]\\\", \\\"canView\\\": \\\"[\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\", \\\"sampleUser1@gmail.com*imageToBeStoredTestEvent2.png\\\"]\\\", \\\"imgDictByImage\\\": \\\"[\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\", \\\"sampleUser1@gmail.com*imageToBeStoredTestEvent2.png\\\"]\\\"}\\\"}"
```

Screenshot of Headers (for all api's so far):

Body	Cookies	Headers (8)	Test Results
Date → Thu, 21 Mar 2019 05:39:31 GMT			
Content-Type → application/json			
Content-Length → 2081			
Connection → keep-alive			
x-amzn-RequestId → b356b928-4b9b-11e9-8c7d-fbd4794c3f4e			
Access-Control-Allow-Origin → *			
x-amz-apigw-id → W4J_AEtjlAMFxqg=			
X-Amzn-Trace-Id → Root=1-5c932393-77311883bf8cf7f9e70b6f9a;Sampled=0			

POST:

<https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/actuallydeleteimage>

Description: Deletes an image related to target user/owner.

Screenshot of info needed to delete an image of a user:

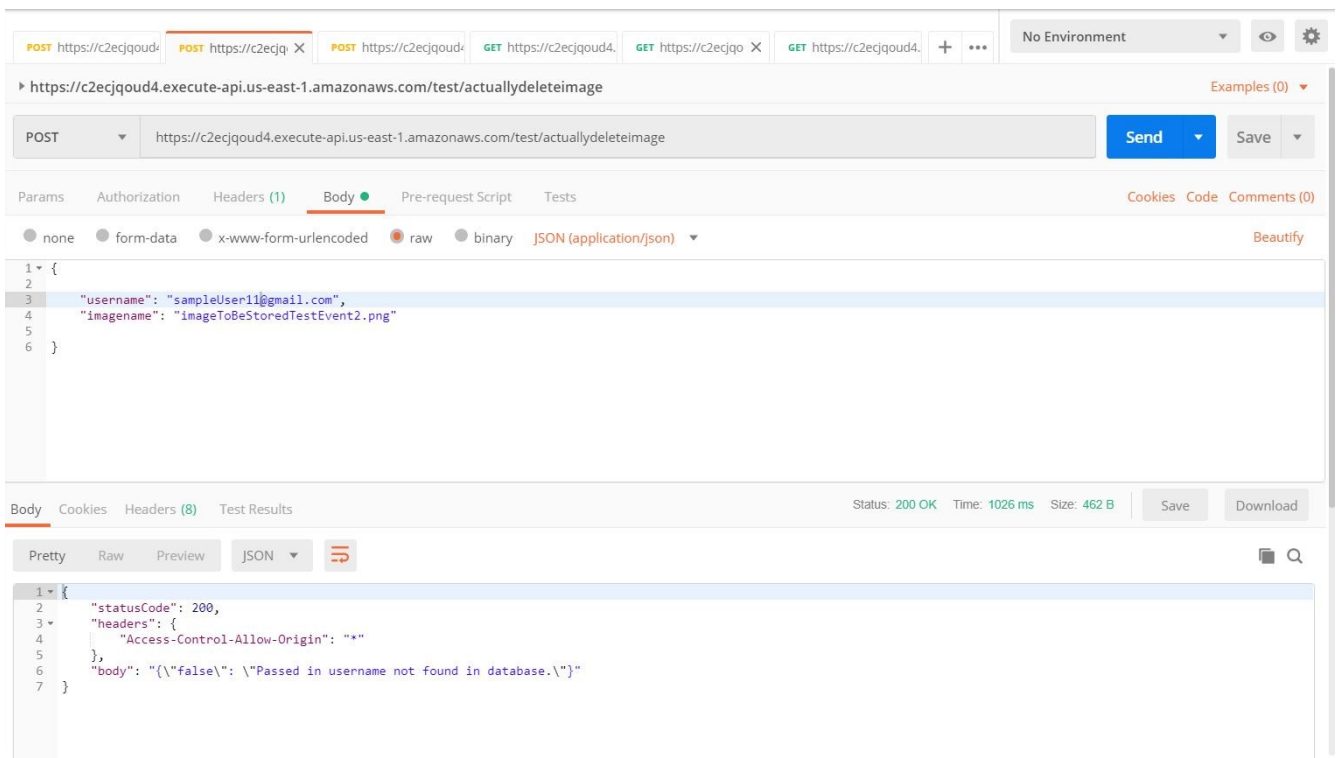
The screenshot shows a Postman interface for a POST request to `https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/actuallydeleteimage`. The request body is a JSON object:

```
1 {
2   "username": "sampleUser1@gmail.com",
3   "imagename": "imageToBeStoredTestEvent2.png"
4 }
5
6
```

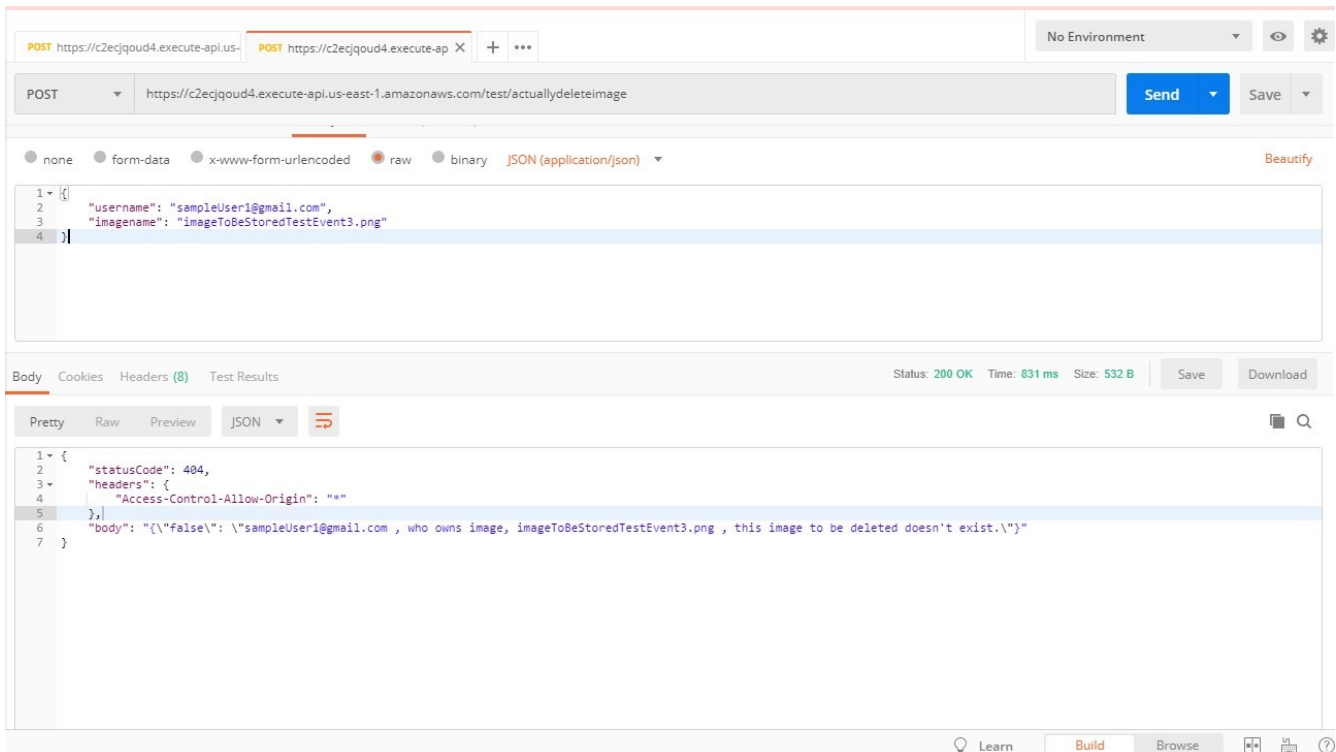
The response is a JSON object with the following structure:

```
1 {
2   "statusCode": 200,
3   "headers": {
4     "Access-Control-Allow-Origin": "*"
5   },
6   "body": "{\\\"username\\\": \\\"sampleUser1@gmail.com\\\", \\\"categories\\\": \\\"[]\\\", \\\"imageName\\\": \\\"[\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\"]\\\", \\\"refToImage\\\": \\\"[\\\"https://s3.amazonaws.com/imageuploader-main-bucket/All_User_Images/sampleUser140gmail.com*imageToBeStoredTestEvent1.png\\\"]\\\", \\\"imgDictByTag\\\": \\\"[\\\"tree\\\": [\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\"], \\\"canopy\\\": [\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\"], \\\"fractal\\\": [\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\"], \\\"colorful\\\": [\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\"], \\\"event postman\\\": [\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\"], \\\"postman event\\\": [\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\"]\\\", \\\"canView\\\": \\\"[\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\": [\\\"sampleUser1@gmail.com\\\"]\\\", \\\"imgDictByImage\\\": \\\"[\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\": [\\\"tree\\\", \\\"fractal\\\", \\\"colorful\\\", \\\"postman event\\\", \\\"event postman\\\", \\\"canopy\\\"]\\\"]\\\"}"
7 }
```

Screenshot of when given username is not found:



Screenshot of when image of an existing user cannot be found:



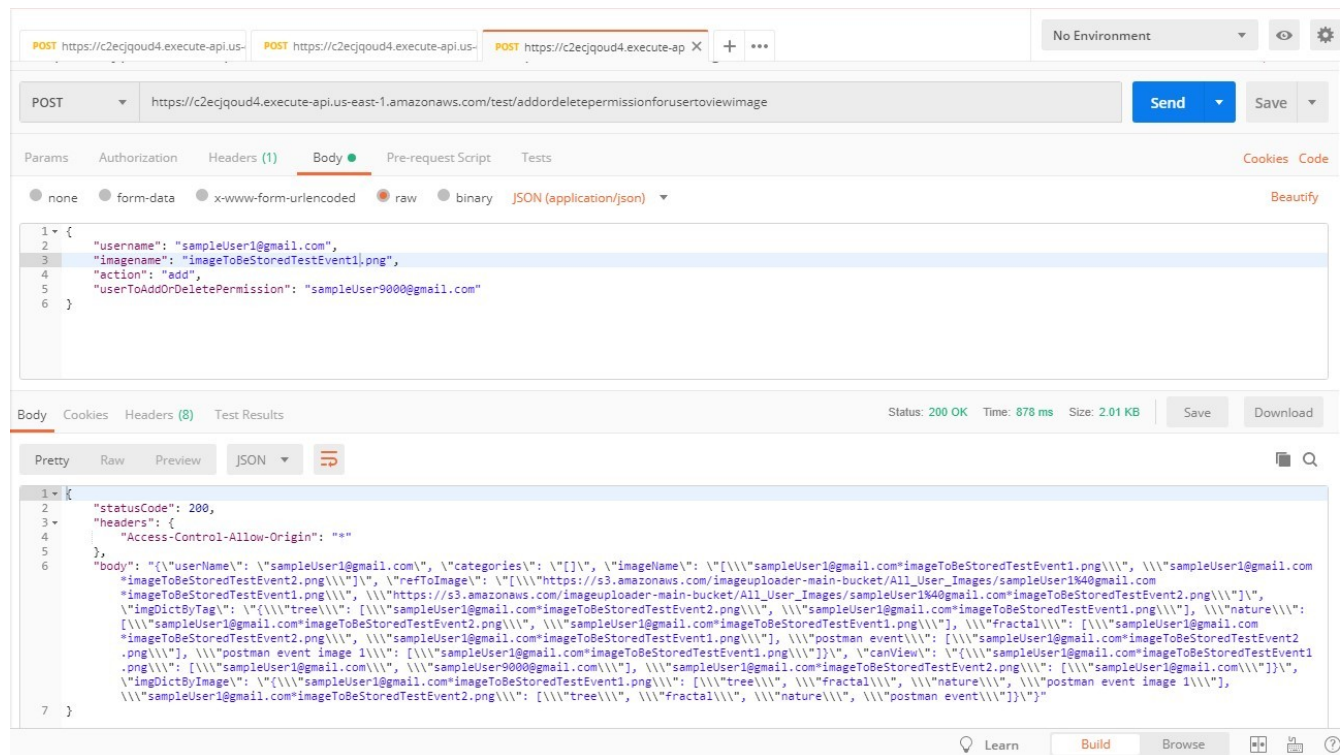
POST:

<https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/addordeletepermissionforusertoviewimage>

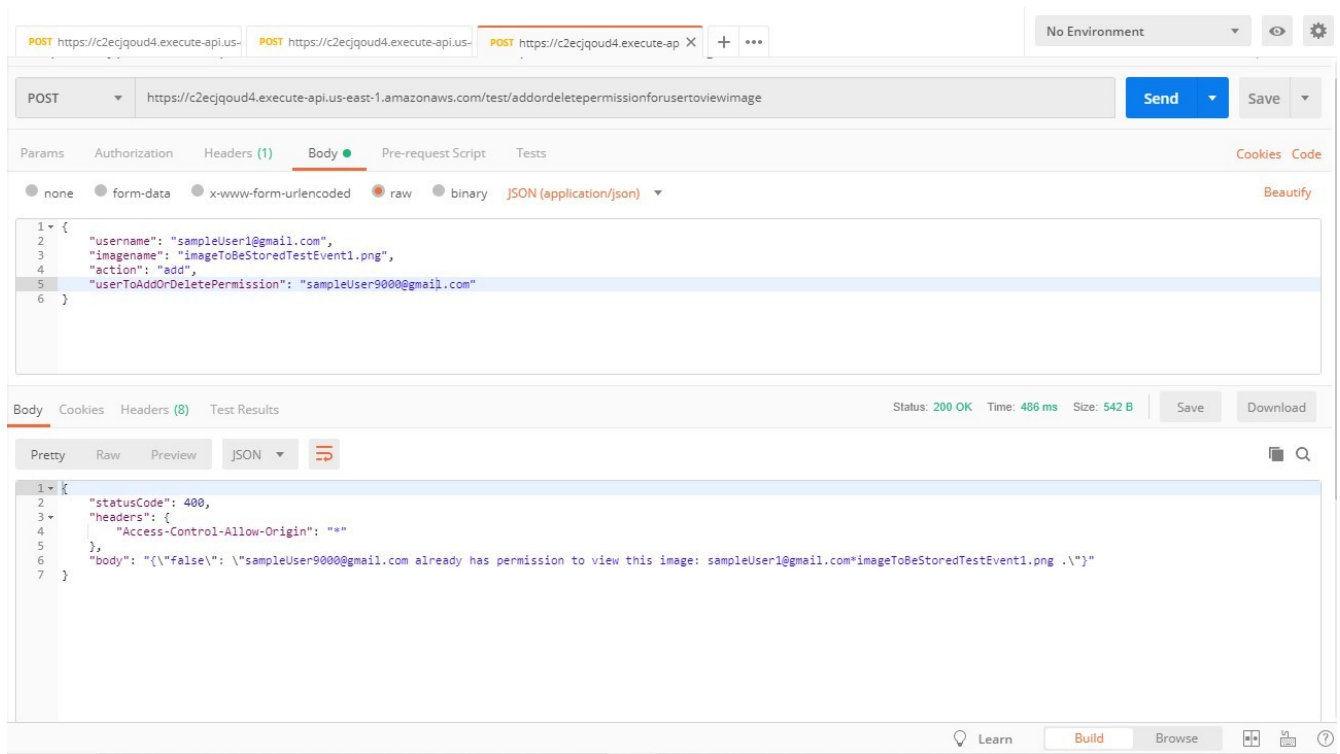
Description: Given an existing username (the owner) and image name, can either "add" or "delete" the access authorization for userToAddOrDeletePermission (the non-owner).

Note: When giving/revoking authorization to a user that is not the owner of the image, the user does not necessarily have to be an existing user.

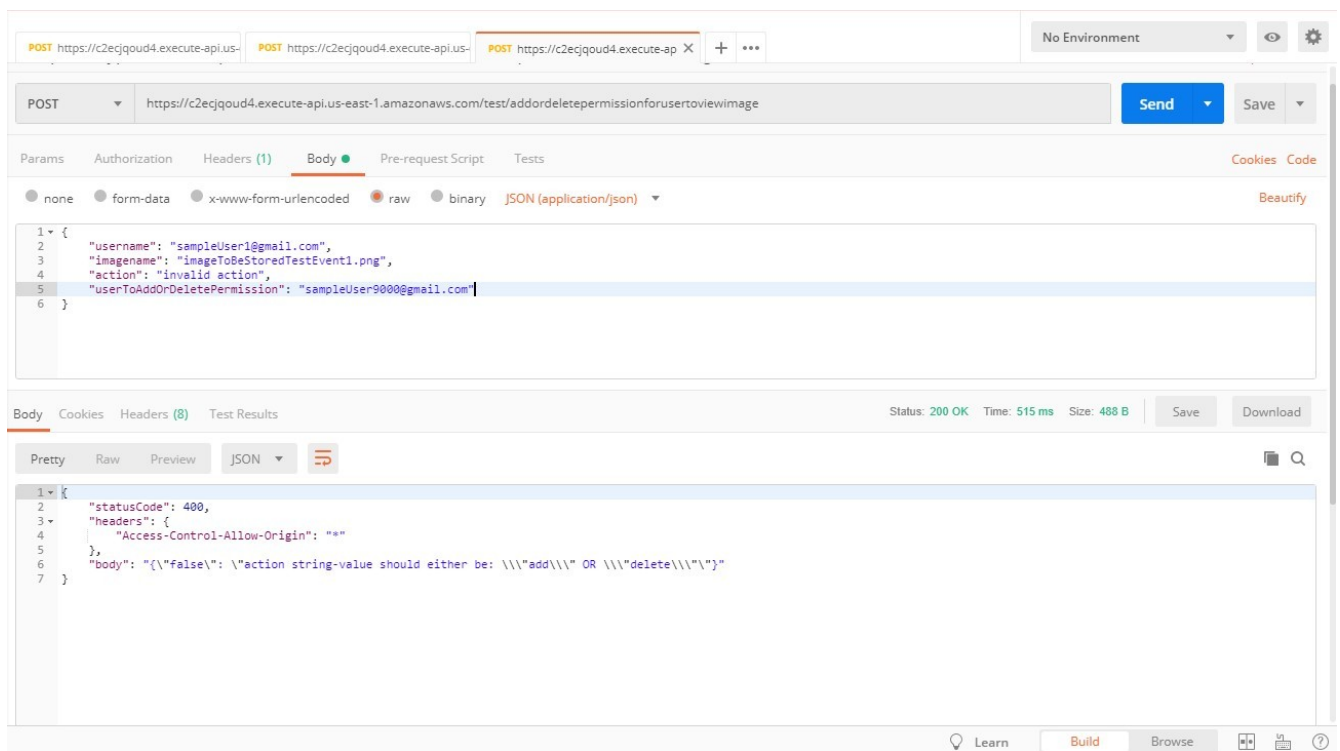
Screenshot of giving access to a user that isn't the owner ("username") of the image:



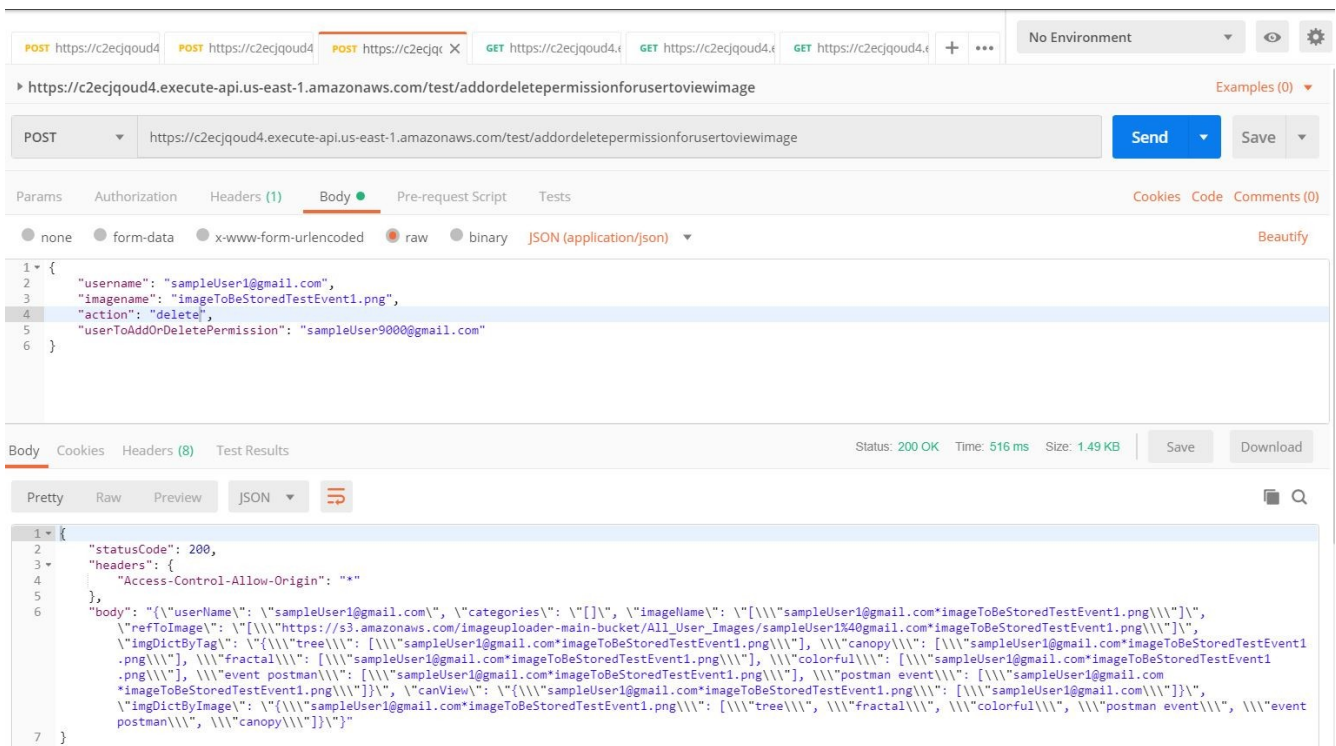
Screenshot of when attempting to give a non-owner (already has access) access to an image:



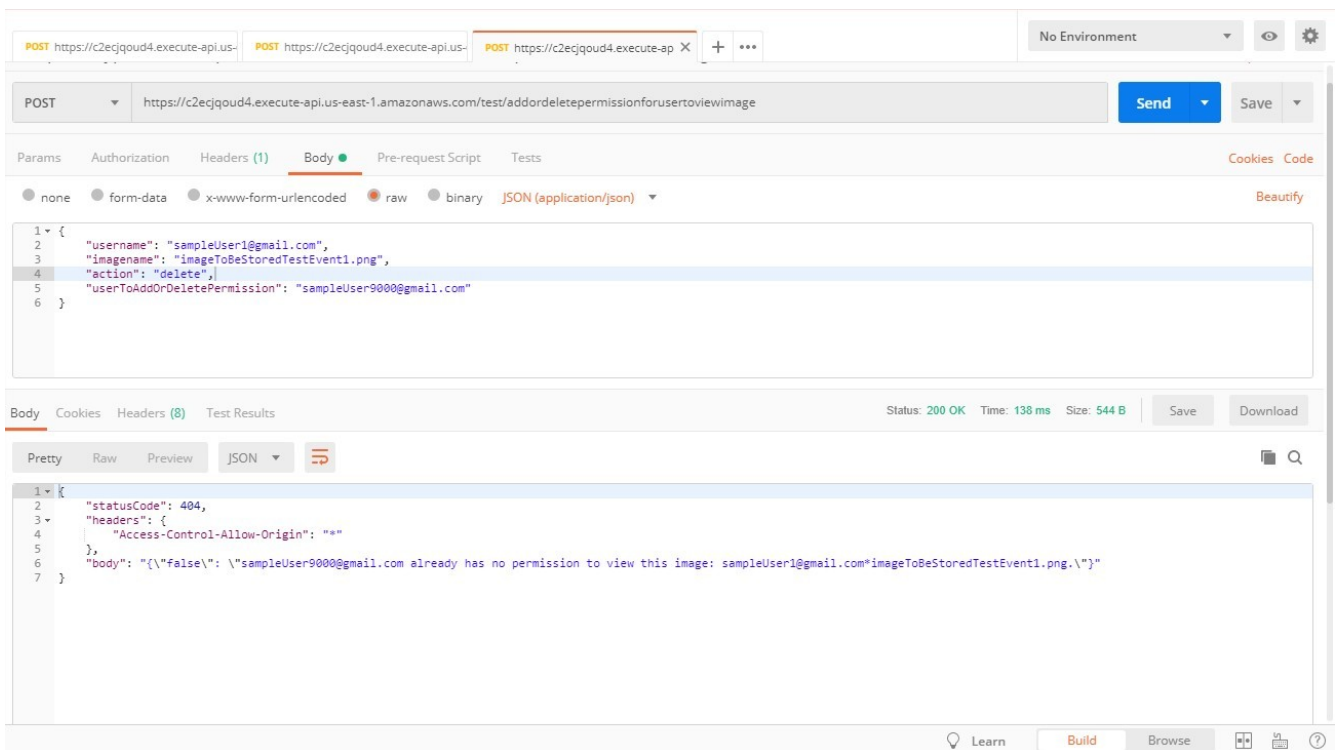
Screenshot of when attempting to pass an “action” that is neither "add" nor "delete":



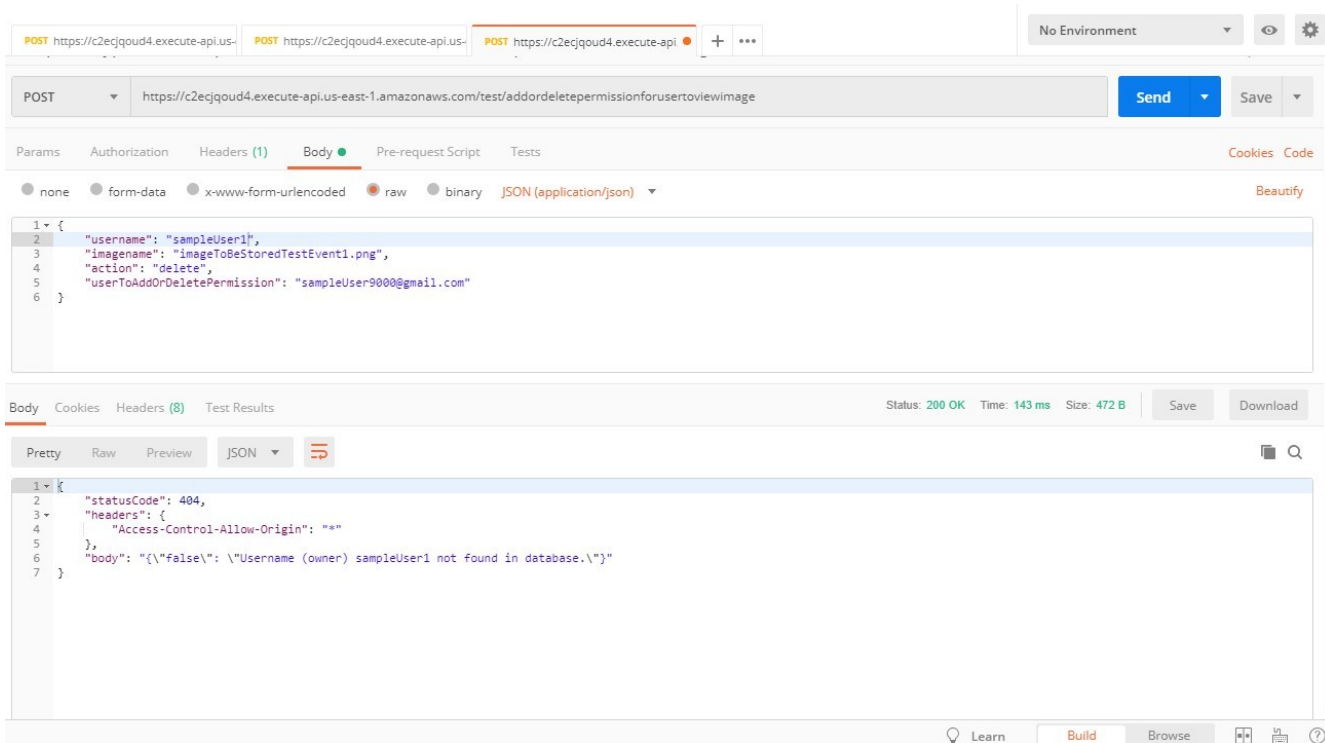
Screenshot of revoking access to a user that isn't the owner (“username”) of the image:



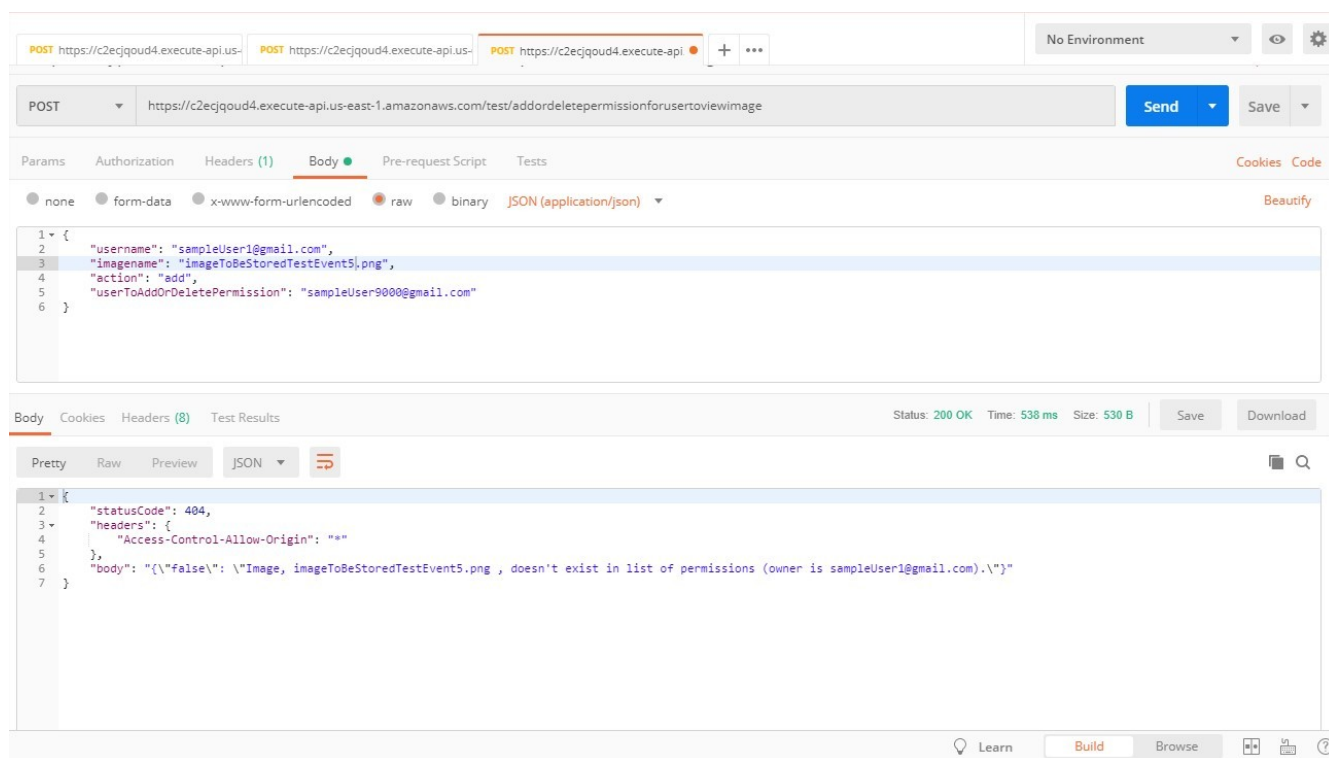
Screenshot of when attempting to revoke non-owner (already had no access) access to an image:



Screenshot of when username (owner) doesn't exist in database:



Screenshot of when given a valid owner (username), an image's viewing permissions are attempted to be modified with either "add" or "delete", but the image itself does not currently exist.



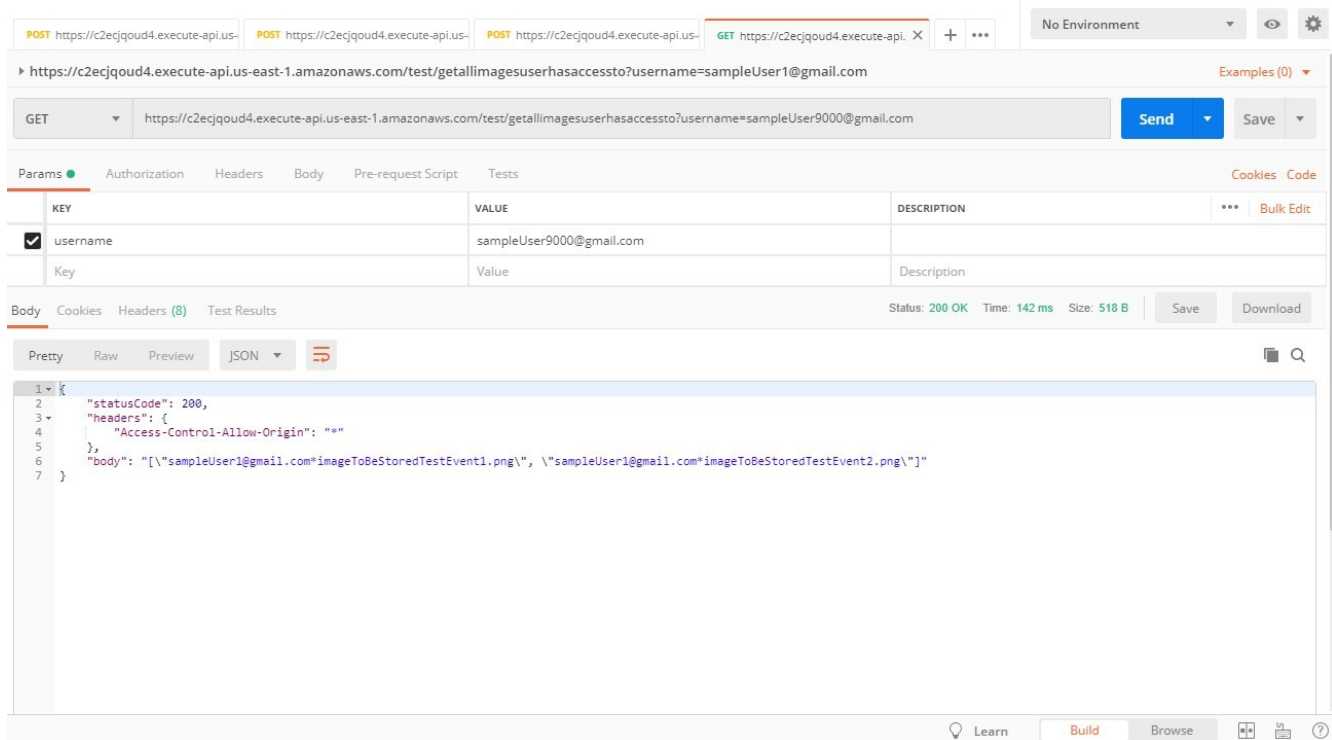
GET:

<https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/getallimagesuserhasaccessto?username=sampleUser9000@gmail.com>

Description: Given a username, return a list of all images shared with username. The list EXCLUDES images that the username owns. Each image's name format is "[username@domain.com](#)*sampleImageName.png".

If username cannot be found in database, an empty list is returned.

Screenshot:



GET:

<https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/getuserimages?username=sampleUser1@gmail.com>

Description: Use this endpoint to get a dictionary of all images, each with a list of tags, for a particular user. Username is the owner of the images. This essentially gets all images owned by username where each image is in format of "[username@domain.com](#)*sampleImageName.png".

POST https://c2ecjqoud4.e... POST https://c2ecjqoud4.e... POST https://c2ecjqoud4.e... GET https://c2ecjqoud4.ex... GET https://c2ecjqoud4.ex... GET https://c2ecjqoud4.X + ... No Environment

► https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/getuserimages?username=sampleUser1@gmail.com Examples (0)

GET https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/getuserimages?username=sampleUser1@gmail.com Send Save

Params Authorization Headers Body Pre-request Script Tests Cookies Code

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> username	sampleUser1@gmail.com	
Key	Value	Description

Body Cookies Headers (8) Test Results Status: 200 OK Time: 872 ms Size: 638 B Save Download

Pretty Raw Preview JSON

```
1 {
2   "statusCode": 200,
3   "headers": {
4     "Access-Control-Allow-Origin": ""
5   },
6   "body": "{\\\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\\\": [\\\"tree\\\", \\\"fractal\\\", \\\"nature\\\", \\\"postman event image 1\\\"], \\\"sampleUser1@gmail.com*imageToBeStoredTestEvent2.png\\\": [\\\"tree\\\", \\\"fractal\\\", \\\"nature\\\", \\\"postman event\\\"]}"
7 }
```

Screenshot of when user is not found in database:

POST https://c2ecjqoud4.e... POST https://c2ecjqoud4.e... POST https://c2ecjqoud4.e... GET https://c2ecjqoud4.ex... GET https://c2ecjqoud4.ex... GET https://c2ecjqoud4. No Environment

► https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/getuserimages?username=sampleUser1@gmail.com Examples (0)

GET https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/getuserimages?username=sampleUser4@gmail.com Send Save

Params Authorization Headers Body Pre-request Script Tests Cookies Code

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> username	sampleUser4@gmail.com	
Key	Value	Description

Body Cookies Headers (8) Test Results Status: 200 OK Time: 156 ms Size: 462 B Save Download

Pretty Raw Preview JSON

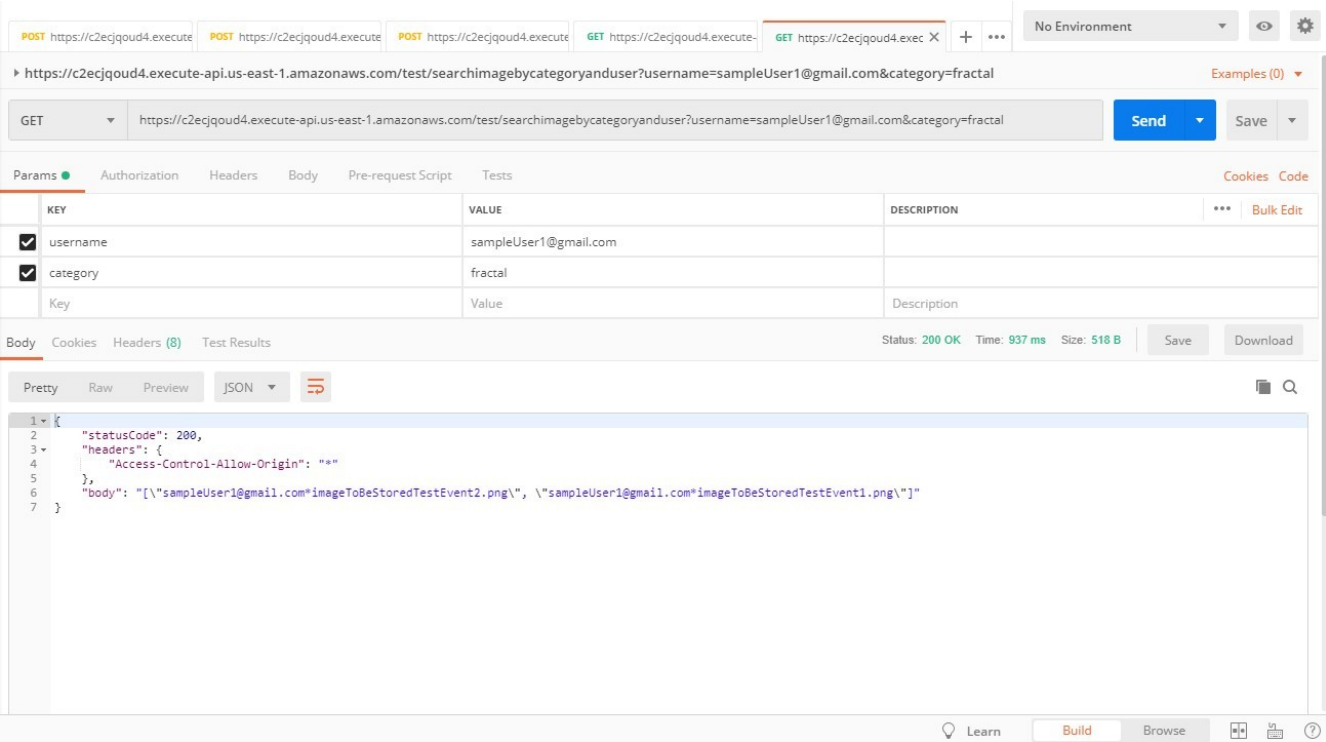
```
1 {
2   "statusCode": 404,
3   "headers": {
4     "Access-Control-Allow-Origin": ""
5   },
6   "body": "{\\\"false\\\": \\\"Username sampleUser4@gmail.com not found.\\\"}"
7 }
```

GET:

<https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/searchimagebycategoryanduser?username=sampleUser1@gmail.com&category=fractal>

Description: Given an existing username (owner) and category (or tag), returns the list of images (must extract image name from each corresponding element in the list) associated with the passed in tag. Each image's name format is "username@domain.com*sampleImageName.png".

Screenshot:



Screenshot of when username specified doesn't exist in database:

POST https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/searchimagebycategoryanduser?username=sampleUser1@gmail.com&category=fractal

GET https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/searchimagebycategoryanduser?username=sampleUser2000@gmail.com&category=fractal

Send Save

Params Authorization Headers Body Pre-request Script Tests Cookies Code

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> username	sampleUser2000@gmail.com	
<input checked="" type="checkbox"/> category	fractal	
Key	Value	Description

Body Cookies Headers (8) Test Results Status: 200 OK Time: 182 ms Size: 468 B Save Download

Pretty Raw Preview JSON

```
1 {
2   "statusCode": 404,
3   "headers": {
4     "Access-Control-Allow-Origin": ""
5   },
6   "body": "{\"false\": \"Username, sampleUser2000@gmail.com , not found.\"}"
7 }
```

Screenshot of when category not found for a user:

POST https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/searchimagebycategoryanduser?username=sampleUser1@gmail.com&category=fractal

GET https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/searchimagebycategoryanduser?username=sampleUser1@gmail.com&category=non-existent

Send Save

Params Authorization Headers Body Pre-request Script Tests Cookies Code

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> username	sampleUser1@gmail.com	
<input checked="" type="checkbox"/> category	non-existent	
Key	Value	Description

Body Cookies Headers (8) Test Results Status: 200 OK Time: 144 ms Size: 490 B Save Download

Pretty Raw Preview JSON

```
1 {
2   "statusCode": 404,
3   "headers": {
4     "Access-Control-Allow-Origin": ""
5   },
6   "body": "{\"false\": \"Category, non-existent ,not found for username sampleUser1@gmail.com \\\"}"
7 }
```

GET:

<https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/getuserswithpermissiontoviewimages?username=sampleUser1@gmail.com>

Description: Given an existing username (owner), returns a dictionary of all images (image name must be extracted first with some sort of string split() function where the delimiter is the '*' character), each with a list of usernames that are allowed to access their respective images.

Screenshot:

The screenshot shows a REST client interface with a GET request to the endpoint `https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/getuserswithpermissiontoviewimages?username=sampleUser1@gmail.com`. The request is successful, returning a 200 OK status with a response time of 871 ms and a size of 636 B. The response body is displayed in JSON format:

```
1 {
2   "statusCode": 200,
3   "headers": {
4     "Access-Control-Allow-Origin": "*"
5   },
6   "body": "{\"sampleUser1@gmail.com*imageToBeStoredTestEvent1.png\": [\"sampleUser1@gmail.com\", \"sampleUser9000@gmail.com\"], \"sampleUser1@gmail.com*imageToBeStoredTestEvent2.png\": [\"sampleUser1@gmail.com\", \"sampleUser9000@gmail.com\"]}"
7 }
```

Screenshot of when username (owner) not found:

POST https://c2ecjq

POST https://c2ecjq

POST https://c2ecjq

GET https://c2ecjqc

GET https://c2ecjqc

GET https://c2ecjqc

GET https://c2ecjqc

GET https://c2ecjqc

GET https://c2ecjqc

+

...

No Environment

▶ https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/getuserswithpermissiontoviewimages?username=sampleUser1@gmail.com

Examples (0)

GEThttps://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/getuserswithpermissiontoviewimages?username=sampleUser23@gmail.com

Send

Save

Params

Authorization

Headers

Body

Pre-request Script

Tests

Cookies

Code

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> username	sampleUser23@gmail.com			
Key	Value	Description		

Body

Cookies

Headers (8)

Test Results

Status: 200 OK

Time: 180 ms

Size: 463 B

Save

Download

Pretty

Raw

Preview

JSON

1

2

3

4

5

6

7

{

"statusCode": 404,

"headers": {

"Access-Control-Allow-Origin": ""

},

"body": "{\\"false\\": \\"Username sampleUser23@gmail.com not found.\\"}"

}

Learn

Build

Browse