

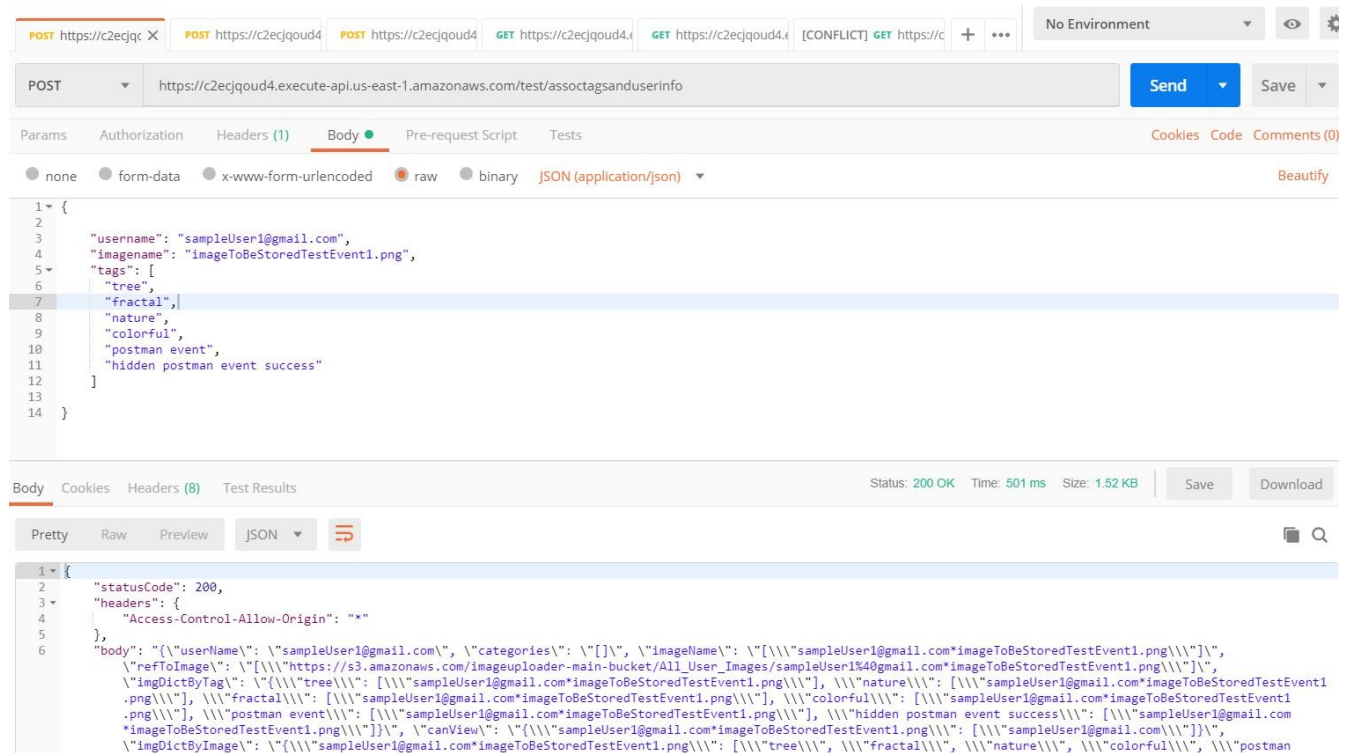
Updated 3/20/2019

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):



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

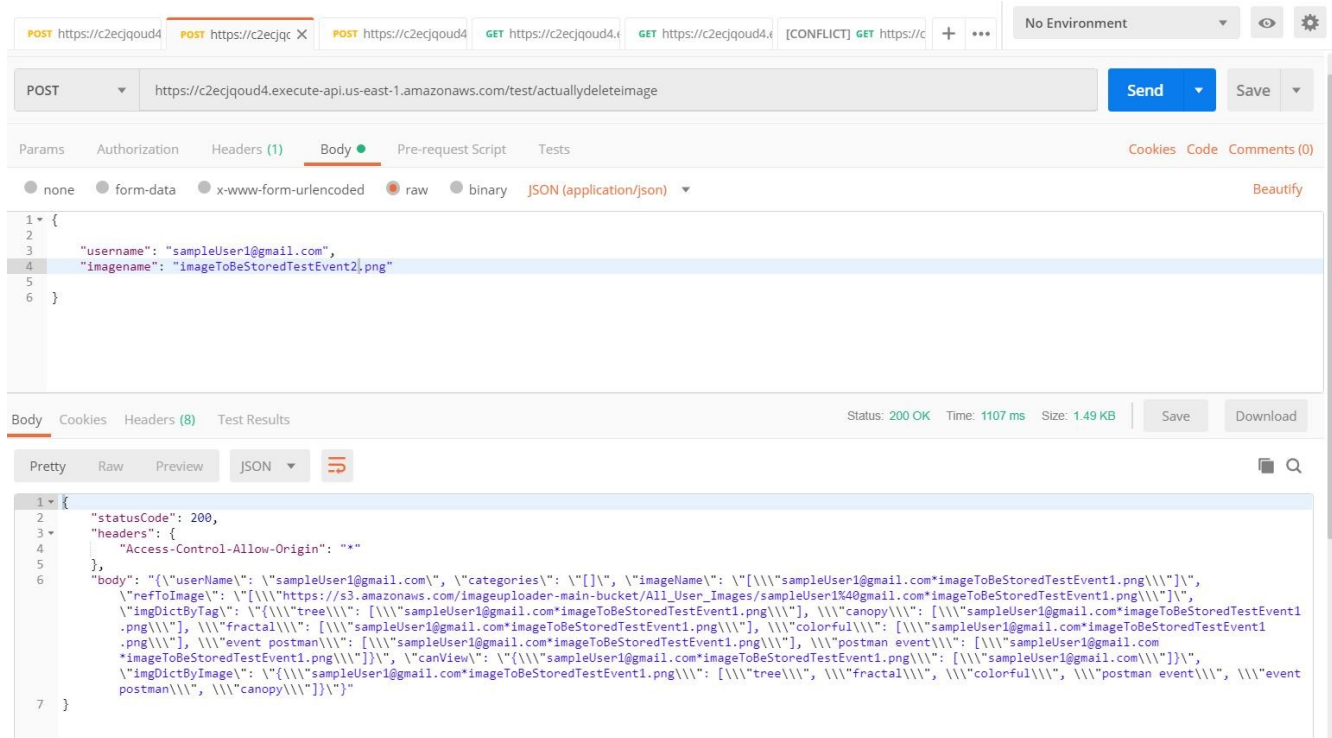


POST:

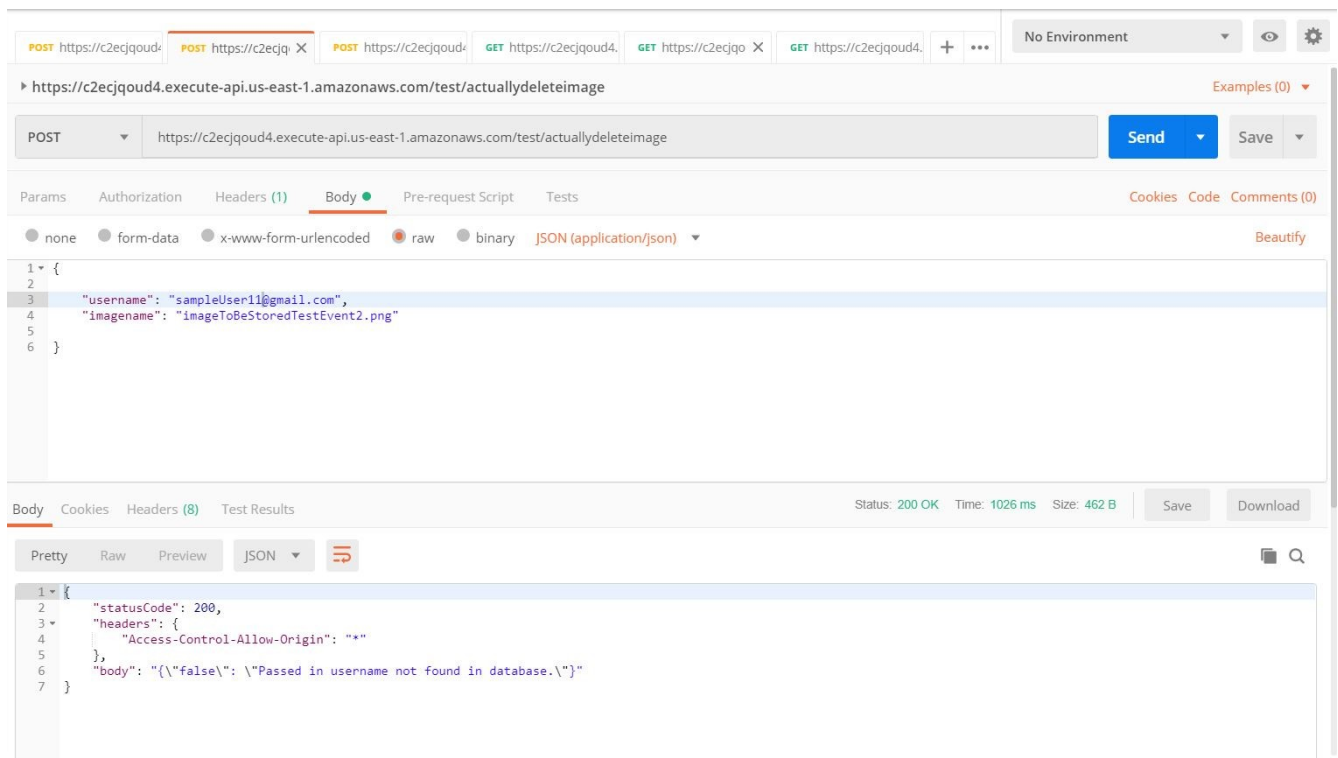
<https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/actuallydeleteimage>

Description: Deletes all images related to given user/owner.

Screenshot of info needed:



Screenshot of when given username is not found:

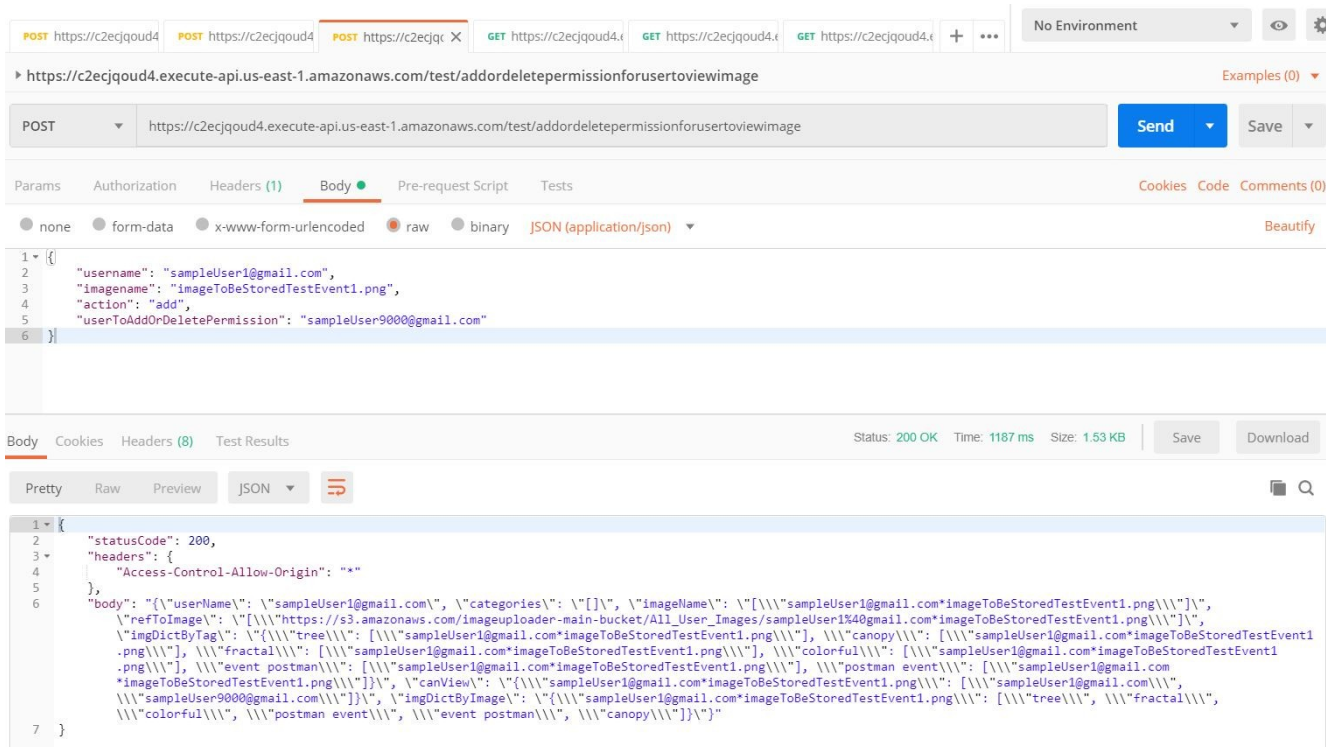


POST:

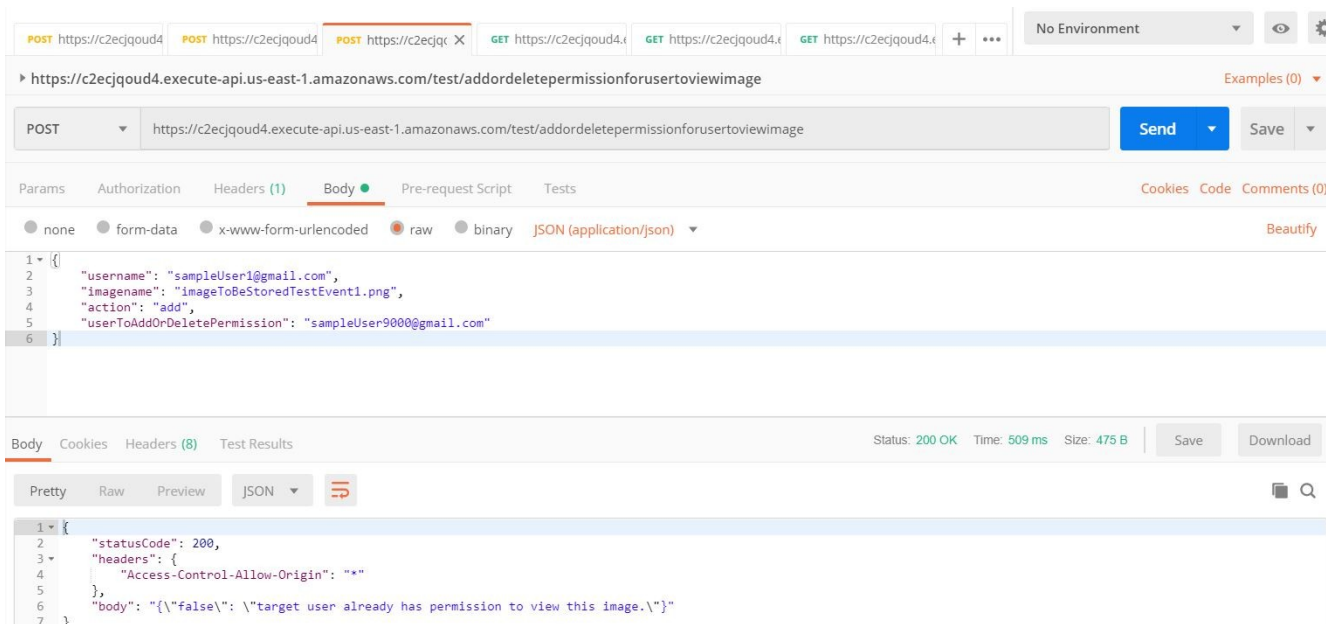
<https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/addordeletepermissionforusertoviewimage>

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

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

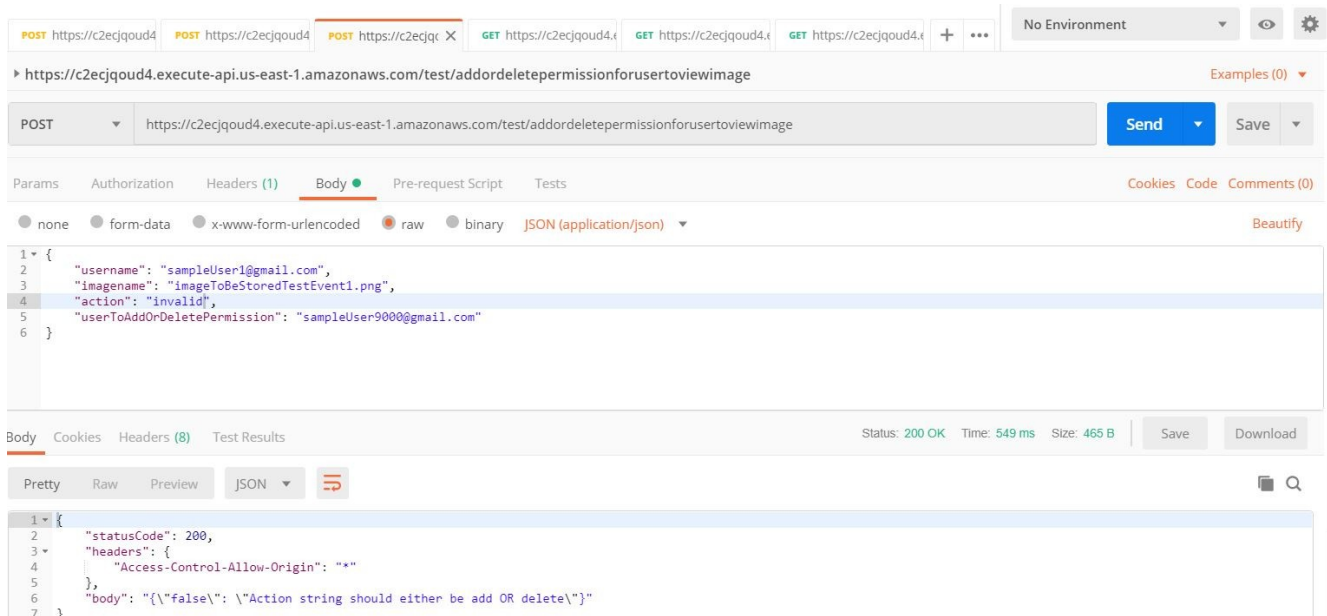


Screenshot of when attempting to give a non-owner (already has access) access to an image:

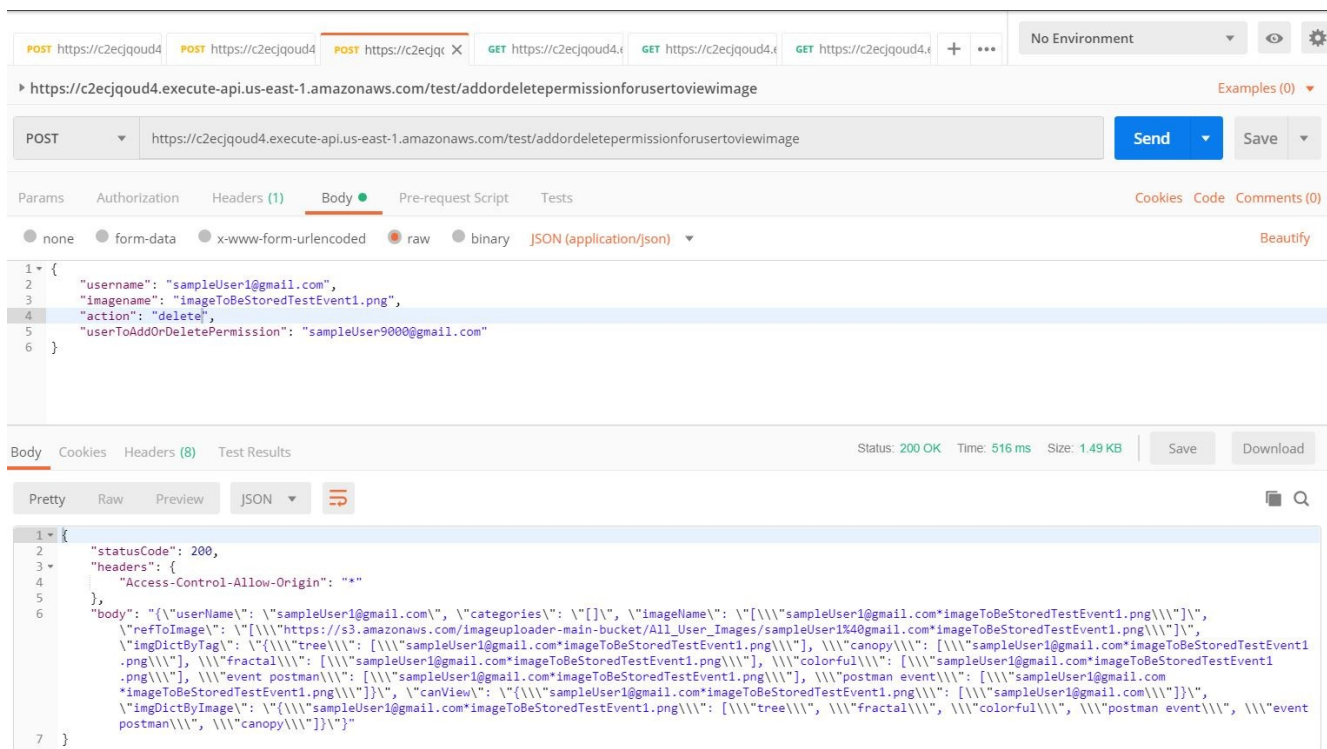


Screenshot of when attempting to pass an “action” that is neither "add" nor "delete":

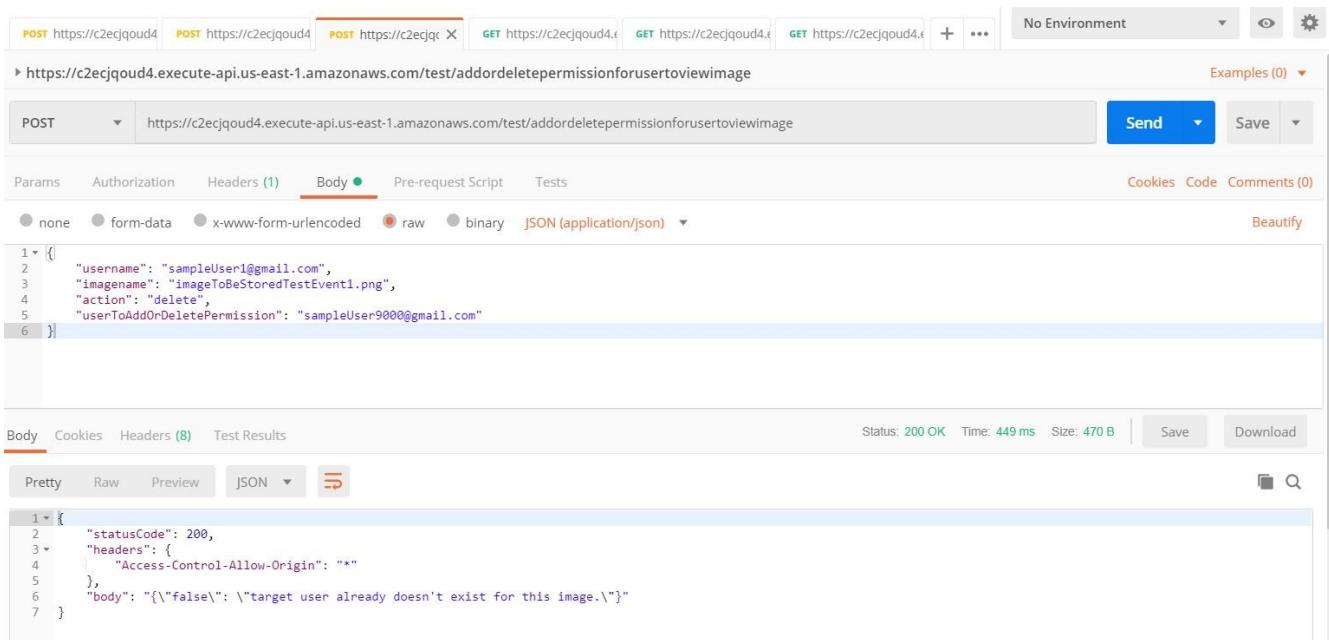




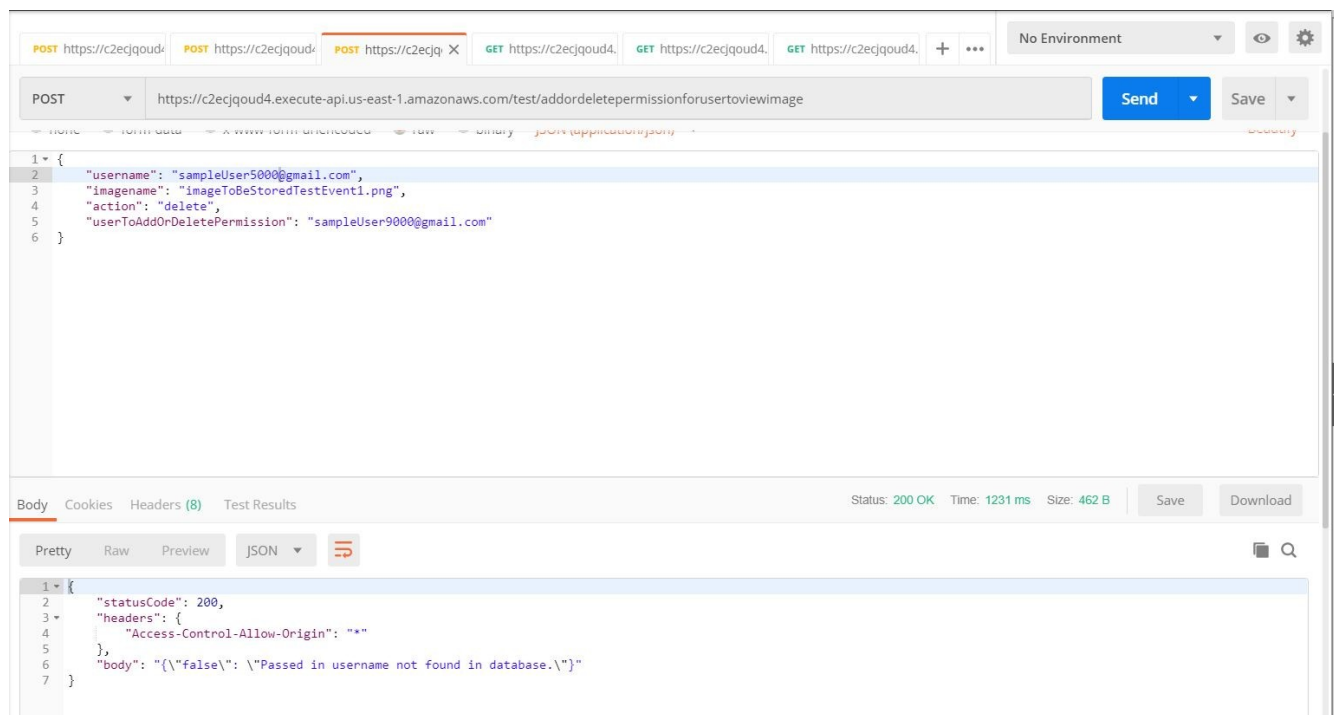
Screenshot of revoking access to a user that isn't the owner ("username") of the image:



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



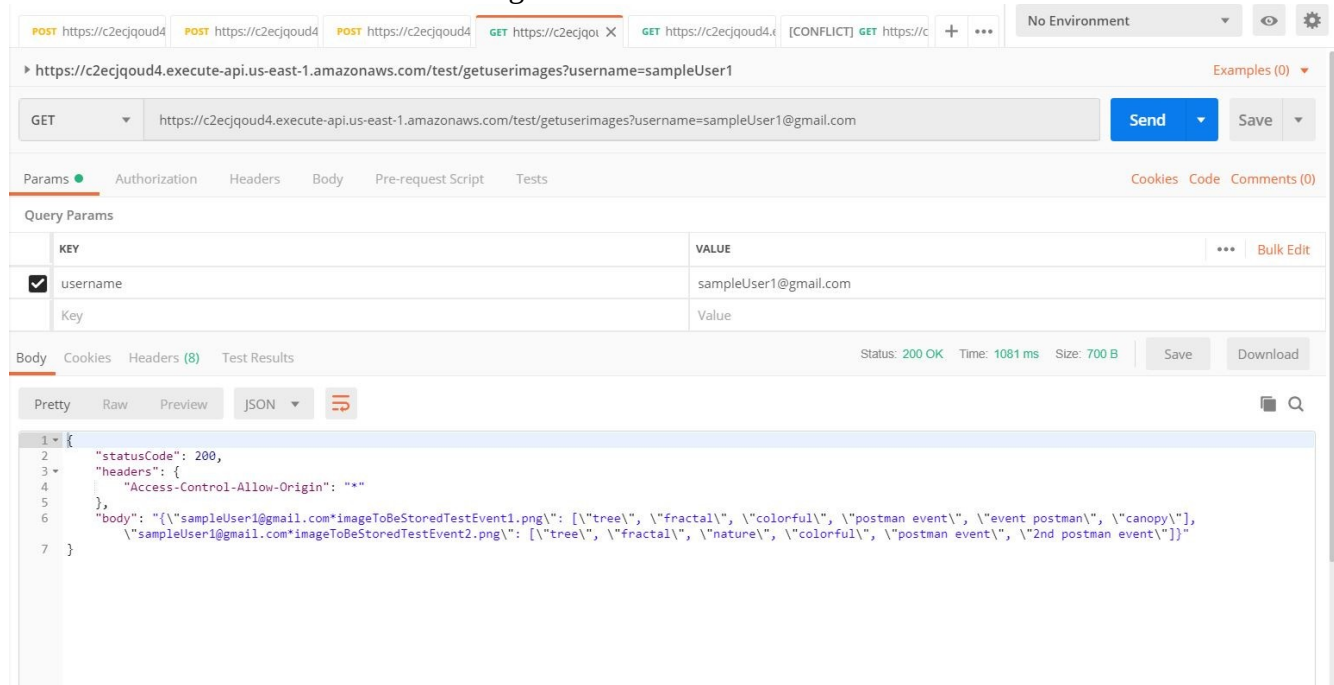
Screenshot of when username (owner) doesn't exist in database:



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.



GET:

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

Description: Given an existing username 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.

Screenshot:



The screenshot shows a REST client interface with a GET request to the following URL: `https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/searchimagebycategoryanduser?username=sampleUser1&category=fractal`. The request parameters are:

KEY	VALUE
username	sampleUser1@gmail.com
category	fractal

The response is a JSON object with the following structure:

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

GET:

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

Description: Given an existing username, 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:

POST https://c2ecjqoud4

POST https://c2ecjqoud4

POST https://c2ecjqoud4

GET https://c2ecjqoud4

GET https://c2ecjqoud4

GET https://c2ecjqoud4 X

+

...

No Environment

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

Examples (0) ▼

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

Send Save

Params Authorization Headers Body Pre-request Script Tests

Cookies Code Comments (0)

Query Params

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

Body Cookies Headers (8) Test Results

Status: 200 OK Time: 1084 ms Size: 490 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\": [\"sampleUser1@gmail.com\"]}"
7 }
```