

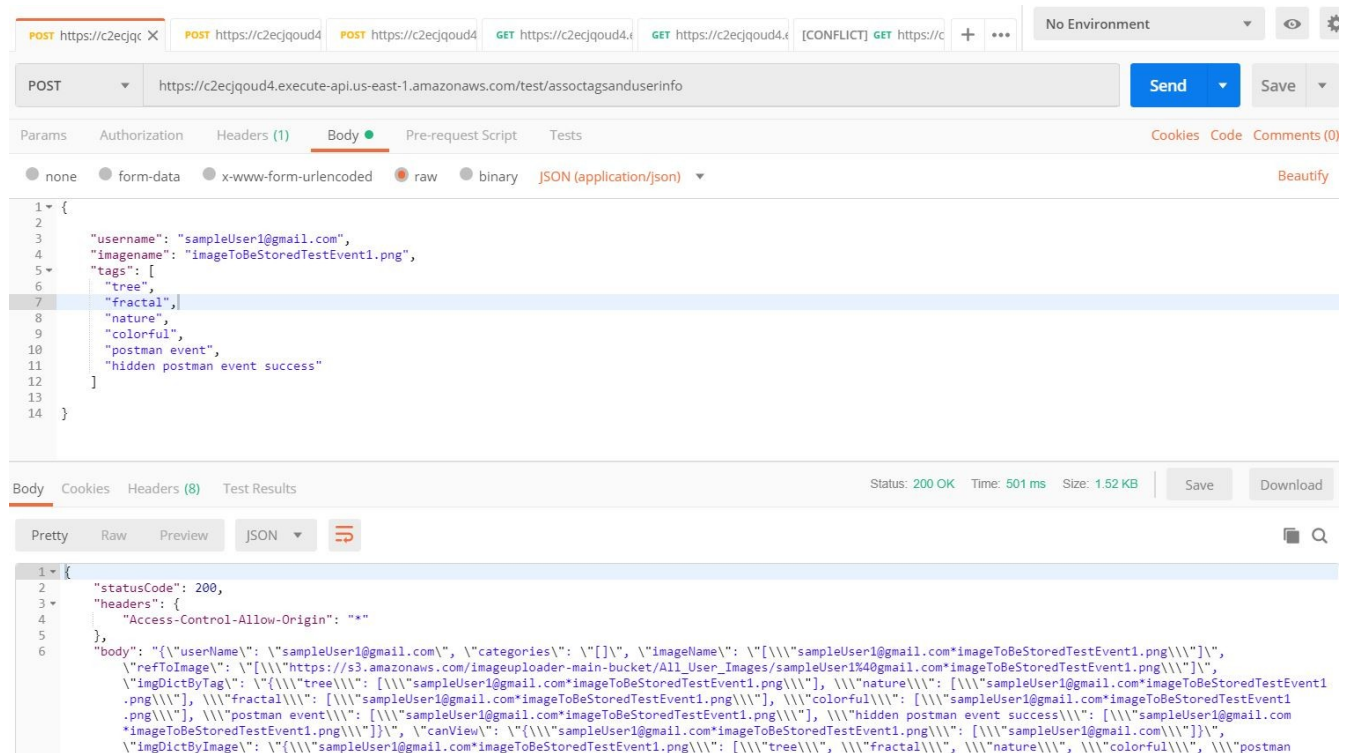
Updated 3/20/2019

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://c2ecjqou4 X POST https://c2ecjqou4 POST https://c2ecjqou4 GET https://c2ecjqou4 GET https://c2ecjqou4 [CONFLICT] GET https://c No Environment

POST https://c2ecjqou4.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://c2ecjqou4 X POST https://c2ecjqou4 POST https://c2ecjqou4 GET https://c2ecjqou4 GET https://c2ecjqou4 [CONFLICT] GET https://c No Environment

POST https://c2ecjqou4.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\\\"]\\\", \\\"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\\\"]\\\"}\\\"}"

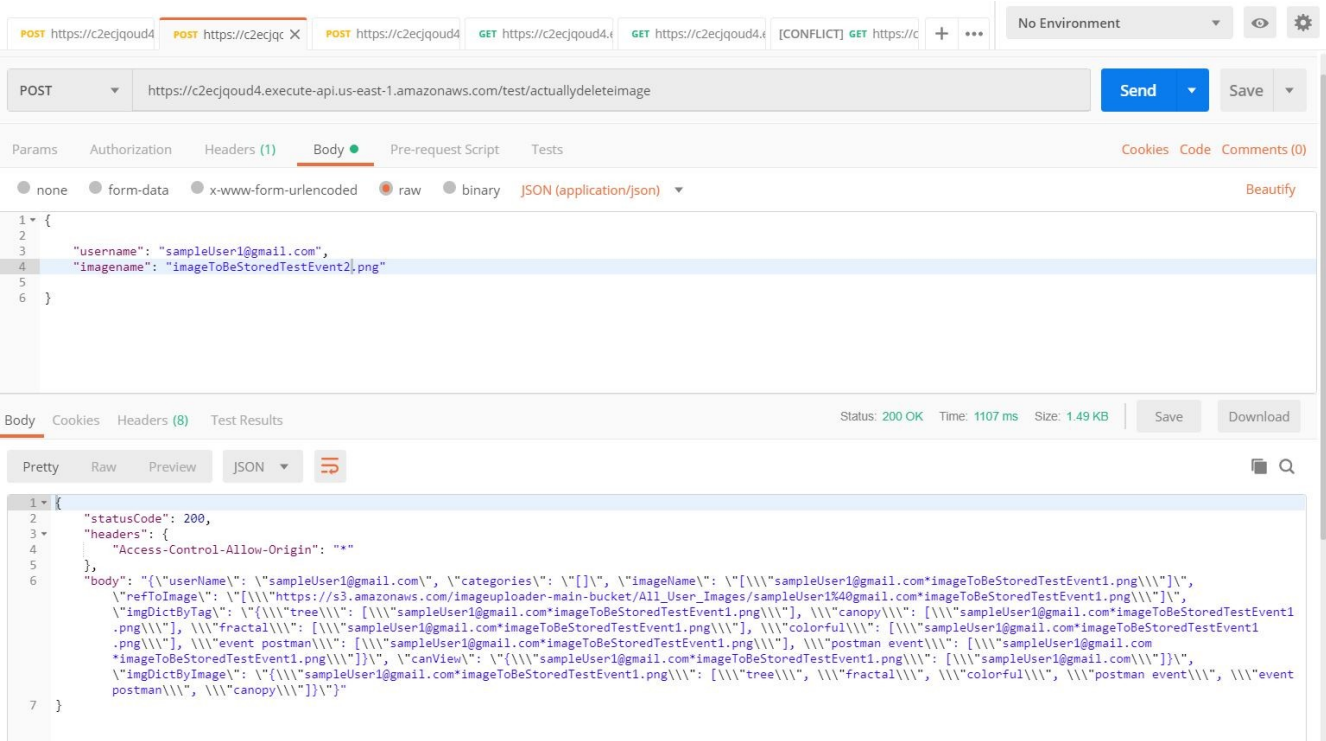
```

POST:

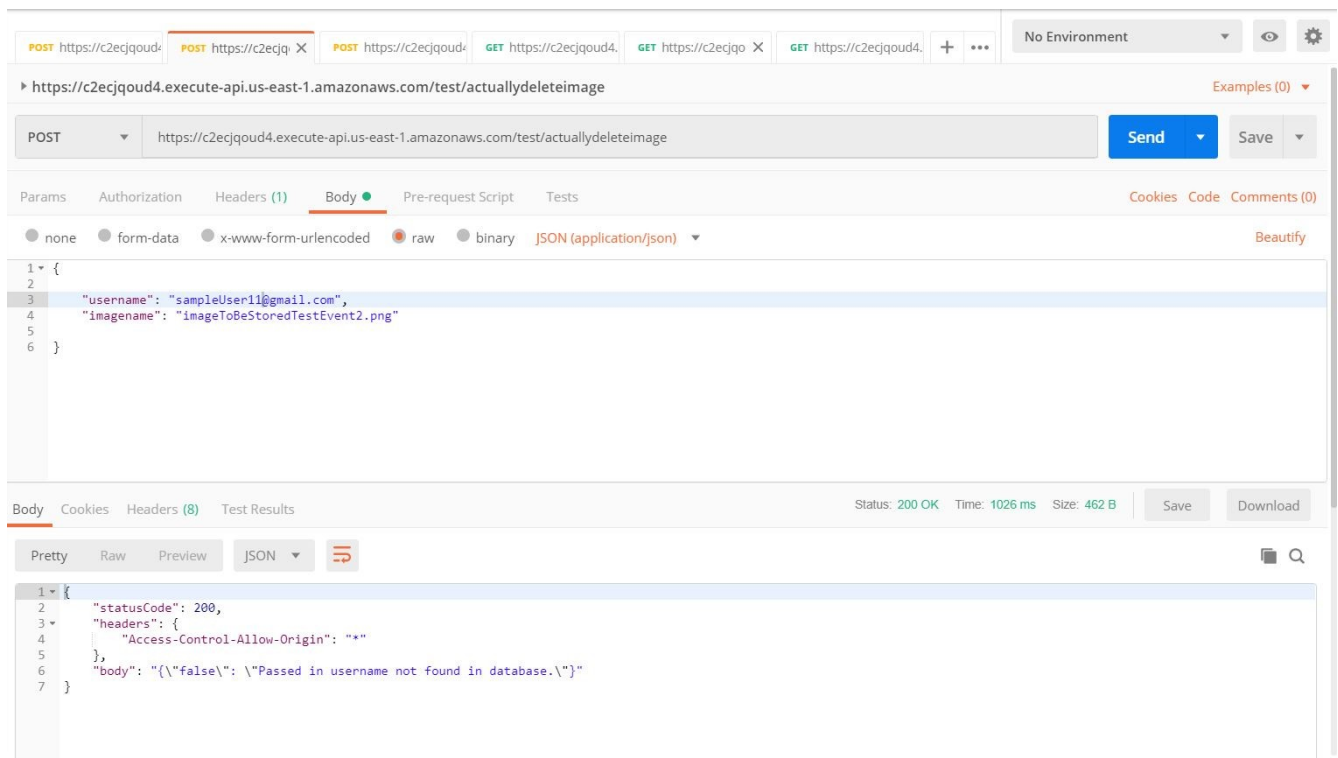
<https://c2ecjqou4.execute-api.us-east-1.amazonaws.com/test/actuallydeleteimage>

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

Screenshot of info needed:



Screenshot of when given username is not found:

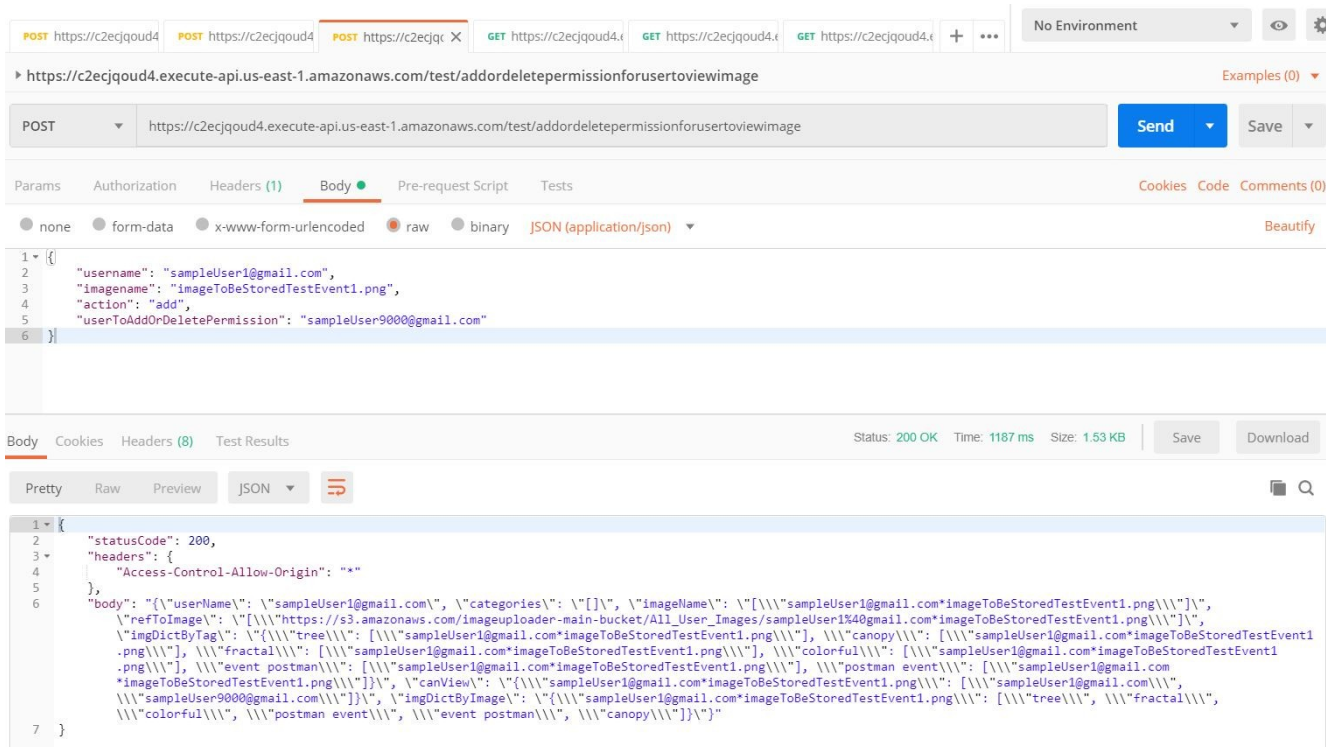


POST:

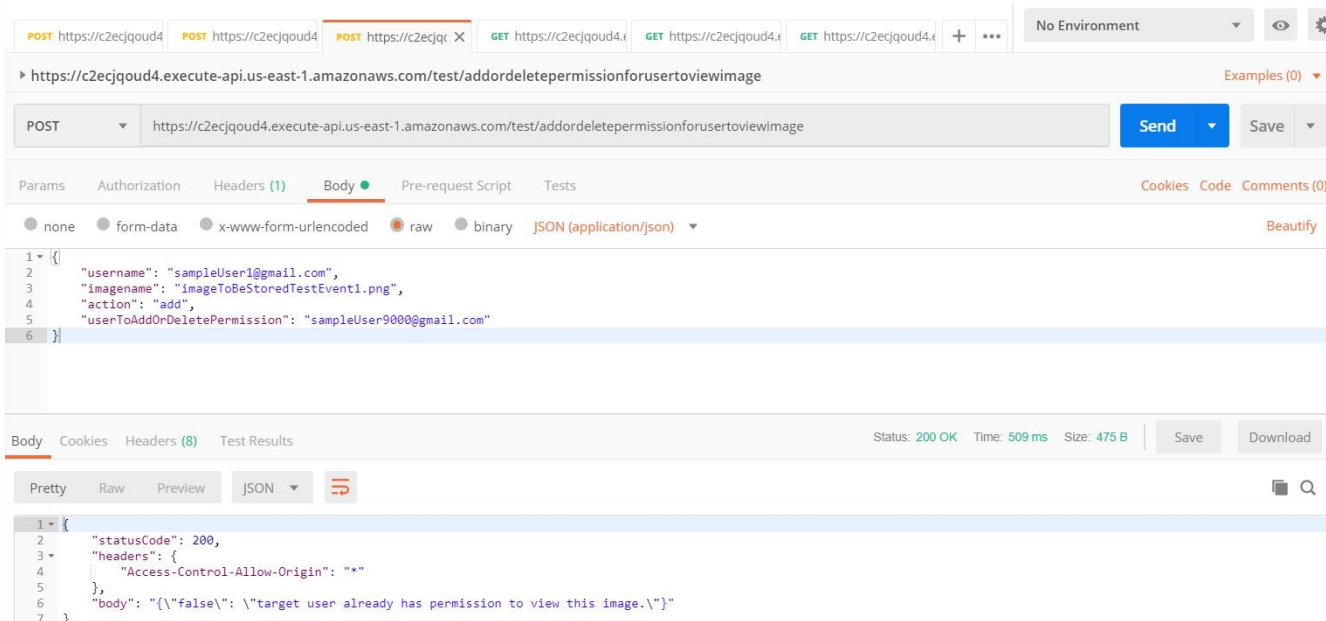
<https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/addordeletepermissionforusertoviewimage>

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

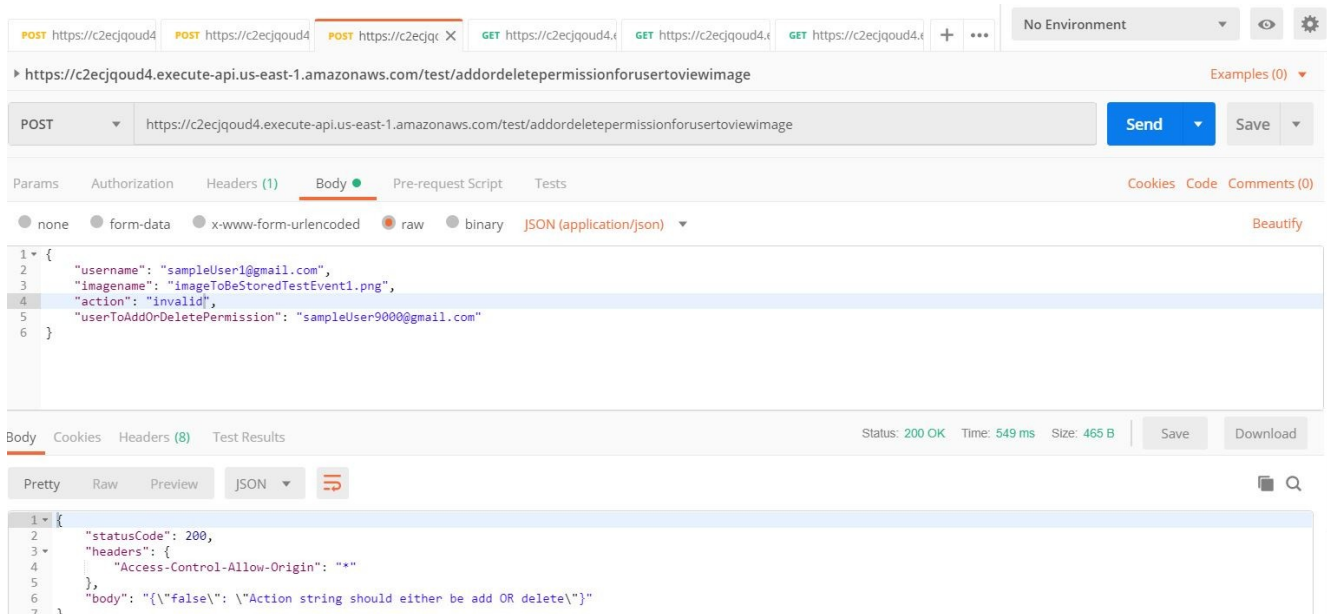
Screenshot of giving access to a user that isn't the owner ("username") of the image:



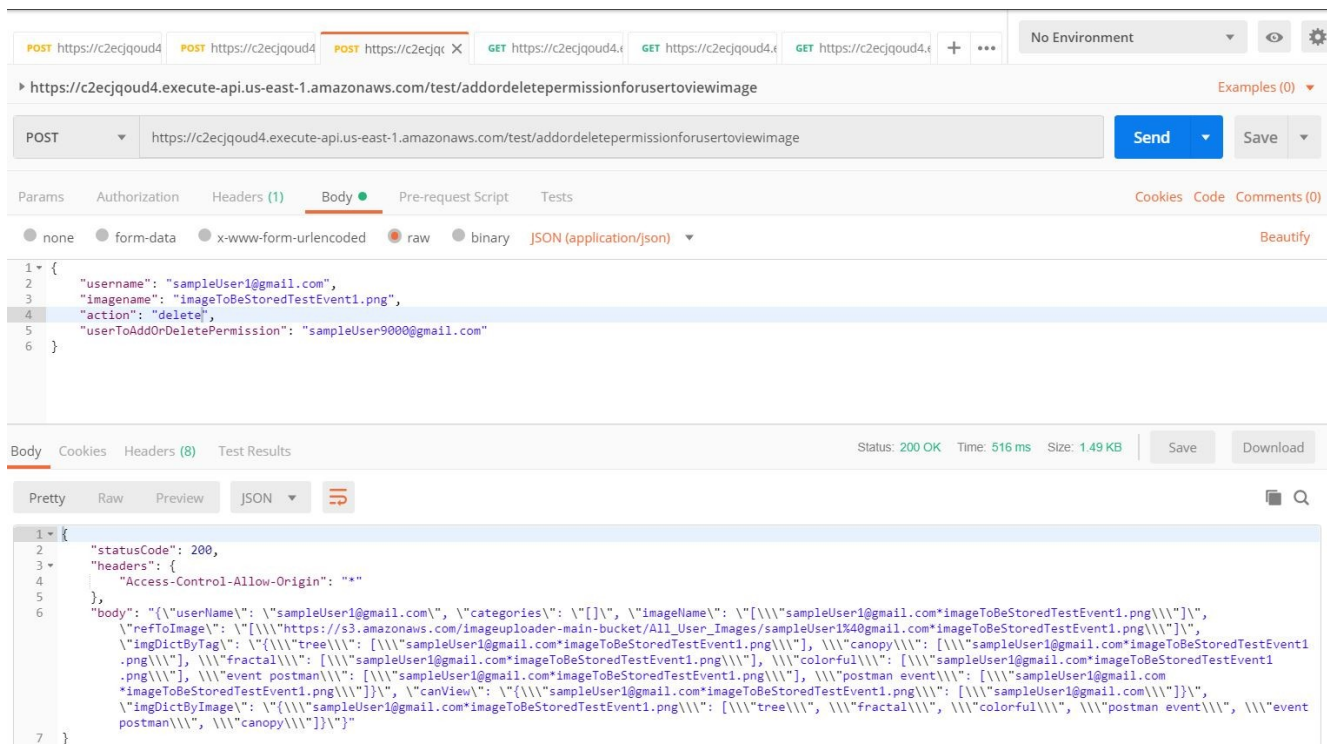
Screenshot of when attempting to give a non-owner (already has access) access to an image:



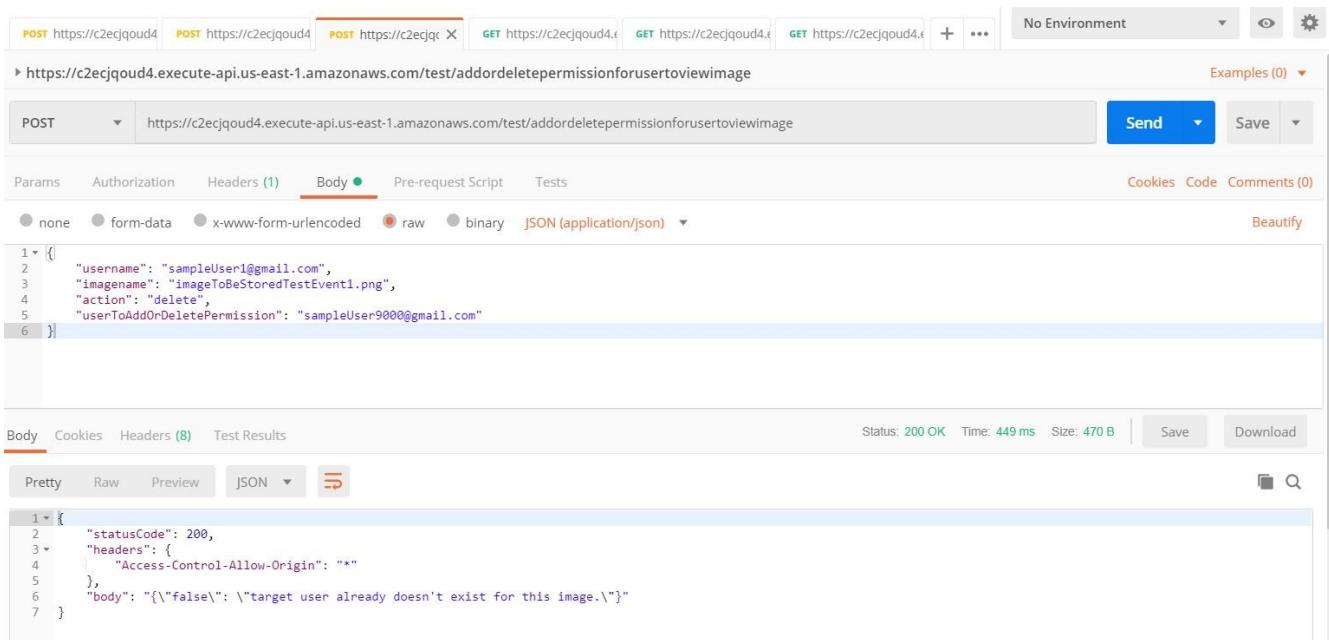
Screenshot of when attempting to pass an “action” that is neither "add" nor "delete":



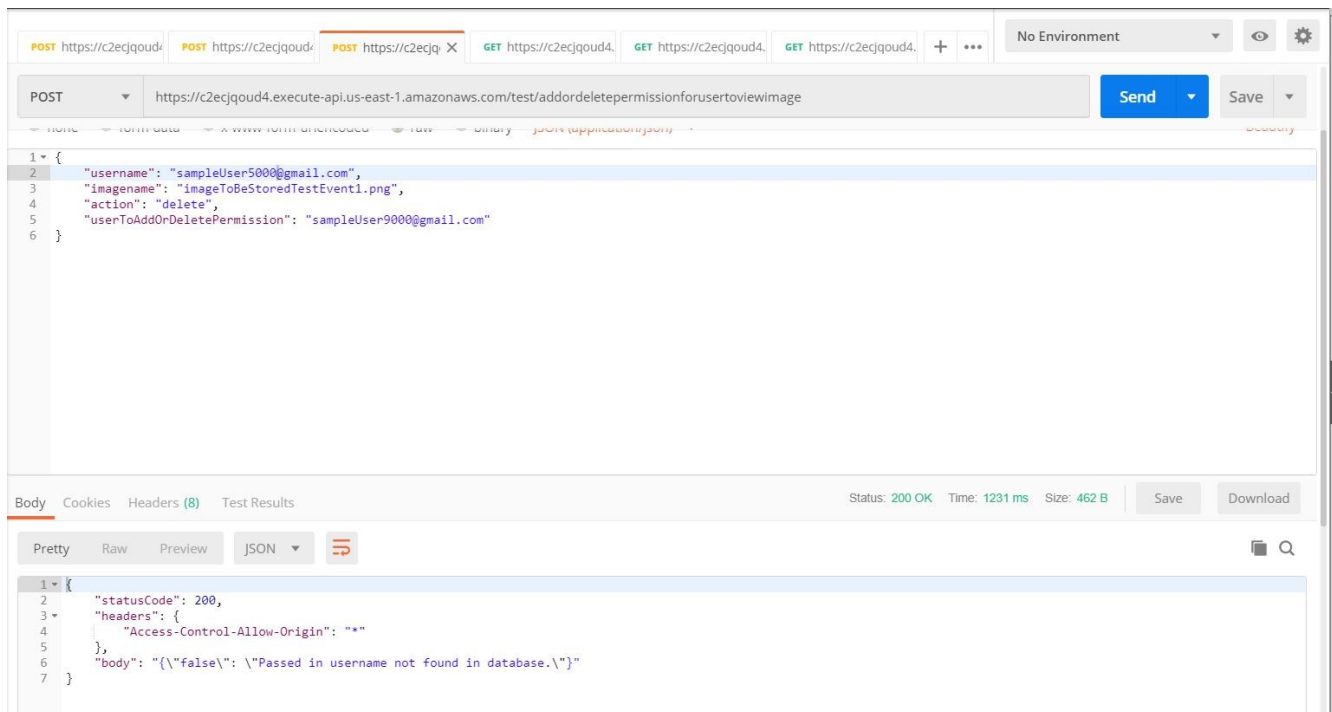
Screenshot of revoking access to a user that isn't the owner ("username") of the image:



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



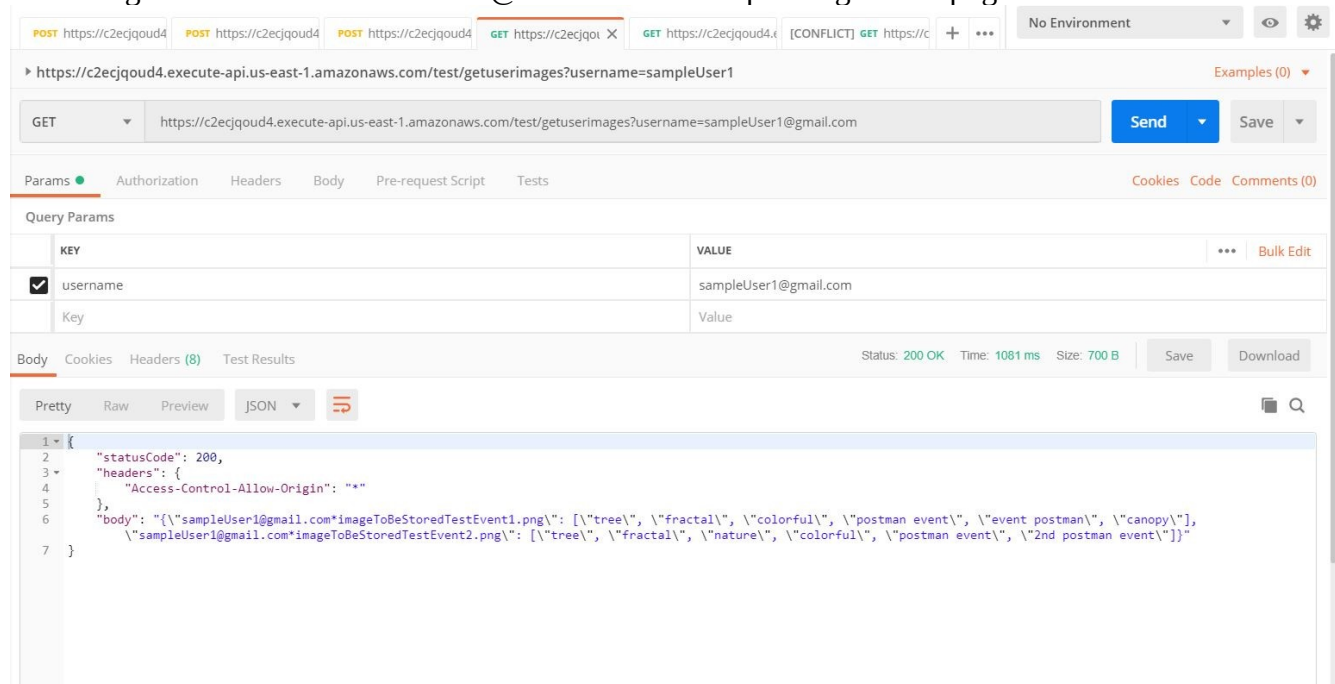
Screenshot of when username (owner) doesn't exist in database:



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. This essentially gets all images owned by username where each image is in format of “username@domain.com*sampleImageName.png”.



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. Each image’s name format is “username@domain.com*sampleImageName.png”.

Screenshot:

POST https://c2ecjqoud4 POST https://c2ecjqoud4 POST https://c2ecjqoud4 GET https://c2ecjqoud4 GET https://c2ecjqoi X GET https://c2ecjqoud4 + ... No Environment

► https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/searchimagebycategoryanduser?username=sampleUser1&category=fractal Examples (0)

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

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

Query Params

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

Body Cookies Headers (8) Test Results Status: 200 OK Time: 1167 ms Size: 461 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\"]"
7 }
```

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

POST https://c2ecjqoud4 POST https://c2ecjqoud4 POST https://c2ecjqoud4 GET https://c2ecjqoud4 GET https://c2ecjqoi GET https://c2ecjqoud4 + ... No Environment

► https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/searchimagebycategoryanduser?username=sampleUser1&category=fractal Examples (0)

GET https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/searchimagebycategoryanduser?username=sampleUser12@gmail.com&category=fractal Send Save

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

Query Params

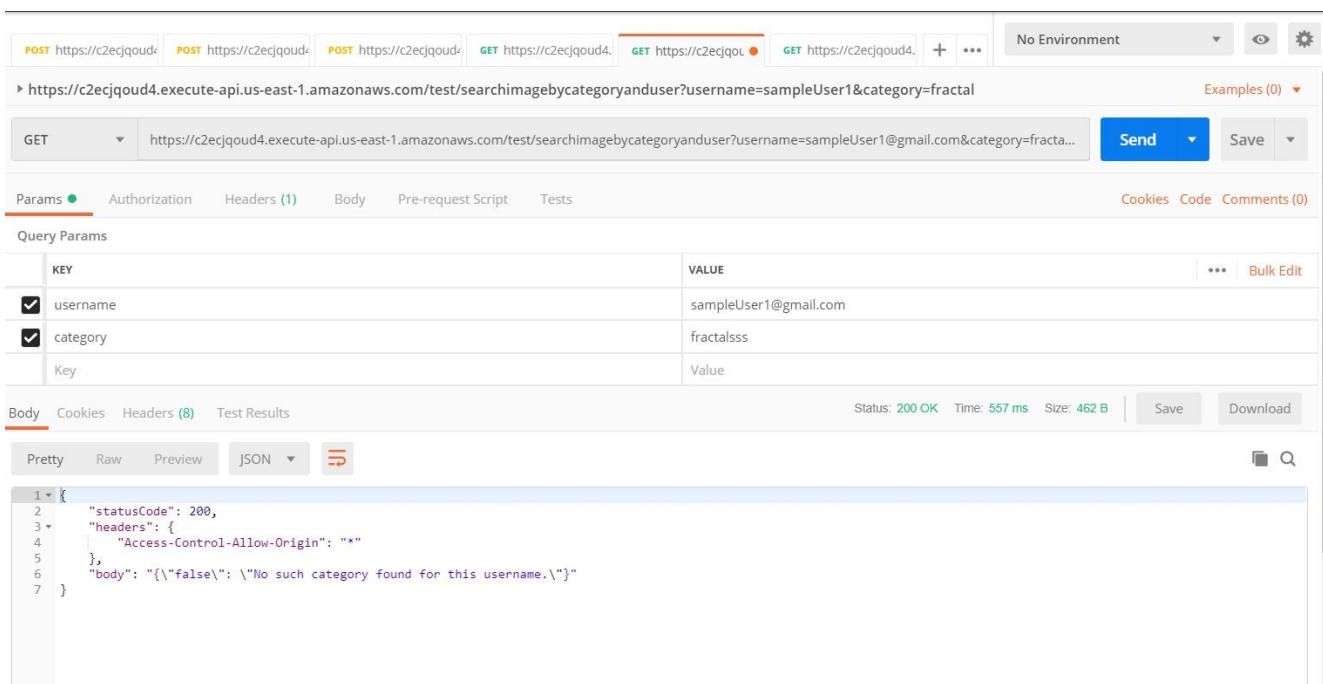
KEY	VALUE	...	Bulk Edit
<input checked="" type="checkbox"/> username	sampleUser12@gmail.com		
<input checked="" type="checkbox"/> category	fractal		
Key	Value		

Body Cookies Headers (8) Test Results Status: 200 OK Time: 1106 ms Size: 439 B Save Download

Pretty Raw Preview JSON

```
1 {
2   "statusCode": 200,
3   "headers": {
4     "Access-Control-Allow-Origin": "*"
5   },
6   "body": "{\"false\": \"No username found.\"}"
7 }
```

Screenshot of when category not found for a username specified:

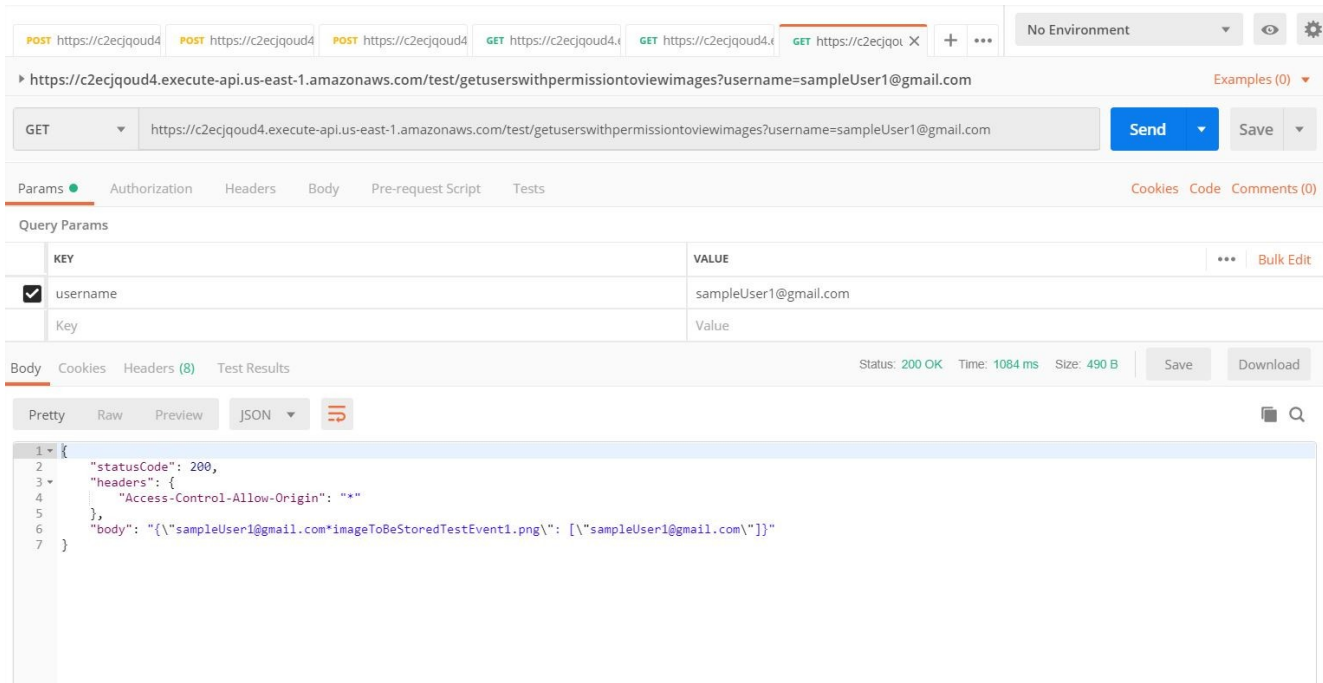


GET:

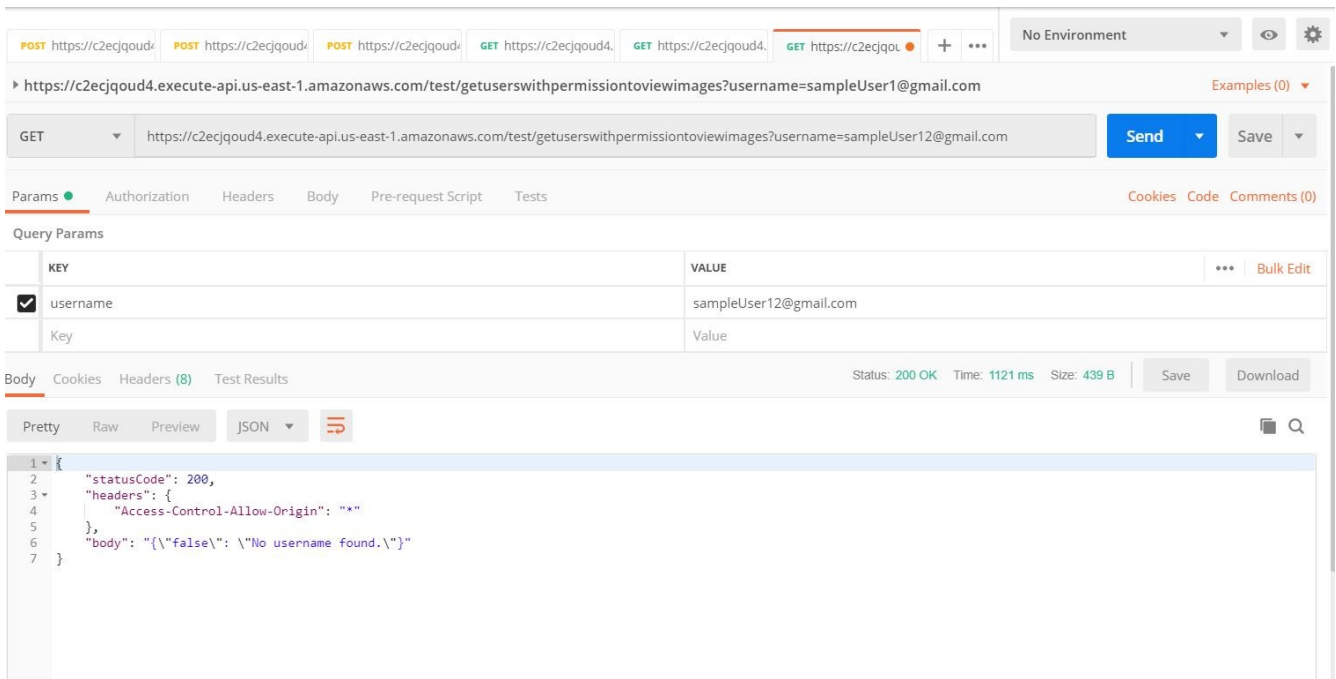
<https://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/getuserswithpermissiontoviewimages?username=sampleUser1@gmail.com>

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:



Screenshot of when username not found:

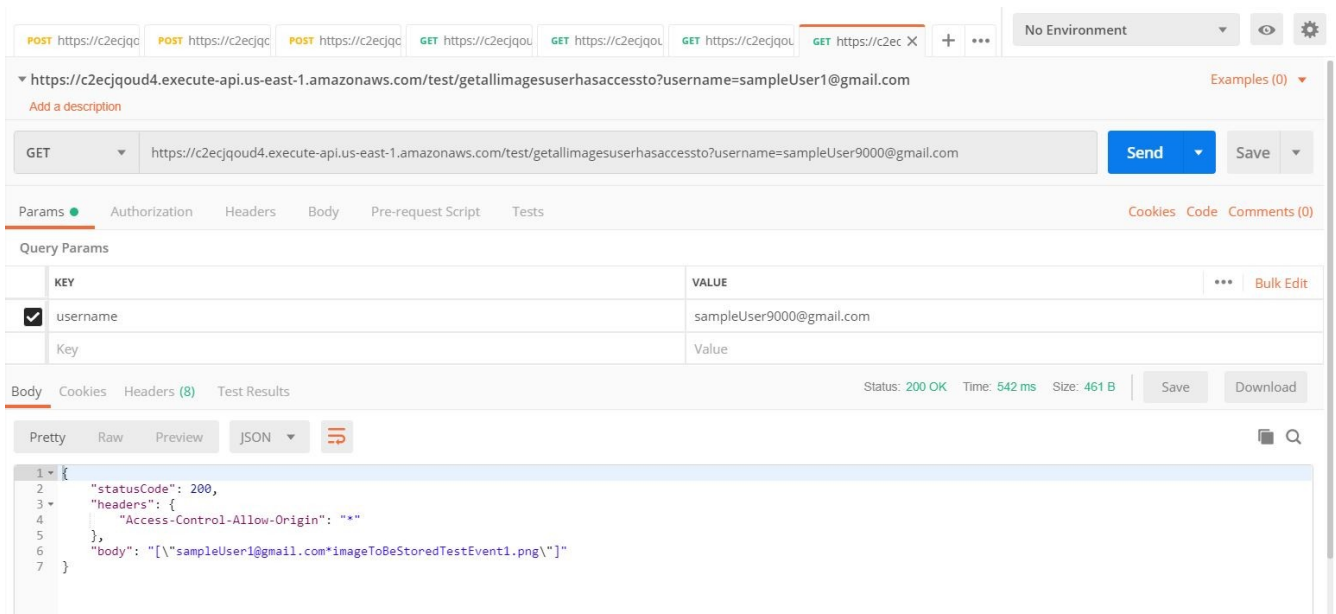


POST:

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

Screenshot:



Screenshot of when username cannot be found in database (list returned is empty):

POST https://c2ecjqcPOST https://c2ecjqcPOST https://c2ecjqcGET https://c2ecjqouGET https://c2ecjqouGET https://c2ecjqouGET https://c2ecjq

No Environment

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

Add a description

GEThttps://c2ecjqoud4.execute-api.us-east-1.amazonaws.com/test/getallimagesuserhasaccessto?username=sampleUserNotValid@gmail.comSendSave

ParamsAuthorizationHeadersBodyPre-request ScriptTestsCookiesCodeComments (0)

Query Params

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

BodyCookiesHeaders (8)Test ResultsStatus: 200 OKTime: 417 msSize: 405 BSaveDownload

PrettyRawPreviewJSON

```
1 {
2   "statusCode": 200,
3   "headers": {
4     "Access-Control-Allow-Origin": "*"
5   },
6   "body": "[]"
7 }
```