

File Edit Selection View Go Run Terminal Help • ibmm.html - New folder - Visual Studio Code

EXPLORER

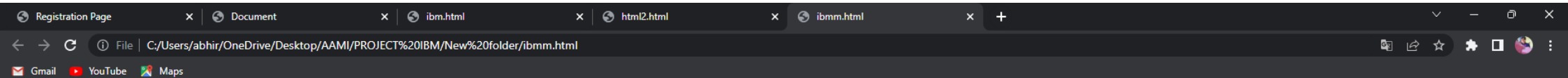
- NEW FOLDER
 - New folder
 - ABHIRAMI.html
 - ALFANIC.html
 - html.html
 - html2.html
 - ibm.html
 - ibmm.html

ibmm.html > endpoint > api-key > resource-instance-id

```
31     }).promise();
32   }
33   function doDeleteBucket() {
34     console.log('Deleting bucket');
35     return cos.deleteBucket({
36       Bucket: 'my-bucket'
37     }).promise();
38   }
39   doCreateBucket()
40     .then(doCreateObject)
41     .then(doDeleteObject)
42     .then(doDeleteBucket)
43     .then(function() {
44       console.log('Finished!');
45     })
46     .catch(function(err) {
47       console.error('An error occurred:');
48       console.error(util.inspect(err));
49     });
50
51   import boto3
52   endpoint = 'https://s3-api.us-geo.objectstorage.softlayer.net'
53   s3 = boto3.resource('s3', endpoint_url=endpoint)
54   for bucket in s3.buckets.all():
55     print(bucket.name)
56   for obj in bucket.objects.all():
57     print(" - %s" % obj.key)
58
59
60   import boto3
61   import pprint as pp
62   endpoint = 'https://s3-api.us-geo.objectstorage.softlayer.net'
63   s3 = boto3.client('s3', endpoint_url=endpoint)
64   print('These are the buckets in this service account:')
65   buckets = s3.list_buckets()
66   pp.pprint(buckets, width=180)
67   for bucket in buckets['Buckets']:
68     name = bucket['name']
69     print("Raw output from 'list_buckets()' in %s:" % name)
70     objects = s3.list_objects(Bucket=name)
71     pp.print(objects)
72
73   import boto3
74   import pprint as pp
75   endpoint = 'https://s3-api.us-geo.objectstorage.softlayer.net'
76   s3 = boto3.client('s3', endpoint_url=endpoint)
77   bucket = s3.create_bucket(Bucket=samepl,
78   CreateBucketConfiguration={
79     LocationConstraint: 'us-cold'})
```

0 0 0 Preview Available

Ln 80, Col 1 Spaces: 4 UTF-8 CRLF HTML 14:32 20-10-2022



```
var ibm = require('ibm-cos-sdk'); var util = require('util'); var config = { endpoint: "", apiKeyId: "", serviceInstanceId: "" }; var cos = new ibm.S3(config); function doCreateBucket() { console.log('Creating bucket'); return cos.createBucket({ Bucket: 'my-bucket', CreateBucketConfiguration: { LocationConstraint: 'us-standard' }, }).promise(); } function doCreateObject() { console.log('Creating object'); return cos.putObject({ Bucket: 'my-bucket', Key: 'foo', Body: 'bar' }).promise(); } function doDeleteObject() { console.log('Deleting object'); return cos.deleteObject({ Bucket: 'my-bucket', Key: 'foo' }).promise(); } function doDeleteBucket() { console.log('Deleting bucket'); return cos.deleteBucket({ Bucket: 'my-bucket' }).promise(); } doCreateBucket().then(doCreateObject).then(doDeleteObject).then(doDeleteBucket).then(function() { console.log('Finished!'); }).catch(function(err) { console.error('An error occurred:'); console.error(util.inspect(err)); }); import boto3 endpoint = 'https://s3-api.us-gio.objectstorage.softlayer.net' s3 = boto3.resource('s3', endpoint_url=endpoint) for bucket in s3.buckets.all(): print(bucket.name) for obj in bucket.objects.all(): print(" - %s" % obj.key) import boto3 import pprint as pp endpoint = 'https://s3-api.us-gio.objectstorage.softlayer.net' s3 = boto3.client('s3', endpoint_url=endpoint) print("These are the buckets in this service account:") buckets = s3.list_buckets() pp.pprint(buckets, width=180) for bucket in buckets['Buckets']: name = bucket['name'] print("Raw output from 'list_buckets()' in %s:" % name) objects = s3.list_objects(Bucket=name) pp.print(objects) import boto3 import pprint as pp endpoint = 'https://s3-api.us-gio.objectstorage.softlayer.net' s3 = boto3.client('s3', endpoint_url=endpoint) bucket = s3.create_bucket(Bucket='samepl', CreateBucketConfiguration={ 'LocationConstraint': 'us-cold' })
```



File Edit Selection View Go Run Terminal Help • ibmm.html - New folder - Visual Studio Code

EXPLORER

- NEW FOLDER
 - New folder
 - ABHIRAMI.html
 - ALFANIC.html
 - html.html
 - html2.html
 - ibm.html
 - ibmm.html

ibmm.html > endpoint > api-key > resource-instance-id

```
1 var ibm = require('ibm-cos-sdk');
2 var util = require('util');
3 var config = {
4   endpoint: '<endpoint>',
5   apiKeyId: '<api-key>',
6   serviceInstanceId: '<resource-instance-id>',
7 };
8 var cos = new ibm.S3(config);
9 function doCreateBucket() {
10   console.log('Creating bucket');
11   return cos.createBucket({
12     Bucket: 'my-bucket',
13     CreateBucketConfiguration: {
14       LocationConstraint: 'us-standard'
15     },
16   }).promise();
17 }
18 function doCreateObject() {
19   console.log('Creating object');
20   return cos.putObject({
21     Bucket: 'my-bucket',
22     Key: 'foo',
23     Body: 'bar'
24   }).promise();
25 }
26 function doDeleteObject() {
27   console.log('Deleting object');
28   return cos.deleteObject({
29     Bucket: 'my-bucket',
30     Key: 'foo'
31   }).promise();
32 }
33 function doDeleteBucket() {
34   console.log('Deleting bucket');
35   return cos.deleteBucket({
36     Bucket: 'my-bucket'
37   }).promise();
38 }
39 doCreateBucket()
40   .then(doCreateObject)
41   .then(doDeleteObject)
42   .then(doDeleteBucket)
43   .then(function() {
44     console.log('Finished!');
45   })
46   .catch(function(err) {
47     console.error('An error occurred:');
48     console.error(util.inspect(err));
49   });
```

Ln 80, Col 1 Spaces: 4 UTF-8 CRLF HTML

26°C Cloudy 14:32 20-10-2022