Cloud Developer Certification Preparation

Exercise 4.5:
Adding object storage services in IBM Bluemix

Exercise 4.5: Prerequisites

Sign up for a 30-day free trial IBM Bluemix account if you don't already have one.

You also need the following software:

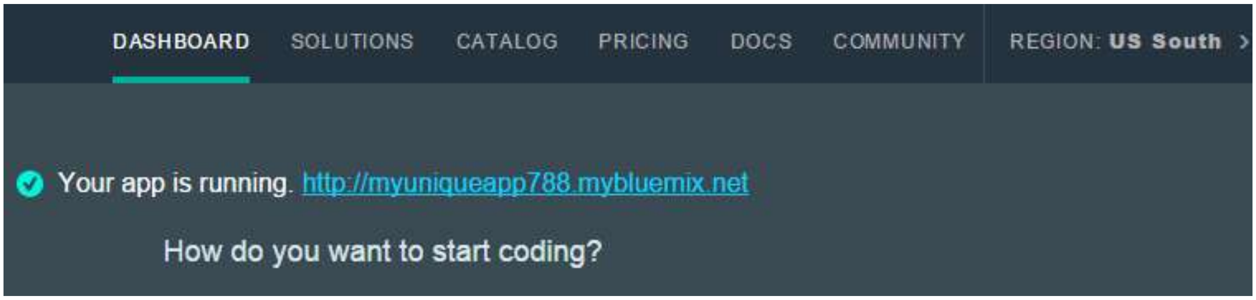
- The latest version of the Google Chrome browser with the free Postman application (for testing the Swift API in the Object Storage Service)
    - Postman is available at: https://www.getpostman.com

Exercise 4.5.1: Creating an app and an Object Storage instance in Bluemix

You will learn how to use the IBM SoftLayer Object Storage Services in IBM Bluemix as a cloud-based permanent storage repository accessible through a REST API.

In this exercise, you'll create an app and an instance of the Object Storage service in IBM Bluemix.

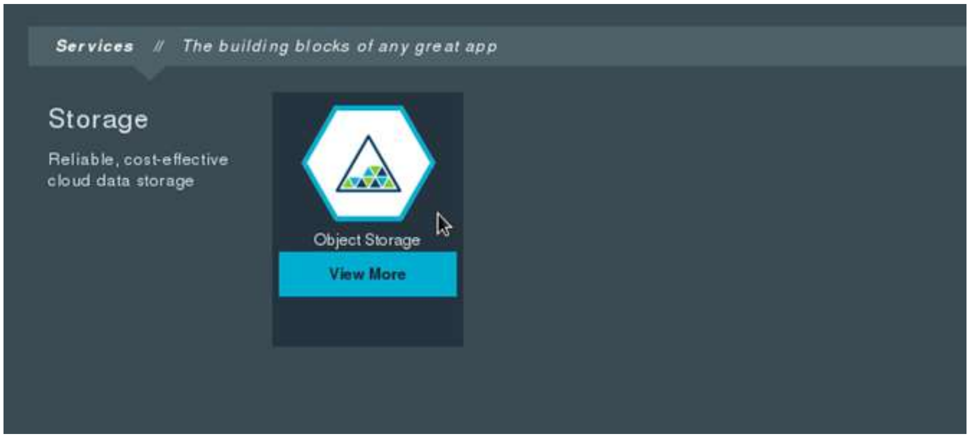
1. In your browser, log in to Bluemix: http://bluemix.net.
2. Make sure you're in the Dashboard tab. If you're not, click the Dashboard link at the top of the page.
3. Scroll down to the **Applications** section and click on **CREATE APP.**
4. Click **WEB.**
5. Click **SDK for Node.js** and then click **CONTINUE**.
6. Give your app a unique name by using alphanumeric characters only and click **FINISH.**
7. Wait for the message that says *Your app is running.*



Your app is running. http://myuniqueapp788.mybluemix.net

How do you want to start coding?

8. Click the **Dashboard** link at the top of the page again.
9. Click **USE SERVICES OR APIS**.
10. Scroll down to the Storage section and click **Object Storage**.


Services // The building blocks of any great app
Storage
Reliable, cost-effective cloud data storage
Object Storage
View More

11. Under **App,** select the app that you just created.

12. Click **CREATE** to create a new instance of Object Storage.

13. Click **RESTAGE** when prompted to restage your app.

Exercise 4.5.2: Getting the credentials required to access the Object Storage service by using REST APIs

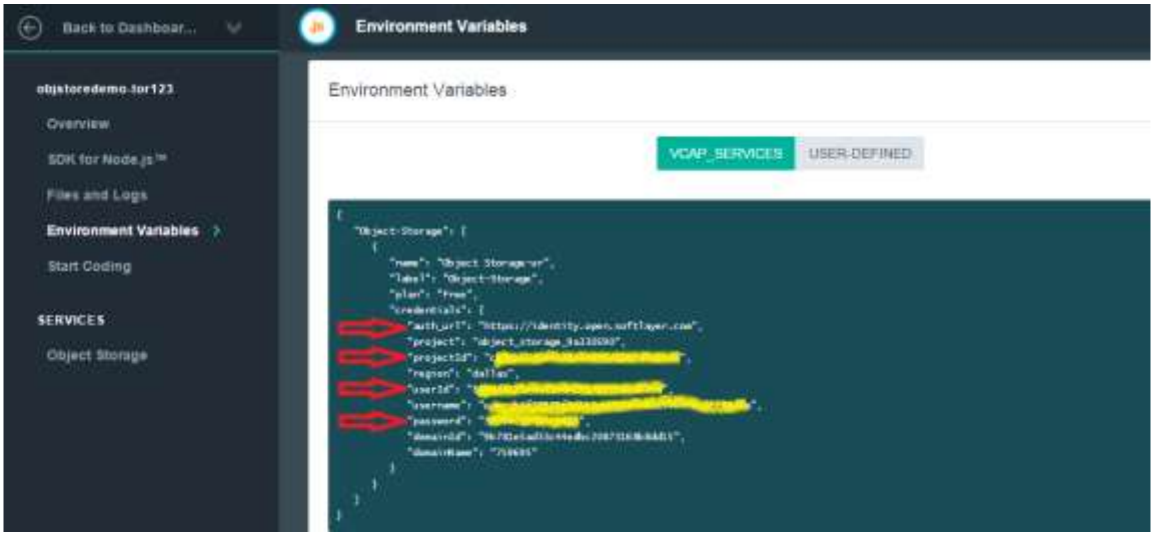
In this section you'll get the credentials required to access your Object Storage Service instance via its REST API.

1. Click the Dashboard link at the top of the page.

2. Scroll down to the **Applications** section and click the icon for the application that you just created.

3. On the left, select **Environment Variables** to show the access information for your instance of the Object Storage service.

4. Copy the values for the following fields (without the double quotation marks) in a file on your system so that you can cut and paste this information in subsequent steps: `auth_url`, `userID`, `projectId`, and `password`.



Environment Variables

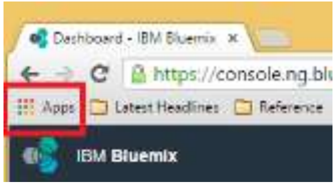
```
{
  "Object-Storage": [
    {
      "name": "Object Storage-sr",
      "label": "Object-Storage",
      "plan": "Free",
      "credentials": {
        "auth_url": "https://identity.open.softlayer.com",
        "project": "object_storage_9a33[illegible]",
        "projectId": "[redacted]",
        "region": "dallas",
        "userId": "[redacted]",
        "username": "[redacted]",
        "password": "[redacted]",
        "domainId": "[illegible]",
        "domainName": "[illegible]"
      }
    }
  ]
}
```

Exercise 4.5.3: Authenticating against your Object Storage service instance by using POSTMAN

In this section you'll use the POSTMAN Chrome application to authenticate against your Object Storage service instance.

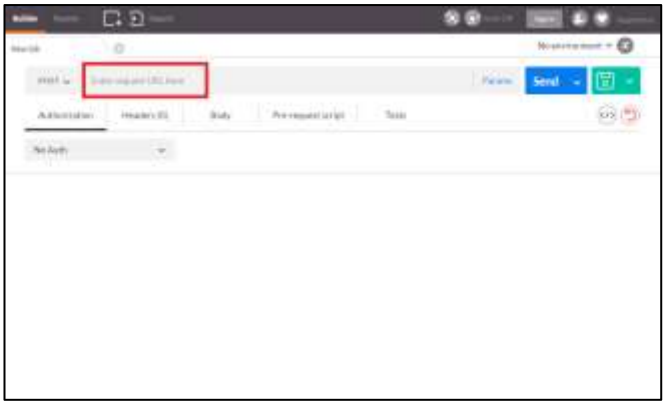
1. If it's not already running, start Chrome.

2. Click the **Apps** icon.



3. Click the icon to launch POSTMAN.

4. Select POST as the method and from the file where you saved the access credentials and copy the `auth_url` to the **Enter request URL here** field.

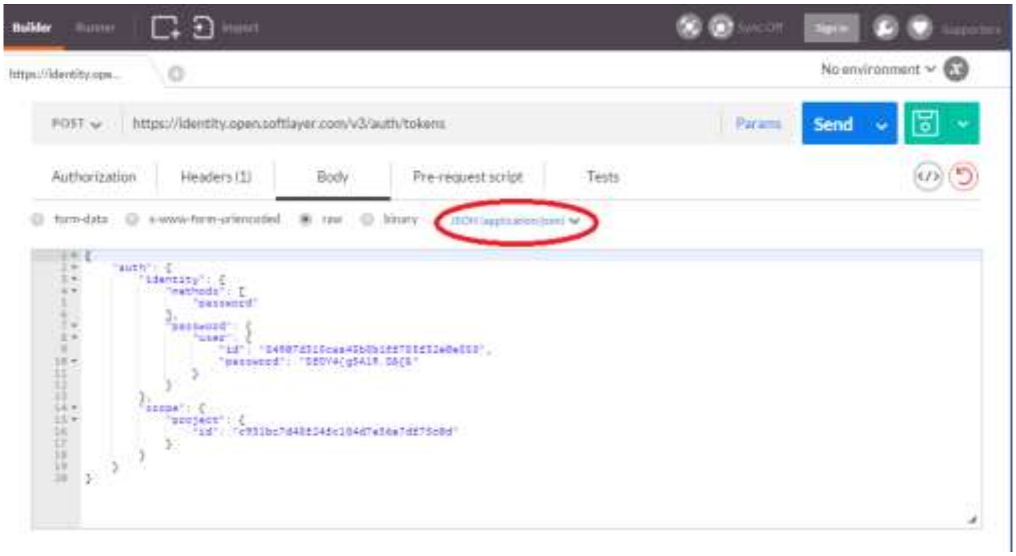


5. Append a forward slash (`/`) to the URL that you just pasted. Then, append `v3/auth/tokens` to the URL making the full url: `auth_url/v3/auth/tokens`

6. Under **Body**, select **raw**, then set type to **JSON(application/json)** and enter the following, replacing with your `userID`, `projectID` and `password:`

```
{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "id": "your userID",
                 "password": "your password"
```

```
                    }
                }
            },
            "scope": {
                "project": {
                    "id": "your projectID"
                }
            }
        }
}
```

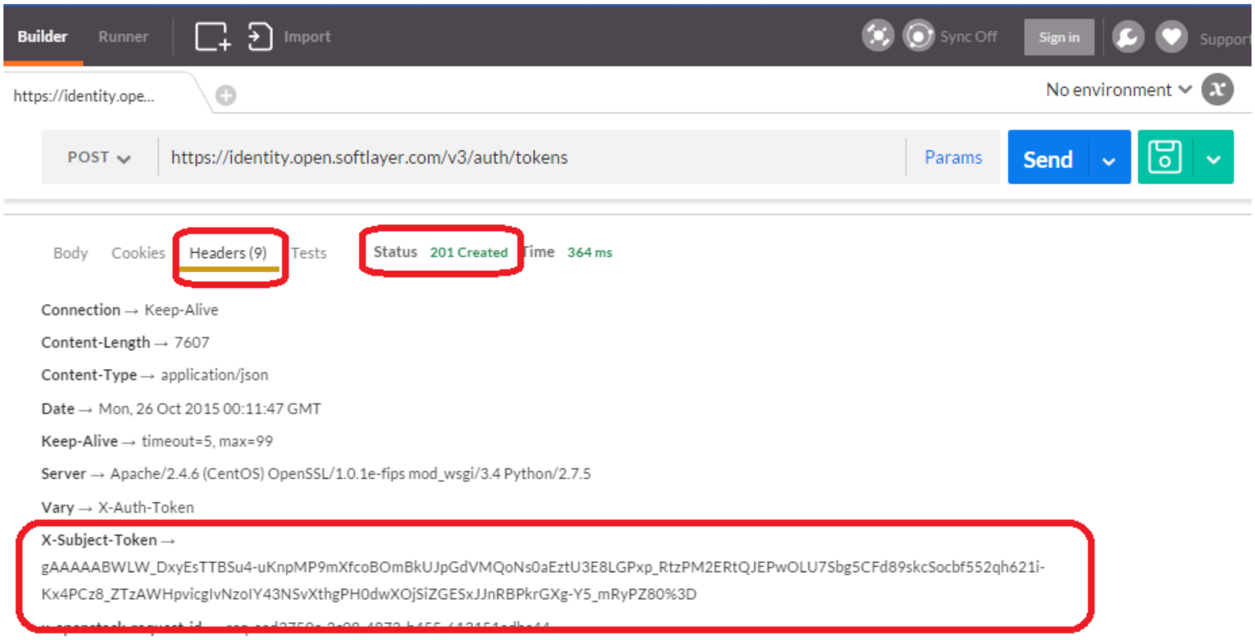


POST https://identity.open.softlayer.com/v3/auth/tokens

```
{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "id": "54807d310caa45b0b1ff70fdf2e0e053",
                    "password": "S8DY4(g5A1#.D6{$"
                }
            }
        },
        "scope": {
            "project": {
                "id": "c931bc7d48524fc184d7e56a7df75c8d"
            }
        }
    }
}
```

7. Click **Send** to send the request.

8. Verify that the status indicates **201 Created** and then click **Headers.**

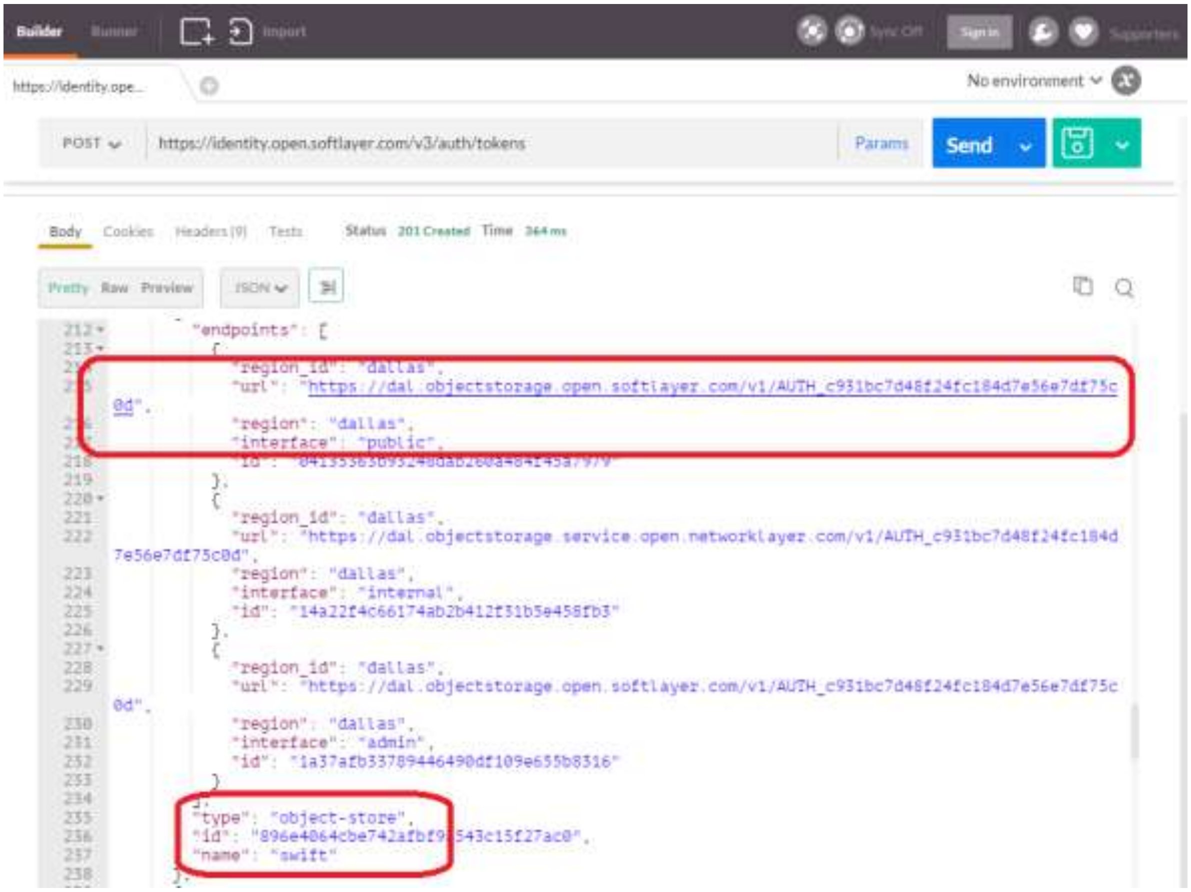


https://identity.open.softlayer.com/v3/auth/tokens

Status 201 Created Time 364 ms

```
Connection → Keep-Alive
Content-Length → 7607
Content-Type → application/json
Date → Mon, 26 Oct 2015 00:11:47 GMT
Keep-Alive → timeout=5, max=99
Server → Apache/2.4.6 (CentOS) OpenSSL/1.0.1e-fips mod_wsgi/3.4 Python/2.7.5
Vary → X-Auth-Token
X-Subject-Token →
gAAAAABWLW_DxyEsTTBSu4-uKnpMP9mXfcoBOmBkUJpGdVMQoNs0aEztU3E8LGPxp_RtzPM2ERtQJEPwOLU7Sbg5CFd89skcSocbf552qh621i-
Kx4PCz8_ZTzAWHpvicgIvNzoIY43NSvXthgPH0dwXOjSiZGESxJJnRBPkrGXg-Y5_mRyPZ80%3D
```

9. Save the value of `X-Subject-Token` from the returned header in your credentials file. Then, go back to the Body of the response and scroll down through the endpoints listed to find the one with type “object-store” and name “swift.” Copy the `url` for the entry matching the “public” interface and also save this to your credentials file.



POST https://identity.open.softlayer.com/v3/auth/tokens

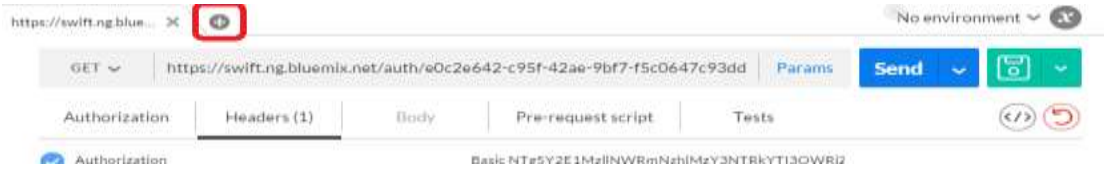
Status 201 Created Time 364 ms

```
        "endpoints": [
          {
            "region_id": "dallas",
            "url": "https://dal.objectstorage.open.softlayer.com/v1/AUTH_c931bc7d48f24fc184d7e56e7df75c
0d",
            "region": "dallas",
            "interface": "public",
            "id": "04135363b093248dab260a484f45a7979"
          },
          {
            "region_id": "dallas",
            "url": "https://dal.objectstorage.service.open.networklayer.com/v1/AUTH_c931bc7d48f24fc184d
7e56e7df75c0d",
            "region": "dallas",
            "interface": "internal",
            "id": "14a22f4c66174ab2b412f31b5e458fb3"
          },
          {
            "region_id": "dallas",
            "url": "https://dal.objectstorage.open.softlayer.com/v1/AUTH_c931bc7d48f24fc184d7e56e7df75c
0d",
            "region": "dallas",
            "interface": "admin",
            "id": "1a37afb33789446490df109e655b8316"
          }
        ],
        "type": "object-store",
        "id": "896e4064cbe742afbf9 543c15f27ac0",
        "name": "swift"
      },
```

Exercise 4.5.4: Using the Swift API to manage containers and objects

In this section you'll run through some typical Swift API operations by using the POSTMAN Chrome application.

1. In POSTMAN, click the **+** icon to open a new tab.



GET https://swift.ng.bluemix.net/auth/e0c2e642-c95f-42ae-9bf7-f5c0647c93dd

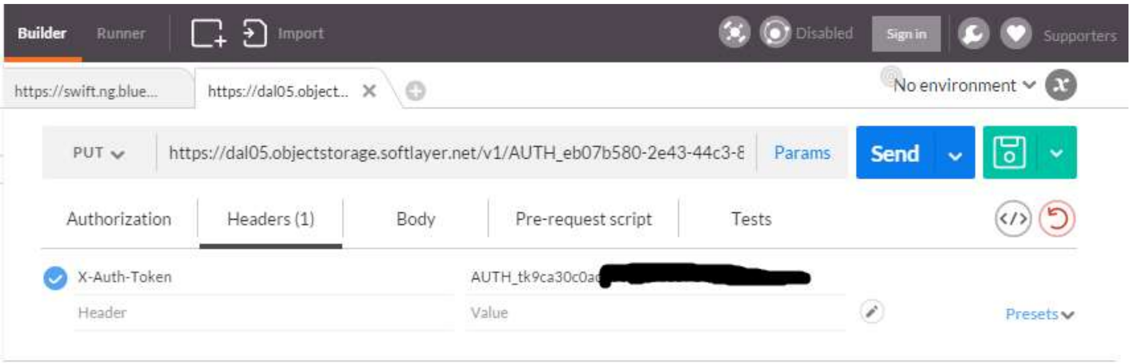
Authorization: Basic NTe5Y2E1MzllNWRmNzhlMzY3NTRkYTI3OWRi2

2. Copy the `url` value that you saved in the previous section into the **URL** field.

3. Append `/mycontainer` to the URL.

4. Change the HTTP verb from **GET** to **PUT** to create a new container.

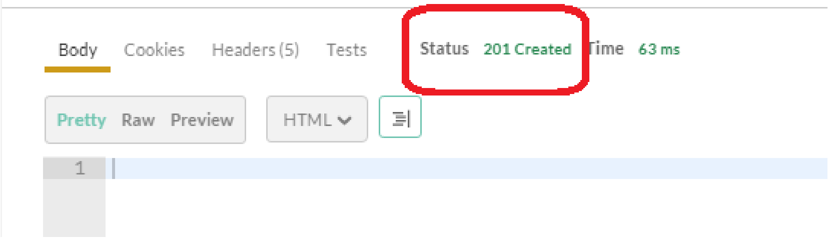
5. Click **Headers(0)** and create a header with the label **X-Auth-Token** and value set to the value of the `X-Subject-Token` that you saved in the previous section.



PUT https://dal05.objectstorage.softlayer.net/v1/AUTH_eb07b580-2e43-44c3-8

X-Auth-Token: AUTH_tk9ca30c0a[redacted]

6. Click **Send** to verify that the returned Status is **201 Created.**

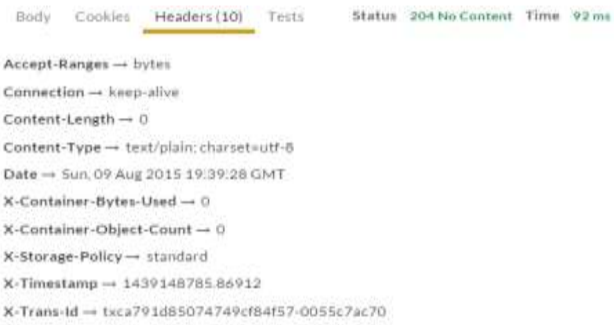


Status 201 Created Time 63 ms

7. Change the HTTP verb from **PUT** to **HEAD** to get the metadata for the new

container. Then, click **Send.** Remember that you're reusing the same header value from the previous request.

8. Verify that returned Status is **204 No Content** and that several metadata values are in the returned HTTP header

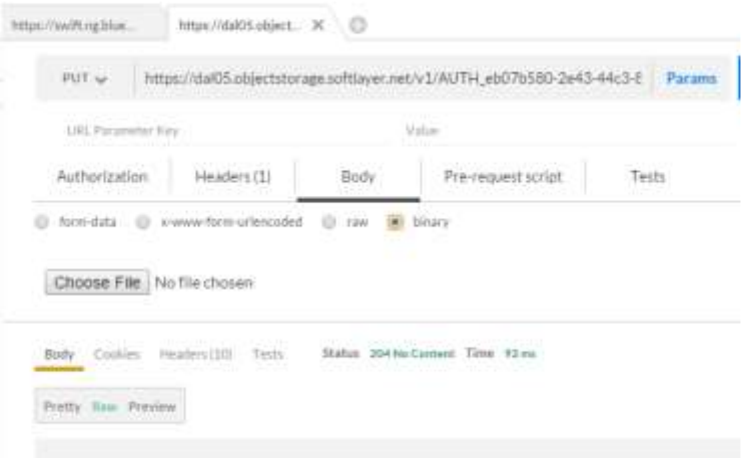


Status 204 No Content Time 92 ms

```
Accept-Ranges → bytes
Connection → keep-alive
Content-Length → 0
Content-Type → text/plain; charset=utf-8
Date → Sun, 09 Aug 2015 19:39:28 GMT
X-Container-Bytes-Used → 0
X-Container-Object-Count → 0
X-Storage-Policy → standard
X-Timestamp → 1439148785.86912
X-Trans-Id → txca791d85074749cf84f57-0055c7ac70
```

9. Change the HTTP verb from **HEAD** to **PUT** to do an upload operation.

10. Click the **Body** tab and then select **binary**.



PUT https://dal05.objectstorage.softlayer.net/v1/AUTH_eb07b580-2e43-44c3-8

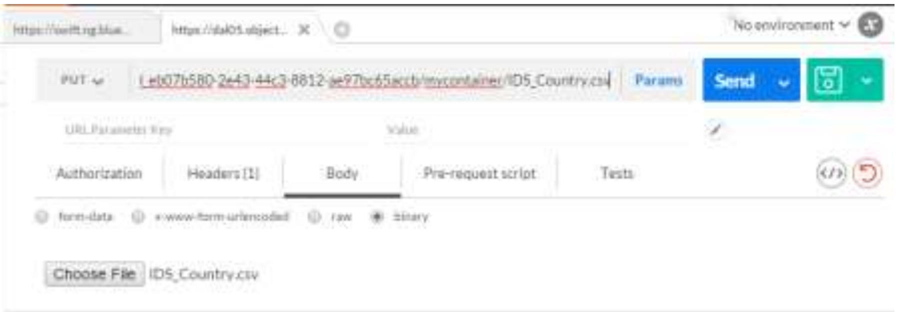
○ form-data ○ x-www-form-urlencoded ○ raw ● binary

No file chosen

Status 204 No Content Time 92 ms

11. Click **Choose File** and select a file on your local system.

12. Go back to the URL field and append `/filename` to the URL where `filename` is the name of the file that you selected. For example if you selected `foo.txt`, then you append `/foo.txt` to the URL.



PUT |_eb07b580-2e43-44c3-8812-ae97bc65accb/mycontainer/ID5_Country.csv

○ form-data ○ x-www-form-urlencoded ○ raw ● binary

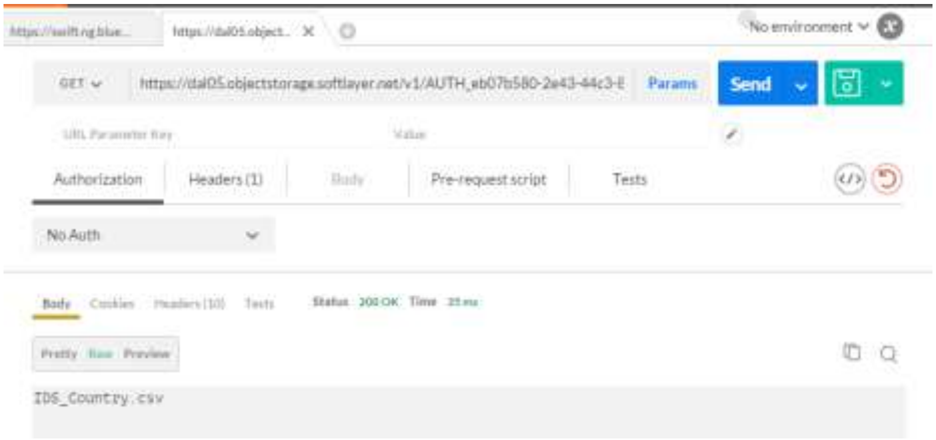
ID5_Country.csv

13. Click **Send** and verify that **Status 201 Created** is returned.

14. Change the HTTP verb from **PUT** to **GET** so that you can send a list request for the container.

15. Remove the file name suffix from the URL.

16. Click **Send** and verify that the name of the file that you just uploaded is returned.



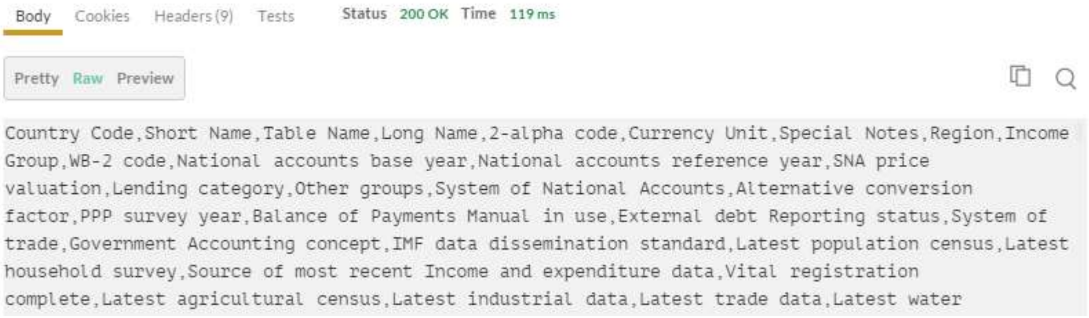
```
https://dal05.objectstorage.softlayer.net/v1/AUTH_eb07b580-2e43-44c3-E
```

```
IDS_Country.csv
```

17. Append `?format=json` to the URL and send the **GET** request again. Verify that information about the file that you uploaded is returned in JSON format.

18. Remove the `?format=json` suffix from the URL and append `/filename` to the URL where `filename` is the name of the file that you uploaded earlier.

19. Click **Send** and verify that the content of the file is returned and that the status is **200 OK.**



```
Country Code,Short Name,Table Name,Long Name,2-alpha code,Currency Unit,Special Notes,Region,Income
Group,WB-2 code,National accounts base year,National accounts reference year,SNA price
valuation,Lending category,Other groups,System of National Accounts,Alternative conversion
factor,PPP survey year,Balance of Payments Manual in use,External debt Reporting status,System of
trade,Government Accounting concept,IMF data dissemination standard,Latest population census,Latest
household survey,Source of most recent Income and expenditure data,Vital registration
complete,Latest agricultural census,Latest industrial data,Latest trade data,Latest water
```

20. Change the HTTP verb from **GET** to **DELETE** to delete the file that you uploaded earlier.

21. Click **Send** and verify that **Status 204 No content** is returned.

22. Change the HTTP verb from **DELETE** to **GET** so that you can send a list request for the container.

23. Remove the file name suffix from the URL.
24. Click **Send** and verify that **Status 204 No content** is returned. Note that there is no content because the container has no files.

You successfully created an instance of the Object Storage service in Bluemix and executed the most common REST API calls.