

# Using HTTP API to create and query Cloudant databases



Estimated time needed: **30 minutes**

## Objectives

After completing this lab you will be able to use HTTP API with curl, the command-line tool, to:

- Create, query and drop databases
- Perform basic commands such as inserting, updating and deleting documents
- Create an index to optimize query time

## Prerequisites

In order to complete this lab, you will need to create an instance of Cloudant on IBM Cloud. If you haven't yet created one, you can create one by referring to the [Create an Instance of IBM Cloudant](#) lab.

Before starting this lab, it'll also be helpful to have an understanding of basic operations with Cloudant. If you're unfamiliar with Cloudant, feel free to take a look at the [Working with Databases in Cloudant](#) lab!

## About This SN Labs Cloud IDE

This Skills Network Labs Cloud IDE provides a hands-on environment for course and project related labs. It utilizes Theia, an open-source IDE (Integrated Development Environment) platform, that can be run on desktop or on the cloud. To complete this lab, we will be using the Cloud IDE based on Theia and Cassandra running in a Docker container.

## Important Notice about this lab environment

Please be aware that sessions for this lab environment are not persisted. Every time you connect to this lab, a new environment is created for you. Any data you may have saved in the earlier session would get lost. Plan to complete these labs in a single session, to avoid losing your data.

Note: While working on this lab, you may be prompted to login when ever your session expires. Use your credentials to authenticate. This may happen when you step out or leave your Cloudant session unattended.

## Exercise 1 - Getting the environment ready

Before you start working with the HTTP API to access the Cloudant, you need to collect the below details of your Cloudant instance.

1. Your Cloudant username
2. Your Cloudant password
3. Your Cloudant url

You can get all these details from the Service Credentials.

► [Click here for Hint](#)

Replace the value of CLOUDANTURL given below with yours.

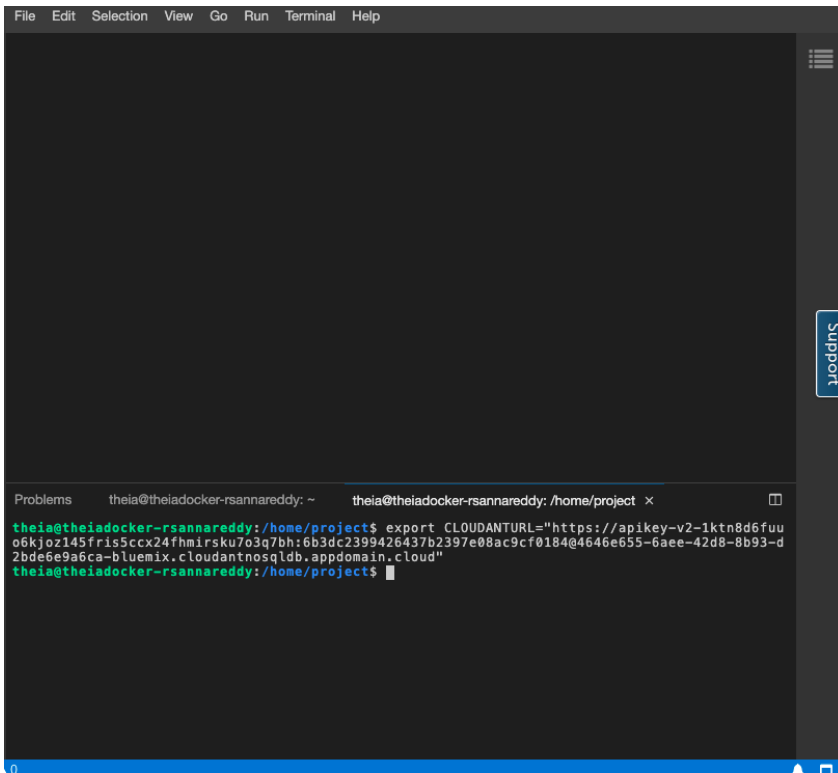
DO NOT use the values given below. They are all expired credentials.

After replacing with your credentials, copy and paste the below command on the terminal.

This url contains your Cloudant username and password. DO NOT MAKE IT PUBLIC OR SHARE IT WITH ANYONE.

1. 1
2. 2
- 1.
2. `export CLOUDANTURL="https://apikey-v2-1ktn8d6fuuo6kjoz145fr1s5ccx24fhm1rsku7o3q7bh:6b3dc2399426437b2397e08ac9cf0184@4646e655-6aee-42d8-8b93-d2bde6e9a6ca-bluemix.cloudantnosqldb.appdomain.cloud"`

Copied!



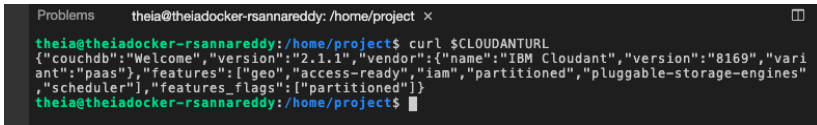
Test if you can reach your Cloudant server from the lab environment.

Run the below command on the terminal.

- 1
1. curl \$CLOUDANTURL

Copied!

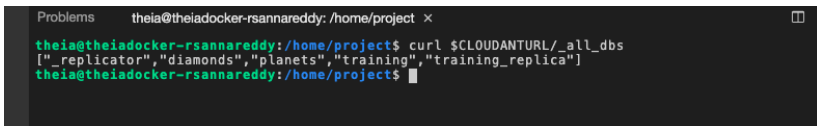
If you get an output similar to the one below, it means you are able to reach your Cloudant server.



Test your credentials.

- 1
1. curl \$CLOUDANTURL/\_all\_dbs

Copied!



If you get a list of your databases, it indicates that your credentials are correct. Now we can proceed with the rest of the lab.

## Exercise 2 - Create a database

Run the below command on the terminal to create a database named animals.

- 1
- 2
- 1.
2. curl -X PUT \$CLOUDANTURL/animals

Copied!

A response of {"ok":true} indicates that the database is successfully created.

Verify by listing all databases.

- 1
- 2
- 1.
2. curl \$CLOUDANTURL/\_all\_dbs

Copied!

## Exercise 3 - Drop a database

Run the below command to delete the database named animals.

```
1. 1
2. 2

1.
2. curl -X DELETE $CLOUDANTURL/animals
```

Copied!

A response of {"ok":true} indicates that the database is successfully deleted.

Verify that the database is deleted by listing all databases.

```
1. 1
2. 2

1.
2. curl $CLOUDANTURL/_all_dbs
```

Copied!

## Exercise 4 - Insert a document

*Create a database named planets.*

- [Click here for Hint](#)
- [Click here for Solution](#)

Run the below command to insert a document in the planets database with \_id of "1".

```
1. 1
2. 2
3. 3
4. 4
5. 5

1.
2. curl -X PUT $CLOUDANTURL/planets/"1" -d '{
3.   "name" : "Mercury",
4.   "position_from_sun" :1
5.   }'
```

Copied!

If you see {"ok":true} in the response, it means the document is successfully inserted.

Note: You can also create a document using POST. When you use POST to create a document, you do not have to mention the id. Cloudant will use an auto generated id for the document.

Verify by listing the document with the \_id "1".

```
1. 1
2. 2

1.
2. curl -X GET $CLOUDANTURL/planets/1
```

Copied!

Make a note of the value for the '\_rev' key. You will need it later.

## Exercise 5 - Update a document

To update a document you need its revision id.

Replace the "\_rev" below with the one you noted earlier.

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7

1.
2. curl -X PUT $CLOUDANTURL/planets/1 -d '{
3.   "name" : "Mercury",
4.   "position_from_sun" :1,
5.   "revolution_time":"88 days",
6.   "_rev":"1-3fb3ccfe80573e1ae334f0cfa7304f6c"
7.   }'
```

Copied!

Verify by listing the document with the \_id "1".

```
1. 1
2. 2

1.
2. curl -X GET $CLOUDANTURL/planets/1
```

Copied!

Make a note of the changed value for the '\_rev' key. You will need it later.

Let us add another key "rotation\_time" with a value of "59 days" to the existing document.

Run the below command to update the document in the planets database with \_id of "1".

REPLACE the value of "\_rev" with what you have copied in the previous step.

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8

1.
2. curl -X PUT $CLOUDANTURL/planets/1 -d '{
3.   "name": "Mercury",
4.   "position_from_sun": 1,
5.   "revolution_time": "88 days",
6.   "rotation_time": "59 days",
```

```
7.   "_rev": "1-3fb3ccfe80573e1ae334f0cfa7304f6c"
8. },
```

Copied!

If you see {"ok":true} in the response, it means the document is successfully updated.

Verify by listing the document with the \_id "1".

```
1. 1
2. 2

1.
2. curl -X GET $CLOUDANTURL/planets/1
```

Copied!

Make a note of the '\_rev' key. You will need it in the next step.

## Exercise 6 - Delete a document

To delete a document you need its revision id.

REPLACE the value of "\_rev" with what you have noted earlier and run the command.

```
1. 1
2. 2

1.
2. curl -X DELETE $CLOUDANTURL/planets/1?rev=2-de9fdd2d971e377c5db2d6425cb38ff1
```

Copied!

If you see {"ok":true} in the response, it means the document is successfully deleted.

Verify by listing the document with the \_id "1".

```
1. 1
2. 2

1.
2. curl -X GET $CLOUDANTURL/planets/1
```

Copied!

## Exercise 7 - Query a database

For this exercise, we'll be using a database called **diamonds**. If you don't have this database, you can create and populate it with the following code:

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12
13. 13
14. 14
15. 15
16. 16
17. 17
18. 18
19. 19
20. 20
21. 21
22. 22
23. 23
24. 24
25. 25
26. 26
27. 27
28. 28
29. 29
30. 30
31. 31
32. 32
33. 33
34. 34
35. 35
36. 36
37. 37
38. 38
39. 39
40. 40
41. 41
42. 42
43. 43
44. 44
45. 45
46. 46
47. 47
48. 48
49. 49
50. 50
51. 51
52. 52
53. 53
54. 54
55. 55
56. 56
57. 57
58. 58
59. 59
60. 60
61. 61
62. 62
63. 63
64. 64
65. 65
66. 66
67. 67
68. 68
69. 69
```

70. 70  
71. 71  
72. 72  
73. 73  
74. 74  
75. 75  
76. 76  
77. 77  
78. 78  
79. 79  
80. 80  
81. 81  
82. 82  
83. 83  
84. 84  
85. 85  
86. 86  
87. 87  
88. 88  
89. 89  
90. 90  
91. 91  
92. 92  
93. 93  
94. 94  
95. 95  
96. 96  
97. 97  
98. 98  
99. 99  
100. 100  
101. 101  
102. 102  
103. 103  
104. 104  
105. 105  
106. 106  
107. 107

```
1. curl -X DELETE $CLOUDANTURL/diamonds
2. curl -X PUT $CLOUDANTURL/diamonds
3.
4. curl -X PUT $CLOUDANTURL/diamonds/1 -d '{
5.     "carat": 0.31,
6.     "cut": "Ideal",
7.     "color": "J",
8.     "clarity": "SI2",
9.     "depth": 62.2,
10.    "table": 54,
11.    "price": 339
12.  }'
13.
14. curl -X PUT $CLOUDANTURL/diamonds/2 -d '{
15.     "carat": 0.2,
16.     "cut": "Premium",
17.     "color": "E",
18.     "clarity": "SI2",
19.     "depth": 60.2,
20.     "table": 62,
21.     "price": 351
22.  }'
23.
24.
25. curl -X PUT $CLOUDANTURL/diamonds/3 -d '{
26.     "carat": 0.32,
27.     "cut": "Premium",
28.     "color": "E",
29.     "clarity": "I1",
30.     "depth": 60.9,
31.     "table": 58,
32.     "price": 342
33.  }'
34.
35.
36.
37. curl -X PUT $CLOUDANTURL/diamonds/4 -d '{
38.     "carat": 0.3,
39.     "cut": "Good",
40.     "color": "J",
41.     "clarity": "SI1",
42.     "depth": 63.4,
43.     "table": 54,
44.     "price": 349
45.  }'
46.
47.
48. curl -X PUT $CLOUDANTURL/diamonds/5 -d '{
49.     "carat": 0.3,
50.     "cut": "Good",
51.     "color": "J",
52.     "clarity": "SI1",
53.     "depth": 63.8,
54.     "table": 56,
55.     "price": 347
56.  }'
57.
58.
59. curl -X PUT $CLOUDANTURL/diamonds/6 -d '{
60.     "carat": 0.3,
61.     "cut": "Very Good",
62.     "color": "J",
63.     "clarity": "SI1",
64.     "depth": 62.7,
65.     "table": 59,
66.     "price": 349
67.  }'
68.
69. curl -X PUT $CLOUDANTURL/diamonds/7 -d '{
70.     "carat": 0.3,
71.     "cut": "Good",
72.     "color": "I",
73.     "clarity": "SI2",
74.     "depth": 63.3,
75.     "table": 56,
76.     "price": 343
77.  }'
78.
79. curl -X PUT $CLOUDANTURL/diamonds/8 -d '{
80.     "carat": 0.23,
81.     "cut": "Very Good",
82.     "color": "E",
83.     "clarity": "VS2",
84.     "depth": 63.8,
85.     "table": 55,
86.     "price": 339
87.  }'
```

```

88.
89. curl -X PUT $CLOUDANTURL/diamonds/9 -d '{
90.     "carat": 0.23,
91.     "cut": "Very Good",
92.     "color": "H",
93.     "clarity": "VS1",
94.     "depth": 61,
95.     "table": 57,
96.     "price": 323
97. }',
98.
99. curl -X PUT $CLOUDANTURL/diamonds/10 -d '{
100.    "carat": 0.31,
101.    "cut": "Very Good",
102.    "color": "J",
103.    "clarity": "SI1",
104.    "depth": 59.4,
105.    "table": 62,
106.    "price": 346
107. }',

```

Copied!

Now you can start querying the diamond database.

Query for diamond with `_id` 1

```

1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9

1.
2. curl -X POST $CLOUDANTURL/diamonds/_find \
3. -H"Content-Type: application/json" \
4. -d' {
5.   "selector":
6.     {
7.       "_id": "1"
8.     }
9.   },

```

Copied!

Query for diamonds with carat size of 0.3

```

1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8

1.
2. curl -X POST $CLOUDANTURL/diamonds/_find \
3. -H"Content-Type: application/json" \
4. -d' { "selector":
5.       {
6.         "carat": 0.3
7.       }
8.     },

```

Copied!

Query for diamonds with price more than 345

```

1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11

1.
2. curl -X POST $CLOUDANTURL/diamonds/_find \
3. -H"Content-Type: application/json" \
4. -d' { "selector":
5.       {
6.         "price":
7.           {
8.             "$gt": 345
9.           }
10.        }
11.      },

```

Copied!

## Exercise 8 - Create an index

You would have noticed a ‘Warning’: “No matching index found, create an index to optimize query time.” in the output of previous queries. That is Cloudant telling you that, you are running a query on a field that is not indexed.

Let us create an index on the key “price”.

```

1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8

1.
2. curl -X POST $CLOUDANTURL/diamonds/_index \
3. -H"Content-Type: application/json" \
4. -d' {
5.   "index": {
6.     "fields": ["price"]

```

```
7.     }
8. },'
```

Copied!

Now let us run the earlier query and you should not see the warning, and the query also runs faster.

Query for diamonds with price more than 345

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11

1.
2. curl -X POST $CLOUDANTURL/diamonds/_find \
3. -H"Content-Type: application/json" \
4. -d'{ "selector":
5.     {
6.         "price":
7.         {
8.             "$gt":345
9.         }
10.     }
11. },'
```

Copied!

## Practice exercises

1. Problem:

*Set the price of the diamond with id 2 to 352.*

- [Click here for Hint](#)
- [Click here for Solution](#)

2. Problem:

*Create an index on the key "carat".*

- [Click here for Hint](#)
- [Click here for Solution](#)

3. Problem:

*Find all diamonds that are more than 0.3 carats.*

- [Click here for Hint](#)
- [Click here for Solution](#)

4. Problem:

*Find all diamonds that have a premium cut.*

- [Click here for Hint](#)
- [Click here for Solution](#)

5. Problem:

*Find all diamonds that are of color 'E'.*

- [Click here for Hint](#)
- [Click here for Solution](#)

## Authors

Ramesh Sannareddy

### Other Contributors

Rav Ahuja

## Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2023-01-18	0.6	Lavanya Rajalingam	Updated correct link for Creating Cloudant Instance
2021-11-01	0.5	Kathy An	Updated lab instructions
2021-04-28	0.4	Steve Ryan	Changed IBM cloud links to markdown format
2021-04-14	0.3	Steve Ryan	Review pass
2021-04-13	0.2	Ramesh Sannareddy	Made updates to the lab
2021-03-24	0.1	Ramesh Sannareddy	Created initial version of the lab

Copyright (c) 2021 IBM Corporation. All rights reserved.