Cloudant DB

Presented by:

IBM

ECOD

IBM Cloud

AGENDA

- What is Cloudant and Where Does It Fit
- Prepare for Lab Exercises
- Lab Exercise 1: Add Views
- Lab Exercise 2: Make RESTful Call to Cloudant DB using Price View
- Lab Exercise 3: Make RESTful Call to Cloudant DB using Price View *for Price Range Search with Count*
- Lab Exercise 4: Make Restful Call to Cloudant DB using Make View
- Lab Exercise 5: Make Restful Call to Cloudant DB using Make View for a Specific Model Search with Count
- Lab Exercise 6: Make Restful Call to Cloudant DB using List Function

What Is Cloudant and Where Does It Fit

Cloudant is a NoSQL Database as a Service in Bluemix Context.

IBM Cloudant is a NoSQL database and an extension of Apache Couch DB. It stores JSON documents, uses JavaScript for MapReduce indexes, and HTTP RESTful for its API.

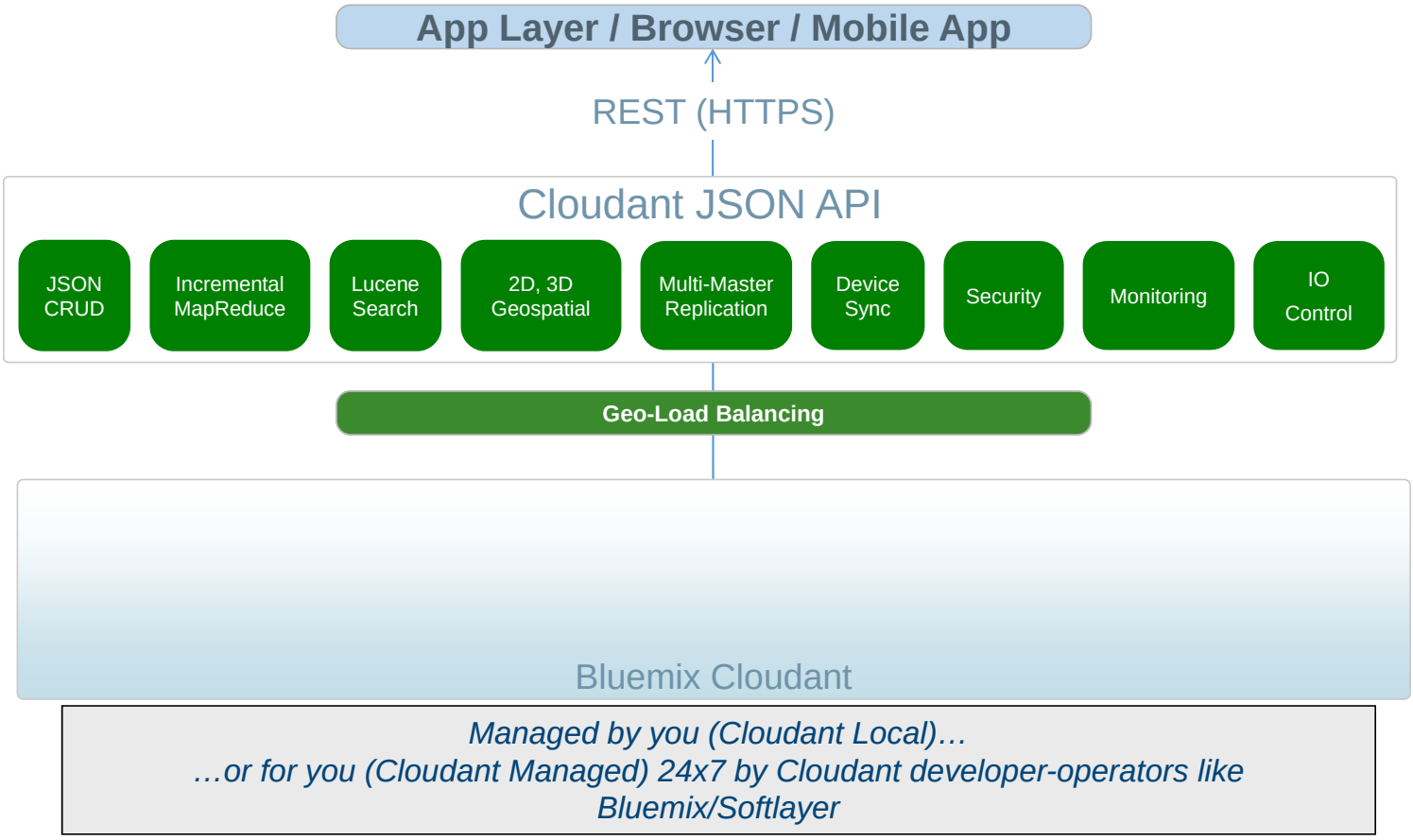
Cloudant is designed for internet in mind, optimized for handling heavy workloads of concurrent reads and writes in the cloud, flexible schema, and massive scaling for fast-growing web and mobile applications.

- Fully managed service as DaaS (Database as Service) to offload the administration and maintenance from end users.
- Massive horizontal scaling than relational DB to compensate for fast growing dataset and surge in concurrent users.
- High availability with offline & online access capability via integrated replication and synch.
- Economical by distributed data nature of the DB deployed and operated on clusters of servers in cloud architecture.
- Flexible schema and intuitive data structures enables building new features into application without locking database.
- Ability to index and query the contents of the documents, range look ups, and analytical queries via MapReduce functions.

Cloudant may not be a good fit for applications requiring transaction ACID with relationships over multiple documents.

- It's not possible for a document store to handle a transaction that operates over multiple documents.
- If you require data warehousing for batch analytics, then often a relational DB or Hadoop-based technology is better. Relational provides strong consistency and transactional rollback capabilities.

Cloudant NoSQL DBaaS



Managed by you (Cloudant Local)…
…or for you (Cloudant Managed) 24x7 by Cloudant developer-operators like Bluemix/Softlayer

Web & Mobile Cloudant Use Cases

MORNINGSTAR	Makes financial figures generated daily from Netezza accessible to millions of on-line users
HOTHEAD	Data layer for mobile gaming apps – 6 nodes to > 200 nodes in 1 year
Microsoft Studios	Powers much of the Xbox One experience
LinkedIn	On-line job posting & search
facebook	“App store” catalog for Oculus VR Rift apps
EasyBib Write Smart.	Worlds largest on-line bibliography, used by over 40 million students, researchers, publishers
Adobe	Mobile version of Adobe Premiere video editing. Mobile sync enables collaborative editing between users in real time
RunKeeper	Personal fitness mobile app with realtime mapping of running routes; a top-5 fitness app in the App Store

Enterprise Cloudant Use Cases

Fidelity Investments	Digital safety deposit box web & mobile app for 20 million Fidelity customers securely stores personal financial data
COMDATA Payment Innovation	On-vehicle geospatial app for truck drivers optimizes routes, guiding them to in-network refueling stations for lowest fuel prices
Rosetta Stone	Web & mobile version of the popular desktop language learning app, used by millions worldwide
	Cloud database for the US Intelligence community
FOOD LION	Recipes, cooking videos, and shopping list app for grocery chain customers
NOVARTIS	On-line analysis of clinical trial data. Faster indexing and order of magnitude less costly than RDBMS
MONSANTO	Bio-informatics (genomic) data analytics as a service – 10x less costly than RDBMS alternative
Akamai	Real-time threat detection within Akamai content delivery network

Cloudant or MongoDB?

- Cloudant and MongoDB are both JSON document databases
  - With: elastic scaling, declarative query syntax, text search, available on-premise or via DBaaS

Requirement	Cloudant	MongoDB	Implications
Data mobility	Mobile Sync	NA	• Cloudant enables users to access data non-stop, even when they're off line • Plus, data-on-device reduces database traffic & network latency for faster access & easier scaling
No data loss	Append-only doc update (to disk)	Update-in-place (to cache, then to disk)	• In MongoDB, data in cache, but not yet on disk can be lost if node crashes.
High availability (cross-data center)	Master-less replication & sync	Master-slave replication	• Cloudant: all replicas are readable & writable, vs. MongoDB where 1 is writeable, and the others are read-only • Spread reading/writing across all data centers & no single points of failure for easier, more economic scaling and less downtime and network latency with Cloudant
Geospatial services	Advanced geo-spatial	Limited geo-spatial	• MongoDB supports bounding box & proximity searches. • Cloudant adds: bounding polygons & ellipses, route optimization, predictive path & more

Prepare For Lab Exercises

Download lab material

Before deep diving into lab exercises, please download lab materials, setup a space in your Bluemix organization, and prepare Cloudant NoSQL DB service instance that you will create in this section.

- ❑ Copy the link, https://github.com/ibmecod/cloudantLab to an internet browser and click.

- ❑ Click Download ZIP when you are on cloudantLab repository master branch page at GitHub.

ibmecod / cloudantLab

Unwatch 6 | Star 0 | Fork 0

Repository to store the use of cloudant db in bluemix lab data and presentation — Edit

1 commit | 1 branch | 0 releases | 1 contributor

Branch: master | cloudantLab / +

initial commit

danielyhcho authored 19 minutes ago — latest commit 9647d797f1

Exercise1.txt	initial commit	19 minutes ago
Exercise2.txt	initial commit	19 minutes ago
Exercise3.txt	initial commit	19 minutes ago
Exercise4.txt	initial commit	19 minutes ago
Exercise5.txt	initial commit	19 minutes ago
Exercise6.txt	initial commit	19 minutes ago
data.json	initial commit	19 minutes ago
upload.bat	initial commit	19 minutes ago

Help people interested in this repository understand your project by adding a README. Add a README

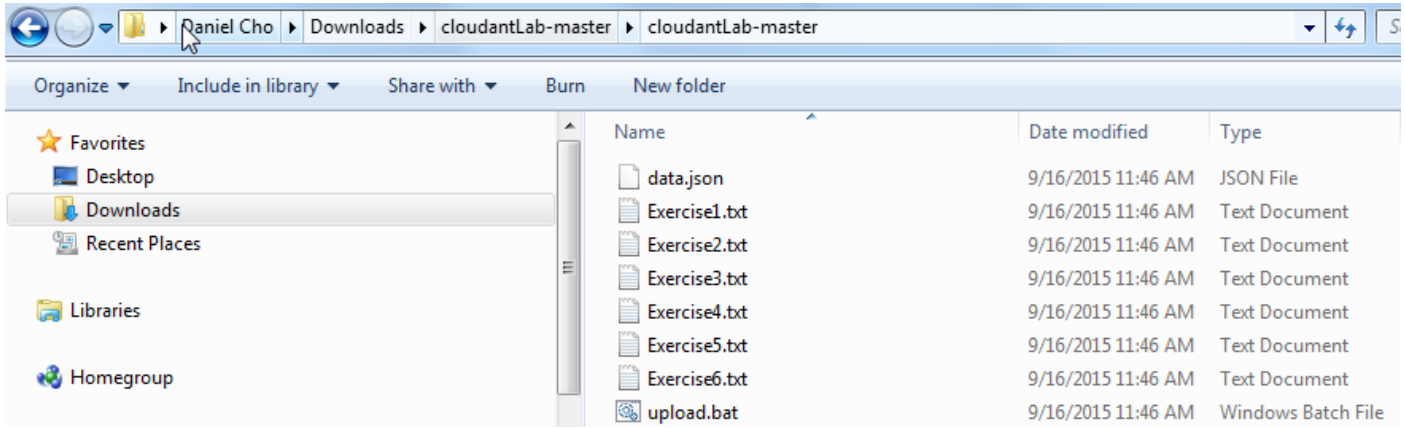
Code
Pull requests 0
Pulse
Graphs
Settings

HTTPS clone URL
https://github.com/:

You can clone with HTTPS, SSH, or Subversion.

Clone in Desktop
Download ZIP

- Unzip cloudantLB-master.zip file in your download folder.

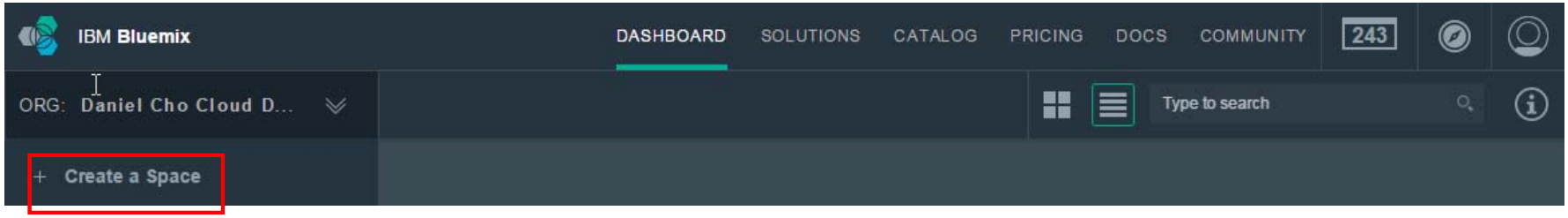


Create a space on Bluemix

For simplicity, please create a space where you will create an unbounded Cloudant NoSQL DB service instance. You will need Bluemix account created as a pre-step.

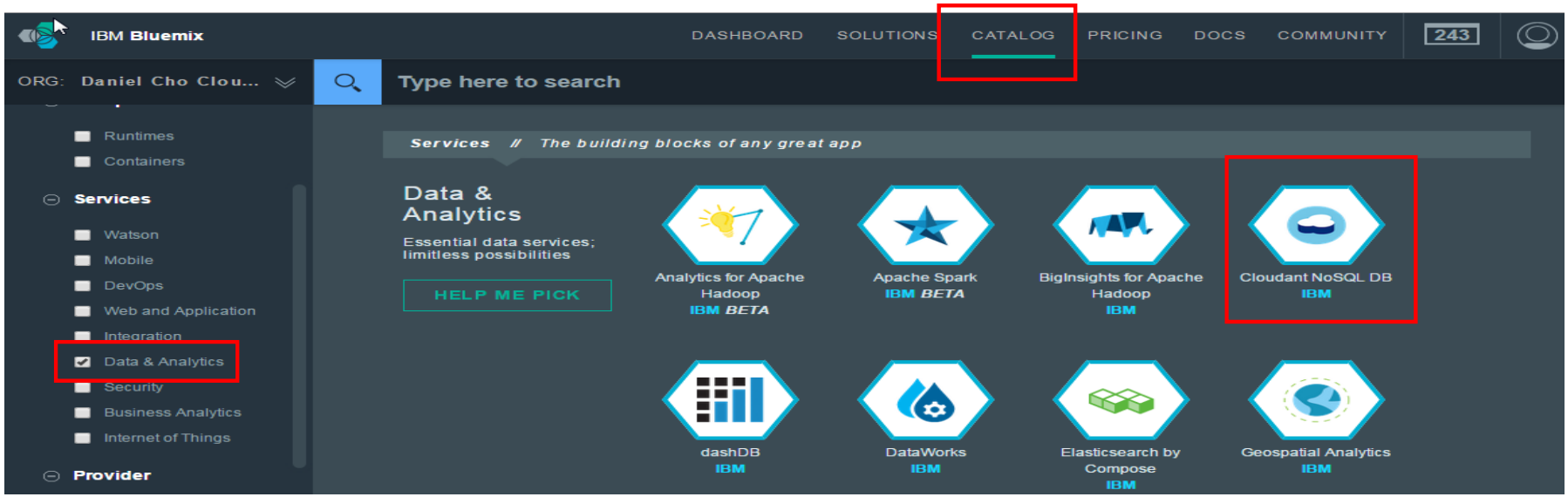
- ❑ Log into Bluemix at https://bluemix.net .

- Click create a space and give the name of ‘Cloudant Lab’.

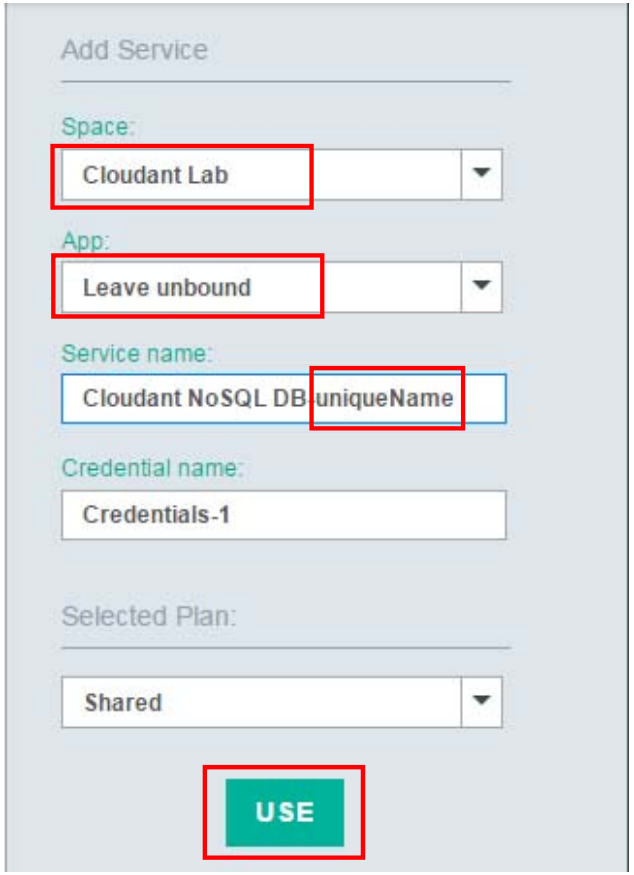


Create an Cloudant NoSQL service instance

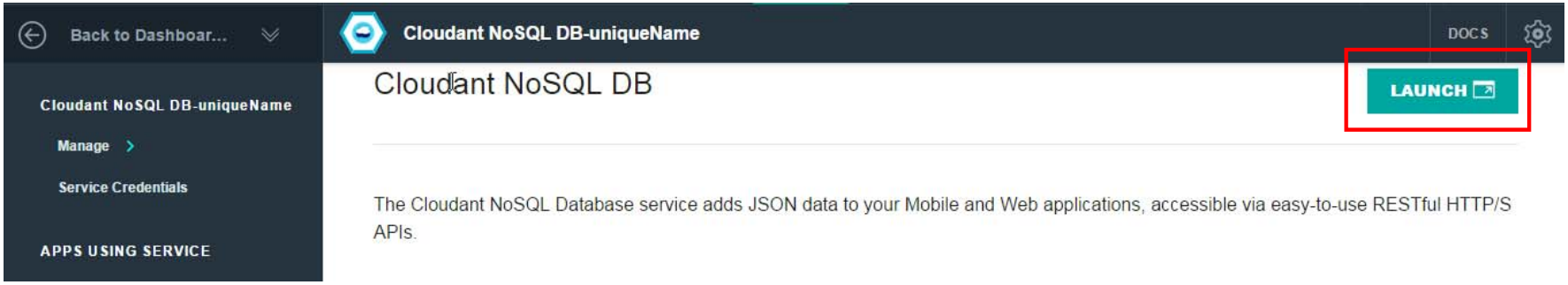
- Click Catalog at the top menu.
- Select Data & Analytics under Services on your left navigation panel.
- Select Cloudant NoSQL DB on your right main panel to create an instance.



- ❑ Create in Cloudant Lab space.
- ❑ Leave the service unbound. It means not bound by any application at this time.
- ❑ Give your own unique name to the service.
- ❑ Take the default for other entries.
- ❑ Then, Click Use.



- You should find the following screen when the Cloudant DB instance is created. Click Launch.

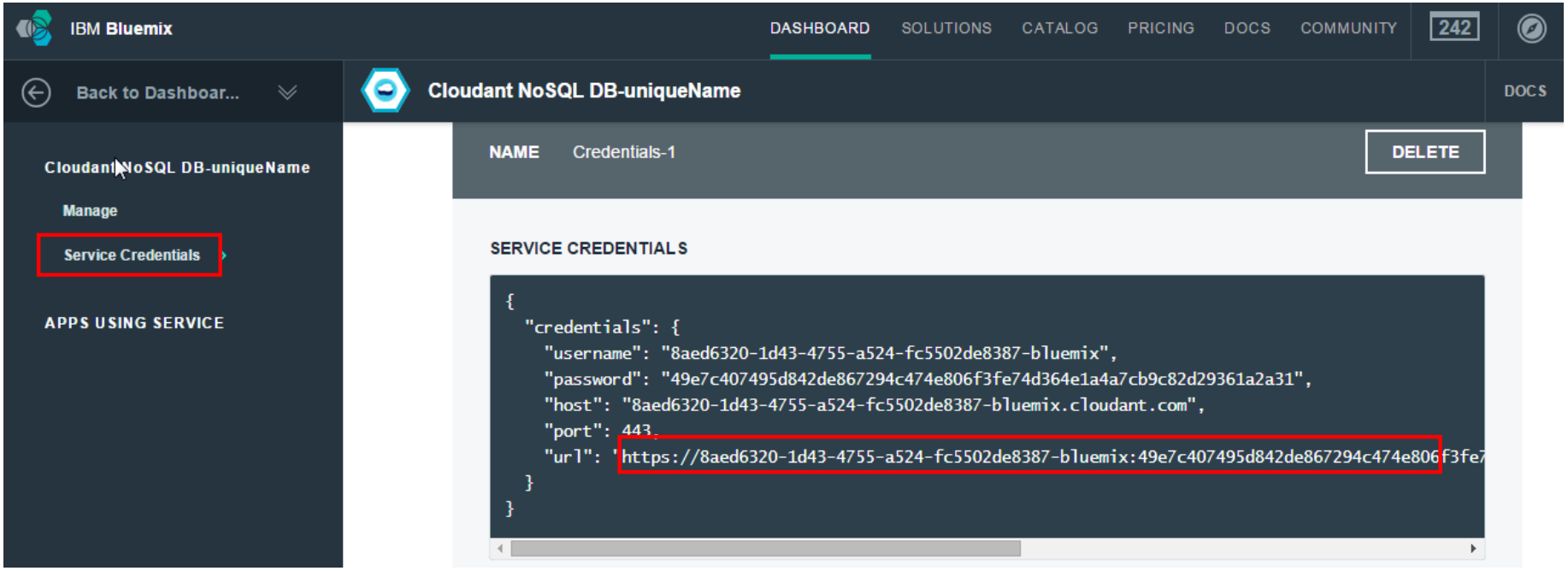


Create a database

- ❑ Click Add New Database.
- ❑ Add 'cars'.
- ❑ Click Create.

Upload lab data to Cloudant DB

- ❑ Navigate back to the Cloudant DB service instance main page.
- ❑ Click Service Credentials on your left navigation panel.
- ❑ Copy URL on your right main panel to access the DB remotely. The URL format is https://username:password@username.cloudant.com



- ❑ Navigate back to the local folder where you previously unzipped the lab material from GitHub.
- ❑ Locate upload.bat file and open it edit mode.
- ❑ Replace the red boxed section with your Cloudant DB instance URL copied from the previous step.
  cURL tool enables data transfer with URL syntax in command lines or scripts. In the lab context, upload.bat file establishes a connection to your Cloudant DB instance, upload initial JSON data documents contained in data.json file, and record the server response in output file.

upload.bat *

```
curl -vX POST https://user:pwd@user.cloudant.com/cars/_bulk_docs -d @- -k -# -o output -H "Content-Type: application/json" < data.json
```

upload.bat

```
curl -vX POST https://8aed6320-1d43-4755-a524-fc5502de8387-bluemix:49e7c407495d842de867294c474e806f3fe74d364e1a4a7cb9c82d29361a2a31@8aed6320-1d43-4755-a524-fc5502de8387-bluemix.cloudant.com/c
```

- Double click upload.bat.
- If it fails, ensure your Cloudant DB service instance URL included in upload.bat file is correctly pasted.
- If the URL is not an issue, type in curl –V at command line. You should see the cURL installation version as the below screen.

```
C:\Users\IBM_ADMIN\Downloads\cloudantLab-master\cloudantLab-master>curl -V
curl 7.43.0 (x86_64-pc-win32) libcurl/7.43.0 OpenSSL/1.0.2c zlib/1.2.8 WinIDN li
bssh2/1.4.3_DEV
Protocols: dict file ftp ftps gopher http https imap imaps ldap pop3 pop3s rtsp
scp sftp smtp smtps telnet tftp
Features: AsynchDNS IDN IPv6 Largefile SSPI Kerberos SPNEGO NTLM SSL libz

C:\Users\IBM_ADMIN\Downloads\cloudantLab-master\cloudantLab-master>_
```

- If curl command is not recognized, install curl-7.43.0-win64-local.msi found in the same unzipped folder. Double click.

Name	Date modified	Type	Size
curl-7.43.0-win64-local.msi	9/16/2015 4:19 PM	Windows Installer ...	4,612 KB
data.json	9/16/2015 11:46 AM	JSON File	1 KB
Exercise1.txt	9/16/2015 11:46 AM	Text Document	2 KB
Exercise2.txt	9/16/2015 11:46 AM	Text Document	1 KB
Exercise3.txt	9/16/2015 11:46 AM	Text Document	1 KB
Exercise4.txt	9/16/2015 11:46 AM	Text Document	1 KB
Exercise5.txt	9/16/2015 11:46 AM	Text Document	1 KB
Exercise6.txt	9/16/2015 11:46 AM	Text Document	1 KB
output	9/16/2015 4:25 PM	File	1 KB
upload.bat	9/16/2015 4:53 PM	Windows Batch File	1 KB

- Double click upload.bat again.
- Check output file. If successful, you should see a response similar to the following.

output

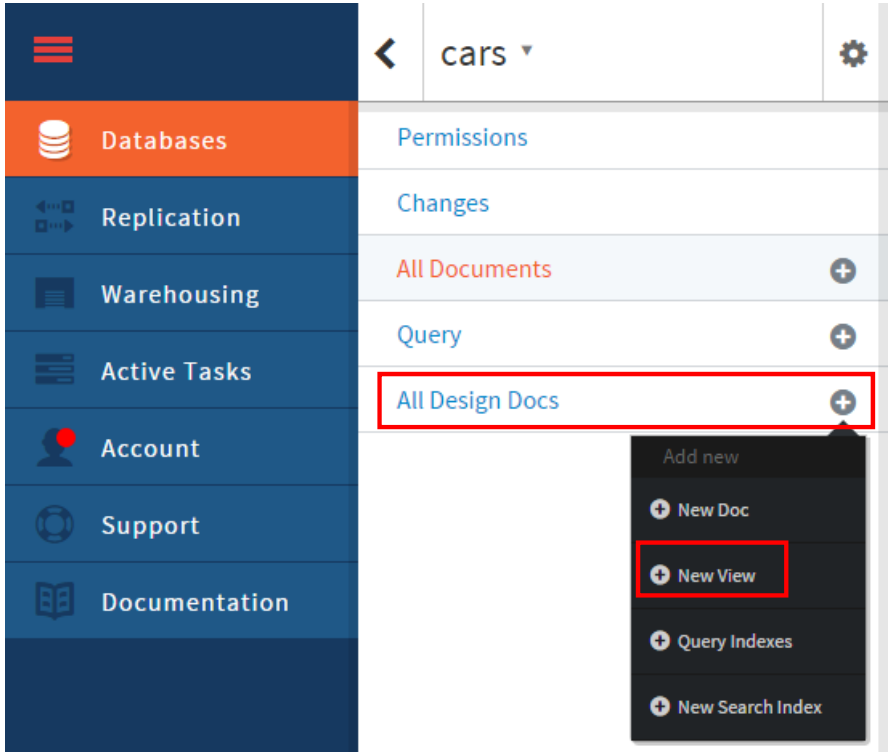

```
[{"id":"car1","rev":"1-369378d33c9bf23ef23161c11f805de7"},
        {"id":"car2","rev":"1-269ce93e3c3b54f425f009a5edcda9b0"},
        {"id":"car3","rev":"1-77fa3d06322089e37cb5040226880c13"},
        {"id":"car4","rev":"1-60225a6be054f702619aee7ec04fa66f"},
        {"id":"car5","rev":"1-b22172fafe2e4abc8dd38f39de43a290"},
        {"id":"car6","rev":"1-a93563db88a04bfd7ecae5664aec37b7"},
        {"id":"car7","rev":"1-b72aeca1e71231dfc0521c9a1cd15b8e"}]
```

Lab Exercise 1: Add Views

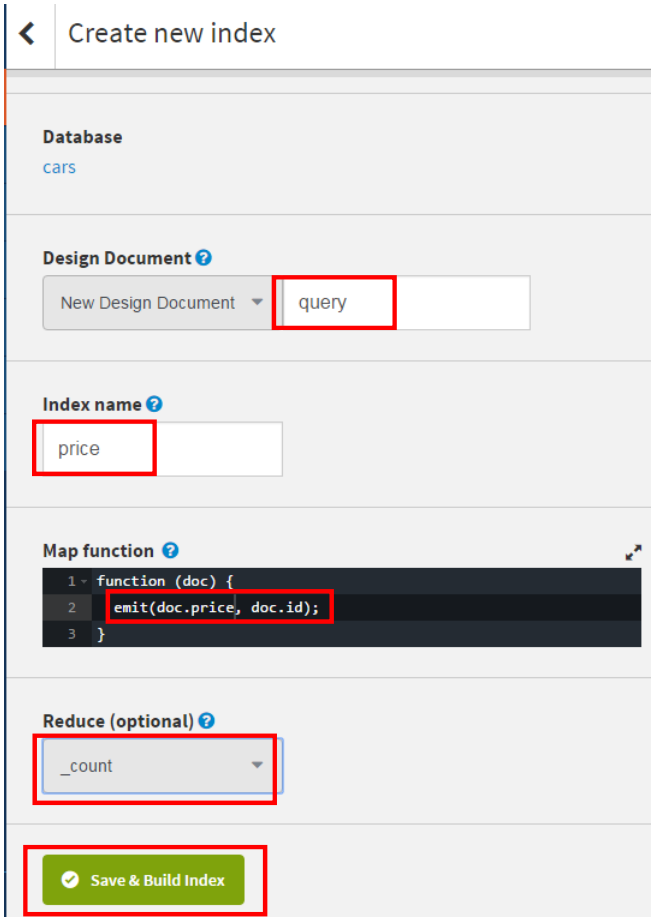
Create Price View

This exercise is to create secondary index, ideal for routine queries. It use MapReduce to build index over large mount of data. These functions are written in JavaScript and held in 'design documents'. Please feel free to visit https://cloudant.com/for-developers/views for additional information.

- ❑ Navigate to cars DB.
- ❑ Click + symbol next to All Design Docs.
- ❑ Click New View.



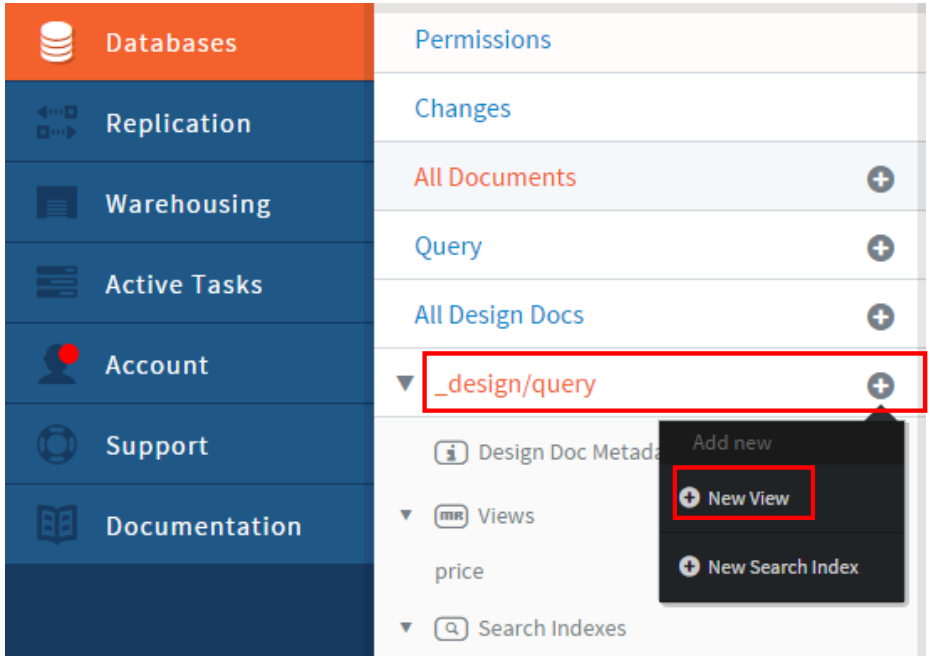
- ❑ Add ‘query’ next to New Deisgn Document.
- ❑ Add ‘price’ under Index Name.
- ❑ Modify Map function with emit(doc.price, doc._id);
- ❑ Select _count under Reduce function.
- ❑ Click Save & Build Index.



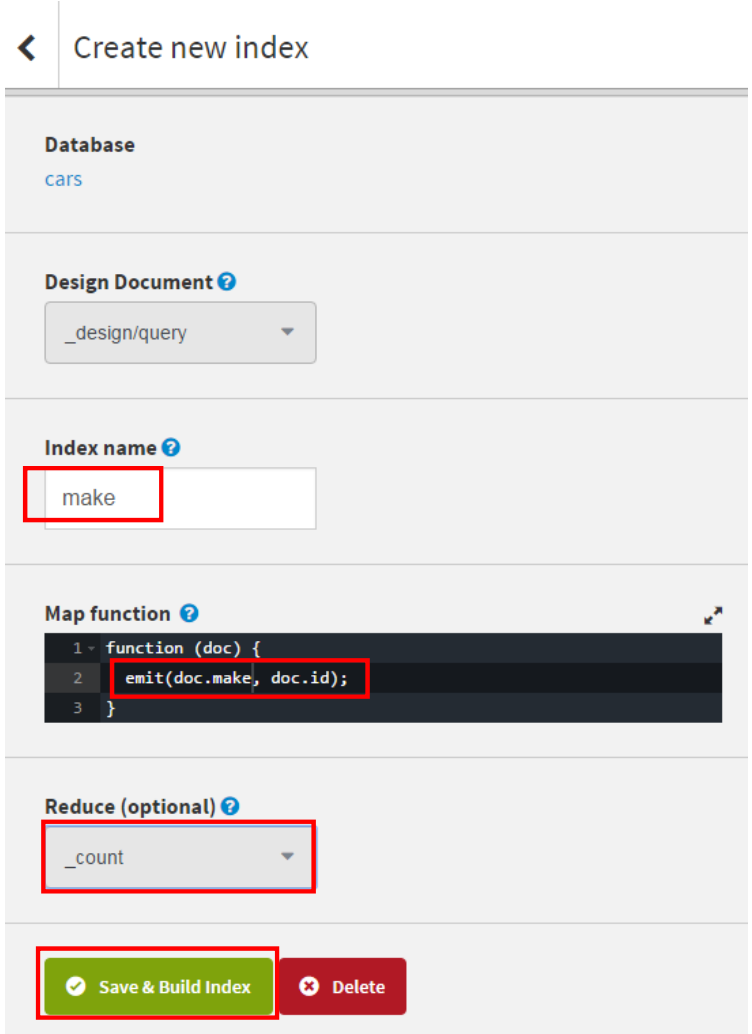
21

Create Make View

- ❑ _design/query document was created in the previous step. Click + symbol next to _design/query.
- ❑ Click New View.



- ❑ Add ‘make’ under Index Name.
- ❑ Modify Map function with emit(doc.make, doc._id);
- ❑ Select _count under Reduce function.
- ❑ Click Save & Build Index.



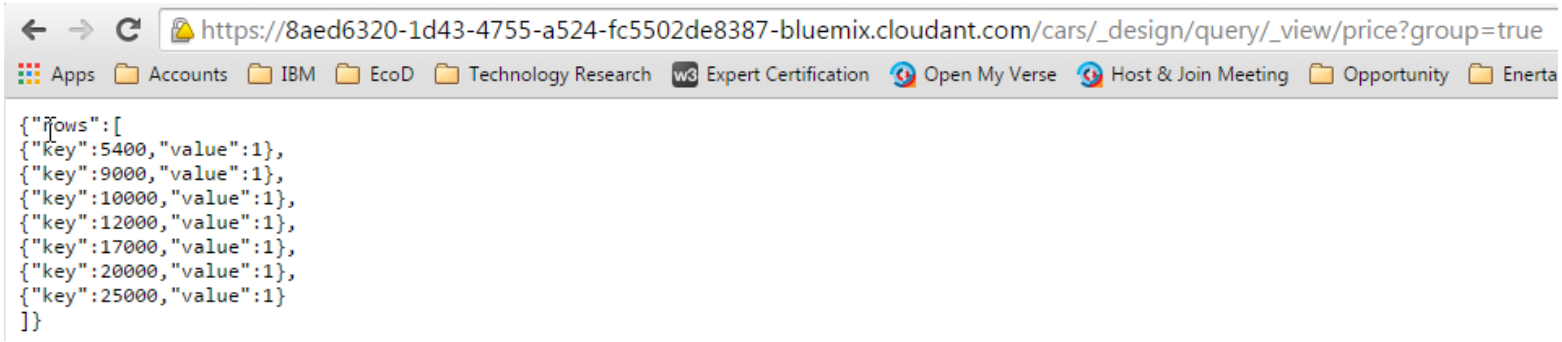
Lab Exercise 2: Make RESTful Call to Cloudant DB using Price View

This exercise is to make a restful call to the Cloudant DB using price view which returns unique price (key) and aggregated count(value) as defined in the MapReduce function.

- ❑ Get the Cloudant DB URL by navigating to Bluemix Cloudant DB service instance main page.
- ❑ Select Service Credential on your left navigation panel.
- ❑ Copy host URL on your right main panel.

- ❑ Open an internet browser.
- ❑ Type in https:// and paste the DB host URL.
- ❑ Type in /cars/_design/query/_view/price?group=true
- ❑ Click Enter. (If user name and password is requested, provide them from the Service Credentials).

Should receive a response similar to the following on your browser main panel.



Lab Exercise 3: Make RESTful Call to Cloudant DB using Price View for Price Range Search with Count

This exercise is to make a restful call to Cloudant DB for a range of prices defined in the URL. Please make a note on the usage of startkey and end alogn with price view.

- ❑ Open an internet browser.
- ❑ Type in https:// and paste the DB host URL.
- ❑ Type in /cars/_design/query/_view/price?group=true&startkey=5000&endkey=30000 >> Click Enter.

https://8aed6320-1d43-4755-a524-fc5502de8387-bluemix.cloudant.com/cars/_design/query/_view/price?group=true&startkey=5000&endkey=30000

Should receive a response similar to the following.

https://8aed6320-1d43-4755-a524-fc5502de8387-bluemix.cloudant.com/cars/_design/query/_view/price?group=true&startkey=5000&endkey=30000

Apps | Accounts | IBM | EcoD | Technology Research | Expert Certification | Open My Verse | Host & Join Meeting | Opportunity | Enertainment | Home Improvement


```
{"rows":[
{"key":5400,"value":1},
{"key":9000,"value":1},
{"key":10000,"value":1},
{"key":12000,"value":1},
{"key":17000,"value":1},
{"key":20000,"value":1},
{"key":25000,"value":1}
]}
```

Let’s change the range to see the return values.

- ❑ Change endkey=10000 in the Rest call URL. Click Enter.

https://8aed6320-1d43-4755-a524-fc5502de8387-bluemix.cloudant.com/cars/_design/query/_view/price?group=true&startkey=5000&endkey=10000

Price (unique key) and aggregated count(value) beyond 10,000 does not return in the response.

https://8aed6320-1d43-4755-a524-fc5502de8387-bluemix.cloudant.com/cars/_design/query/_view/price?group=true&startkey=5000&endkey=10000

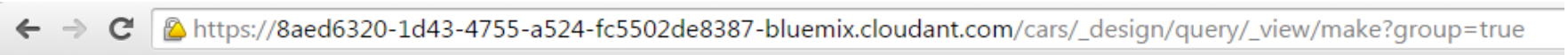
Apps | Accounts | IBM | EcoD | Technology Research | Expert Certification | Open My Verse | Host & Join Meeting | Opportunity | Enertainment | Home Improvement


```
{"rows":[
{"key":5400,"value":1},
{"key":9000,"value":1},
{"key":10000,"value":1}
]}
```

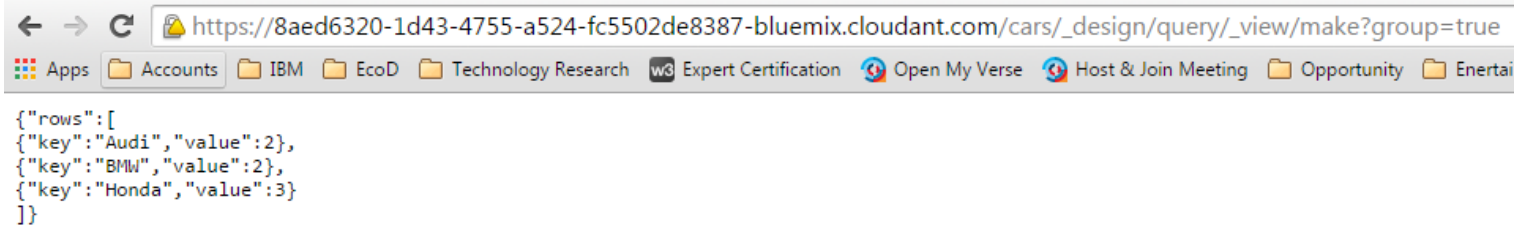
Lab Exercise 4: Make RESTful Call to Cloudant DB using Make View

This exercise is to make a restful call to the Cloudant DB using make view which returns unique make (key) and aggregated count(value) as defined in the MapReduce function.

- ❑ Open an internet browser.
- ❑ Type in https:// and paste the DB host URL.
- ❑ Type in /cars/_design/query/_view/make?group=true
- ❑ Click Enter.



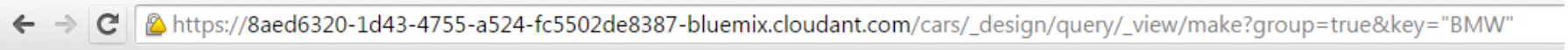
Should receive a response similar to the following.



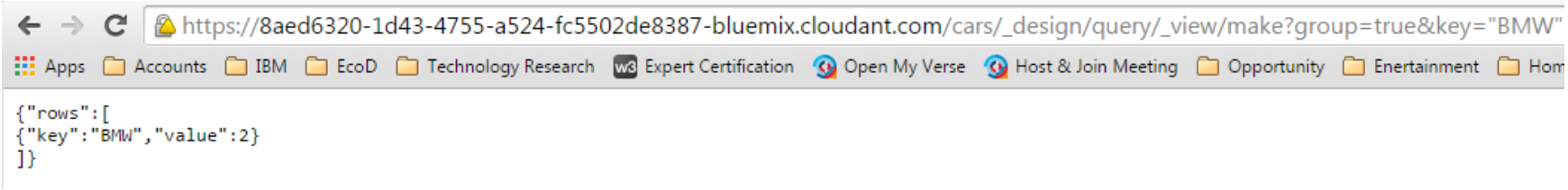
Lab Exercise 5: REST Call to Cloudant DB using Make View for a Specific Model Search with Count

This exercise is to make a restful call to the Cloudant DB using make view targeting a specific model indicated by the key parameter in the URL

- ❑ Open an internet browser.
- ❑ Type in https:// and paste the DB host URL.
- ❑ Type in /cars/_design/query/_view/make?group=true&key="BMW"
- ❑ Click Enter.



Should receive a response similar to the following.

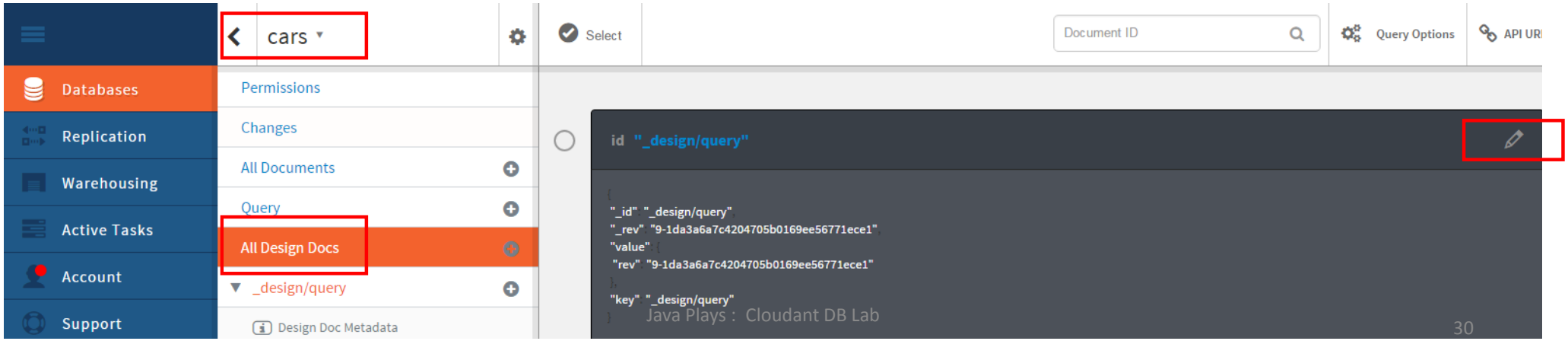


Lab Exercise 6: Make Restful Call to Cloudant DB using List Function

Create List

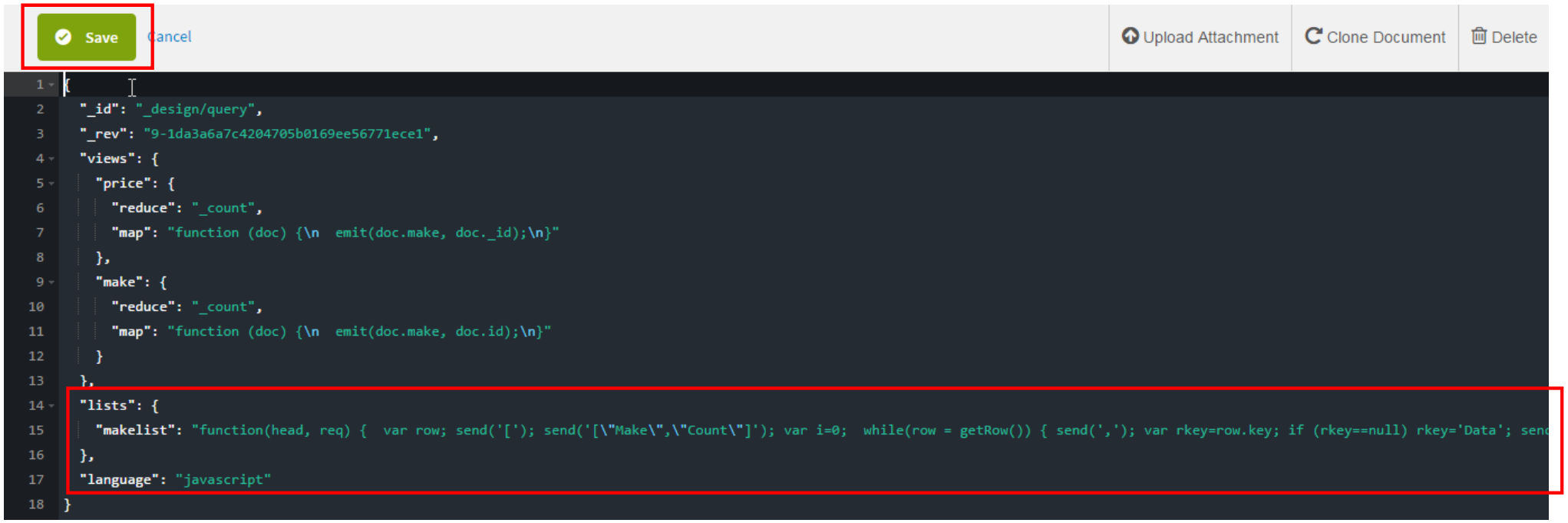
List function is to customize the format of MapReduce query results. In this exercise, you will add a list function written in JavaScript returning data in [Make, Count] list format from the query result using make view.

- ❑ Navigate to the Cloudant DB management view.
- ❑ Click cars DB.
- ❑ Click All Design Docs.
- ❑ Click pencil symbol to edit the query design document.



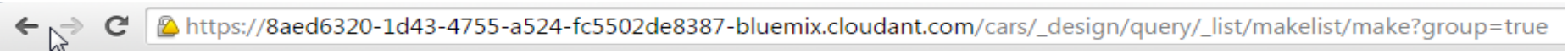
- Add the lists with makelist function as in the following screen.

"lists": {
"makelist": "function(head, req) { var row; send('['); send('[\"Make\",\"Count\"]'); var i=0; while(row = getRow()) { send(','); var rkey=row.key; if (rkey==null) rkey='Data'; send('[\"'+rkey+'\",'+row.value+']'); i++; }send(']');}"
},
"language": "javascript"

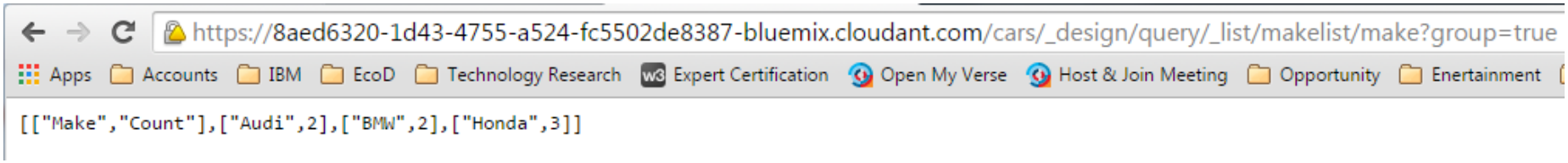


Test Lists Function

- ❑ Open an internet browser.
- ❑ Type in https:// and paste the DB host URL.
- ❑ Type in /cars/_design/query/_list/makelist/make?group=true
- ❑ Click Enter.



Data returns in list format with Make and Count as defined in the query design.



This ends the Cloudant DB Lab.

Thank you !