



# IBM CLOUD

INTEGRATION OF PHP APPLICATION WITH CLOUDANT  
DATABASE

-Bikash Das

# Integration of php application with Cloudant database in ibmcloud

- ▶ **Part A** - Deploying locally
  - ▶ Cloudant
  - ▶ CouchDB
  - ▶ http methods
  - ▶ Curl
  - ▶ futon
  - ▶ Php sending data over http
- ▶ **Part B** - Deploying in cloud
  - ▶ Create a php instance
  - ▶ Create a Cloudant service and link them
  - ▶ Upload the source files to the cloud

# Cloudant

- ▶ **Cloudant** is an IBM software product, which is primarily delivered as a cloud-based service.
- ▶ Secure: Data's are encrypted
- ▶ Data Flexibility: Flexible JSON schema and powerful API that is compatible with Apache CouchDB
- ▶ General Availability: Available in all IBM Cloud regions

NOTE: for this demonstration purpose, I will use in CouchDB.

Later we will use Cloudant

# CouchDB

- ▶ CouchDB falls into the category of databases called NoSQL
- ▶ NoSQL is an acronym for Not Only SQL
- ▶ SQL database stores info in terms of tables and have schemas whereas NoSQL are generally less bound to schema.
- ▶ CouchDB data models are documents.
- ▶ Every documents has a required ID field and that makes the document uniquely identifiable.
- ▶ Documents are unique records in CouchDB. They are lot like json objects

# CouchDB Document

A Sample Document



```
1 {  
2   "_id": "121ebd24666e0fde290172acbf00d5d4",  
3   "_rev": "1-9650007cab14f567a6aeef7cda9c2cfa",  
4   "type": "user",  
5   "name": "Bikash",  
6   "msg": "A secret message"  
7 }
```

Each document has reserved fields for metadata such as id, revision and deleted.

**\_id** = it stores the id of the document which is unique

**\_rev** = it indicates the revision id. Every time you make changes to the document a new rev value will be generated.

# HTTP METHODS

- ▶ Four main HTTP Methods for communicating with CouchDB
  - ▶ GET, PUT, POST and DELETE
- ▶ Four main operations we perform on databases
  - ▶ Create, Read, Update and Delete (CRUD)

CRUD	HTTP Methods
CREATE	POST
READ	GET
UPDATE	PUT
DELETE	DELETE

- ▶ We will use CURL to explore each of the http methods.

# curl

- ▶ cURL – Client for URLs or Client URL Request Library
- ▶ CouchDB cURL utility is used to communicate with CouchDB database
- ▶ curl is a command-line tool for getting or sending data including files using URL syntax
- ▶ Curl tool is used to transfer data from or to a server by using one of the supported protocols like HTTP, HTTPS, FTP, etc
- ▶ It is a command-line utility to access HTTP protocol straight away from the command line.
- ▶ We can also access a webpage using cURL e.g. curl [www.google.com](http://www.google.com)

# Using curl to access CouchDB

- ▶ Download CouchDB from <http://couchdb.apache.org/> and install
  - ▶ Go to C:\Apache24\conf
  - ▶ Open httpd.conf file. Search for #LoadModule rewrite\_module modules/mod\_rewrite.so **Remove #**
  - ▶ Also search for Options Indexes FollowSymLinks, below you will find AllowOverride **None**, Change to **All**
- ▶ Check if installed
  - ▶ In cmd type: **curl http://localhost:5984/**
  - ▶ If reply come {couch: welcome ..} it means installed
- ▶ LIST ALL THE DATABASE: curl -X GET http://localhost:5984/\_all\_dbs
- ▶ Create a database: curl -X PUT http://localhost:5984/dbname
  - ▶ **Note:** dbname = your database name



# Using curl to access CouchDB

- ▶ Get all documents from database
  - ▶ `curl -X GET http://localhost:5984/database_name/_all_docs`
- ▶ Create a document in a database
  - ▶ `curl -X PUT -H "Content-Type:application/json" -d '{"type":"user","Name":"Bikash Das", "Location":"Pondicherry University"}' http://localhost:5984/mydb/1`
- ▶ Update a document in a database
  - ▶ Pass the revision id to update a document
  - ▶ `curl -H 'Content-Type: application/json' -X PUT http://localhost:5984/mydb/1/ -d '{"Name":"Bikash","Location":"India", "_rev":"1-5106e13a3423cd7f9a47d9254d8d0e81"}'`

# Using curl to access CouchDB

- ▶ Delete a document
  - ▶ First get the revision id (`_rev`) of a document
  - ▶ `Curl -H 'Content-Type: application/json' -X DELETE http://localhost:5984/database_name/database_id?rev=_rev`
  - ▶ E.g. : `curl -H 'Content-Type: application/json' -X DELETE http://localhost:5984/mydb/1?rev=3-ace56d6d59d6eda4a8331cf6b82c0e1a`
- ▶ When username and password is set:
  - ▶ add `-u username:password`
  - ▶ `curl -u username:password -X GET http://localhost:5984/`
- ▶ To see how the connection is being established. Add `-v` with any curl statement, for e.g.:
  - ▶ `curl -v -X GET http://localhost:5984/_all_dbs`

# Futon

- ▶ CouchDB comes with a build-in web-based administration console called **Futon**.
- ▶ Futon has a simple interface which allows you to manage databases, users and documents .
- ▶ It comes packed with CouchDB. There's no need to download separately.
- ▶ Open browser and visit
  - ▶ [http://127.0.0.1:5984/\\_utils/](http://127.0.0.1:5984/_utils/)

# PHP — SENDING DATA OVER HTTP

- ▶ The curl extension to php is designed to allow you to use a variety of web resources from within your PHP script.
- ▶ There are four steps:
  - ▶ **Initialize Curl** -> `$curl = curl_init();`
  - ▶ **Set your options** -> `curl_setopt($curl, CURLOPT_URL, http://www.php.net)`
  - ▶ **Execute curl query** -> `curl_exec($curl);`
  - ▶ **Close URL** -> `curl_close($curl)`
- ▶ Step 2 is the complicated one, because majority of the work takes place here

# PHP

Curl options	description
CURLOPT_POST	1 if you want Curl to do a regular HTTP POST
CURLOPT_RETURNTRANSFER	1 if you want Curl to return the transfer data instead of printing it out directly
CURLOPT_URL	A string containing the URL you want Curl to fetch
CURLOPT_HEADER	("Content-Type: application/json"
CURLOPT_USERPWD	A String formatted in the username:password manner, for Curl to give to the remote server if requested
CURLOPT_POSTFIELDS	A String containing the data to post in the HTTP "POST" operation
CURLOPT_CUSTOMREQUEST	Used when deleting

# Curl to php equivalent

Curlopt\_header =>  
("Content-Type: application/json")

Bikash Das  
PU

```
curl -X PUT -H "Content-Type:application/json"  
-d '{"type":"user","Name":"Bikash Das", "Location":"Pondicherry University"}'  
http://localhost:5984/mydb/id01
```

Curlopt\_post => 1

Curlopt\_url => "http://localhost:5984/mydb/"

Curlopt\_postfields => '{"type":"user","Name":"Bikash Das", "Location":"Pondicherry University"}'

Pass the id along with the url

# PHP — CREATING A DOCUMENT

```
<form action="welcome.php" method="post">
  <input id="name" name="name" type="text" required placeholder="Enter Name"><br>
  <input id="msg" name="msg" type="text" required placeholder="Enter Secret Message"><br>
  <input id="s" type="Submit" value="Send">
</form>
```

Index.html

welcome.php file

```
$user = new stdClass;
$user->type = 'user';
$user->name = $_POST["name"];
$user->msg = $_POST["msg"];
//echo json_encode($user);

$curl = curl_init();
// curl options
$options = array(
    CURLOPT_URL => 'http://localhost:5984/mydb',
    CURLOPT_POST => 1,
    CURLOPT_POSTFIELDS => json_encode($user),
    CURLOPT_HTTPHEADER => array ("Content-Type: application/json"),
    CURLOPT_RETURNTRANSFER => true /* if not, curl_exec() throws output*/
);
curl_setopt_array($curl, $options);
curl_exec($curl);
curl_close($curl);
```

Post data coming from  
the form is stored in  
\$user



# PHP — DISPLAYING DOCUMENTS

The code snippet to fetch a document from the database using id


```
<?php
function get($id){
    // Get cURL resource
    $curl = curl_init();
    curl_setopt_array($curl, [
        CURLOPT_RETURNTRANSFER => true, //or 1
        CURLOPT_URL => 'localhost:5984/mydb/'.$id,
        CURLOPT_HTTPHEADER => array ("Content-Type: application/json"),
    ]);
    // Send the request & save response to $resp
    $resp = curl_exec($curl);
    $res= json_decode($resp,true); //result is associative array php
    curl_close($curl);
    echo $res;
}
```



# PHP — DELETING DOCUMENT

The code to delete a document

You need to pass database name, revision id and document id to delete a document



```
function deldoc($db,$rev,$id){
    $curl = curl_init();
    curl_setopt_array($curl, [
        CURLOPT_RETURNTRANSFER => true, //or 1
        CURLOPT_URL => 'http://localhost:5984/' . $db . '/' . $id . '?rev=' . $rev,
        CURLOPT_CUSTOMREQUEST => 'DELETE'
    ]);
    curl_exec($curl);
    curl_close($curl);
    echo 'deleted document with id: ' . $id;
}
```

# Part B

- ▶ We will create a php instance and add a Cloudant service to it. Link them and upload the source files.

# Creating PHP instance

-IBMCLOUD

- ▶ **Sign Up** if you don't have IBM cloud account url:  
<https://console.bluemix.net/registration/>
- ▶ **Log In** using email and password
- ▶ Click on **Catalog** in the navigation bar
- ▶ Click on **Compute** in the side navigation bar
- ▶ Scroll down and choose PHP
- ▶ Give app name, host name and set region (which location's data centre you want to use) and click Create
- ▶ Click on **Visit App URL**
- ▶ A Sample App is Running

# Download IBM Cloud CLI

- ▶ Use IBM Cloud command line interface to download, modify, and redeploy your Cloud Foundry applications and service instances.
- ▶ [https://console.bluemix.net/docs/cli/reference/ibmcloud/download\\_cli.html#install\\_use](https://console.bluemix.net/docs/cli/reference/ibmcloud/download_cli.html#install_use)
- ▶ Select the installer based on your Operating System.
- ▶ After downloading, Install it.

# LOGIN through CLI

- ▶ Open command prompt in windows or Terminal in Linux
- ▶ Go to the folder where your files are stored or create a new folder
- ▶ Login to ibmcloud
- ▶ C:\Users\Bikash\Desktop\demo> **ibmcloud login**
- ▶ Enter the Email and Password
- ▶ Now type: **ibmcloud target --cf** to set CF API endpoint, Org and Space.
- ▶ Deploy: **ibmcloud app push app\_name -b php\_buildpack**
  - ▶ Note: app\_name should be the name of your app you set while creating php instance

# Creating Cloudbant service instance

-IBMCLOUD

- ▶ **Sign Up** if you don't have IBM cloud account url:  
<https://console.bluemix.net/registration/>
- ▶ **Log In** using email and password
- ▶ Click on **Catalog** in the navigation bar
- ▶ Click on **Databases** in the side navigation bar
- ▶ Choose Cloudbant
- ▶ Give **Service name**, choose region and set authentication methods and click **Create**
- ▶ Service is **started**, you will be redirected to dashboard

# Link database with php instance

- ▶ Go to **Dashboard**, click on the name of the php application
- ▶ In the side navigation bar, click on **connection**
- ▶ To the right, you will see a button **Create connection**. Click on that.
- ▶ Leave the default and click on **Connect**.
- ▶ Click on **Restage** button.
- ▶ Done.

# Service Credential

- ▶ **Dashboard**, click on database name
- ▶ Click on **service credential** from the side navigation bar
- ▶ Click **new credential**, Under service id, select auto generate
- ▶ Click **Add**
- ▶ A new Service Credential will be added which contains the username, password and the URL to connect to the Cloudant database with php application
- ▶ Click on **View credentials** and copy the URL



# Service Credential

```
{
  "apikey": "pYi9MPcRw1hYnMP70H8JIiLaE-e6mXCGwvv4MAe-Y05B",
  "host": "d9cb693a-8304-4ff1-93a9-85516f86cc3d-bluemix.cloudantnosqldb.appdomain.cloud",
  "iam_apikey_description": "Auto generated apikey during resource-key operation for Instance - crn:v1:bluemix:public:cloudantnosqldb:eu-gb:a/f3a2a451fa264d7591a20f7472f4d6df:82009d43-d2e1-4d81-a2e0-82cda41998da::",
  "iam_apikey_name": "auto-generated-apikey-86ce4abb-3127-4066-918b-ef7fc4a64dc8",
  "iam_role_crn": "crn:v1:bluemix:public:iam::::serviceRole:Manager",
  "iam_serviceid_crn": "crn:v1:bluemix:public:iam-identity::a/f3a2a451fa264d7591a20f7472f4d6df::serviceid:ServiceId-cd80a067-9133-4182-ae24-46db6c186461",
  "password": "60f2a099aa6775d3284dc69f850d36e79975f2bafcf01fc2c2a61e6abaa1d7d2",
  "port": 443,
  "url": "https://d9cb693a-8304-4ff1-93a9-85516f86cc3d-bluemix:60f2a099aa6775d3284dc69f850d36e79975f2bafcf01fc2c2a61e6abaa1d7d2@d9cb693a-8304-4ff1-93a9-85516f86cc3d-bluemix.cloudantnosqldb.appdomain.cloud",
  "username": "d9cb693a-8304-4ff1-93a9-85516f86cc3d-bluemix"
}
```

Copy the URL and paste it in the CURLOPT\_URL field in php source code

# Final Change

- ▶ Copy the URL and paste it in the `CURLOPT_URL` field in php source code. The URL contains the `username:password`.
- ▶ Upload the file
  - ▶ First login from terminal: `ibmcloud login`
  - ▶ Enter IBM CLOUD account: `email and password`
  - ▶ Upload the source code
  - ▶ `C:\> ibmcloud app push app_name -b php_buildpack`
- ▶ Visit the URL
- ▶ And Voila. It works!

# References

- ▶ CouchDB The Definitive Guide
- ▶ PHP in a Nutshell book
- ▶ JSON PHP W3SCHOOL.COM
- ▶ <https://www.javatpoint.com/couchdb-create-document>
- ▶ <https://codular.com/curl-with-php>
- ▶ ibmcloud documentation
- ▶ [https://console.bluemix.net/docs/cli/reference/ibmcloud/download\\_cli.html#install\\_use](https://console.bluemix.net/docs/cli/reference/ibmcloud/download_cli.html#install_use)
- ▶ <https://searchcloudcomputing.techtarget.com/definition/Cloud-Foundry>
- ▶ <http://thisinterestsme.com/php-error-handling-curl/>