

## cURL Utility

cURL utility is a way to communicate with CouchDB.

It is a tool to transfer data from or to a server, using one of the supported protocols *HTTP, HTTPS, FTP, FTPS, TFTP, DICT, TELNET, LDAP or FILE*. This command is designed to work without user interaction. cURL offers a busload of useful tricks like proxy support, user authentication, ftp upload, HTTP post, SSL *https*: connections, cookies, file transfer resume and more.

The cURL utility is available for operating systems such as UNIX, Linux, Mac OS X and Windows. It is a command-line utility to access HTTP protocol straight away from the command line.

## Using cURL Utility

You can access any website using cURL utility by simply typing cURL followed by the website address as shown below –

```
curl www.tutorialspoint.com/
```

By default, the cURL utility returns the source code of the requested page. It displays this code on the terminal window.

## cURL Utility Options

cURL utility provides various options to work with, and you can see them in cURL utility help.

The following code shows some portion of cURL help.

```
$ curl --help
```

While communicating with CouchDB, certain options of cURL utility were extensively used. Following are the brief descriptions of some important options of cURL utility including those used by CouchDB –

### -X flag

When communicating with an HTTP server, -X flag allows to specify a request method *GET, POST, PUT etc.* to override the default method *GET*.

When communicating with an FTP server, -X flag allows to specify FTP command to override the default command *LIST*.

### -H

This flag specifies a custom header to pass on to the server. For example, to send a file to the server, we have to use the "content type" that defines the type of content of the file that we are going to upload.

### -d flag

Using this flag of cURL, you can send data along with the HTTP POST request to the server, as if it was filled by the user in the form and submitted.

## Example

Suppose there is a website and you want to login into it or send some data to the website using -d flag of cURL utility as shown below.

```
curl -X PUT http://mywebsite.com/login.html -d userid=001 -d password=tutorialspoint
```

It sends a post chunk that looks like "**userid=001&password=tutorialspoint**". Likewise you can also send documents *JSON* using -d flag.

## -o flag

Using this flag, cURL writes the output of the request to a file.

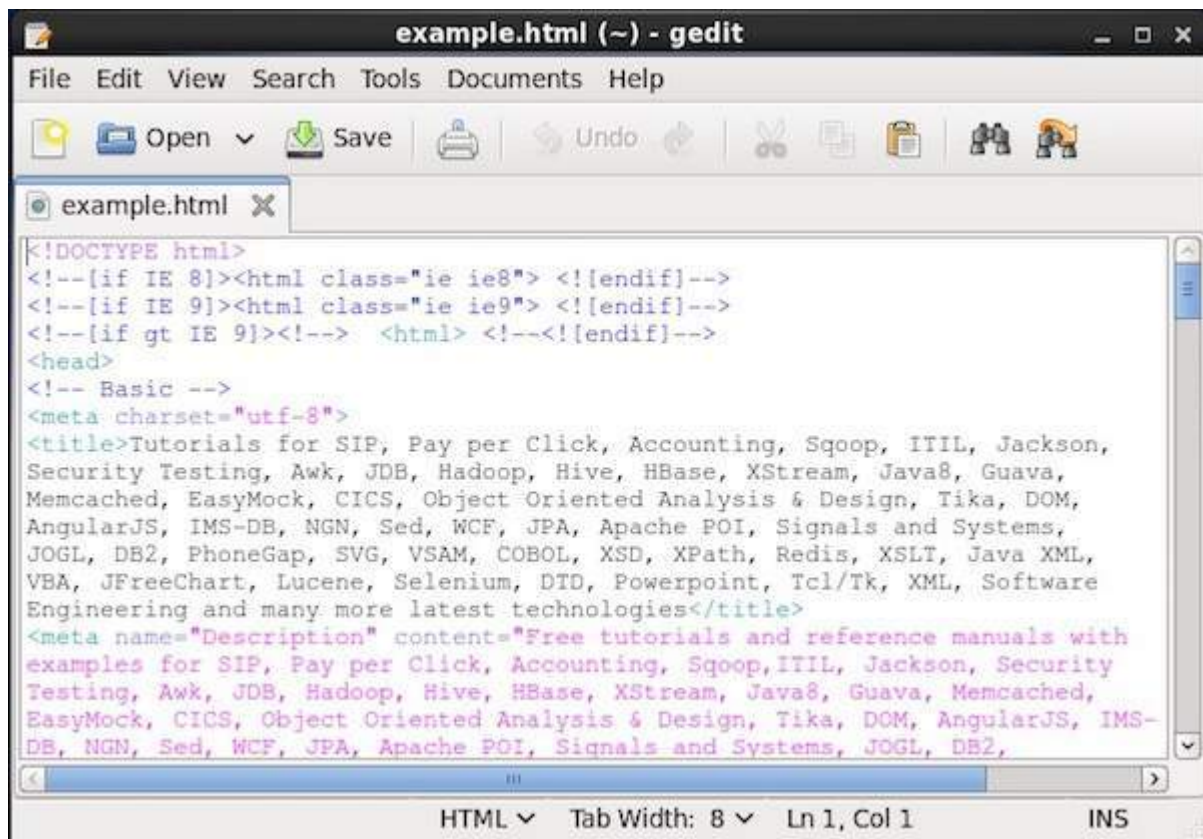
### Example

The following example shows the use of **-o** flag of cURL utility.

```
$ curl -o example.html www.tutorialspoint.com/index.htm
% Total % Received % Xferd Average Speed Time Time Time Current
   Dload Upload Total Spent Left Speed
100 81193 0 81193 0 0 48168 0 --:--:-- 0:00:01 --:--:--
58077
```

This gets the source code of the homepage of tutorialspoint.com, creates a file named example.com and saves the output in the file named example.html.

Following is the snapshot of the **example.html**.



## -O

This flag is similar to **-o**, the only difference is with this flag, a new file with the same name as the requested url was created, and the source code of the requested url will be copied to it.

### Example

The following example shows the use of **-O** flag of cURL utility.

```
$ curl -O www.tutorialspoint.com/index.htm
% Total % Received % Xferd Average Speed Time Time Time Current
   Dload Upload Total Spent Left Speed
100 81285 0 81285 0 0 49794 0 --:--:-- 0:00:01 --:--:--
```

It creates a new file with the name index.htm and saves the source code of the index page of tutorialspoint.com in it.

## Hello CouchDB

You can access the homepage of the CouchDB by sending a GET request to the CouchDB instance installed. First of all make sure you have installed CouchDB in your Linux environment and it is running successfully, and then use the following syntax to send a get request to the CouchDB instance.

```
curl http://127.0.0.1:5984/
```

This gives you a JSON document as shown below where CouchDB specifies the details such as version number, name of the vendor, and version of the software.

```
$ curl http://127.0.0.1:5984/
{
  "couchdb" : "Welcome",
  "uuid" : "8f0d59acd0e179f5e9f0075fa1f5e804",
  "version" : "1.6.1",
  "vendor" : {
    "name":"The Apache Software Foundation",
    "version":"1.6.1"
  }
}
```

## List of All Databases

You can get the list of all the databases created, by sending a get request along with the string "**\_all\_dbs string**". Following is the syntax to get the list of all databases in CouchDB.

```
curl -X GET http://127.0.0.1:5984/_all_dbs
```

It gives you the list of all databases in CouchDB as shown below.

```
$ curl -X GET http://127.0.0.1:5984/_all_dbs
[ "_replicator" , "_users" ]
```

## Creating a Database

You can create a database in CouchDB using cURL with PUT header using the following syntax –

```
$ curl -X PUT http://127.0.0.1:5984/database_name
```

## Example

As an example, using the above given syntax create a database with name **my\_database** as shown below.

```
$ curl -X PUT http://127.0.0.1:5984/my_database
{"ok":true}
```

## Verification

Verify whether the database is created, by listing out all the databases as shown below. Here you can observe the name of newly created database, "**my\_database**" in the list

```
$ curl -X GET http://127.0.0.1:5984/_all_dbs
[ "_replicator " , "_users" , "my_database" ]
```

## Getting Database Info

You can get the information about database using the get request along with the database name. Following is the syntax to get the database information.

### Example

As an example let us get the information of the database named **my\_database** as shown below. Here you can get the information about your database as a response.

```
$ curl -X GET http://127.0.0.1:5984/my_database

{
  "db_name" : "my_database",
  "doc_count" : 0,
  "doc_del_count" : 0,
  "update_seq" : 0,
  "purge_seq" : 0,
  "compact_running" : false,
  "disk_size" : 79,
  "data_size" : 0,
  "instance_start_time" : "1423628520835029",
  "disk_format_version" : 6,
  "committed_update_seq" : 0
}
```

## Futon

Futon is the built-in, web based, administration interface of CouchDB. It provides a simple graphical interface using which you can interact with CouchDB. It is a naive interface and it provides full access to all CouchDB features. Following is the list of those features –

### Databases –

- Creates databases.
- Destroys databases.

### Documents –

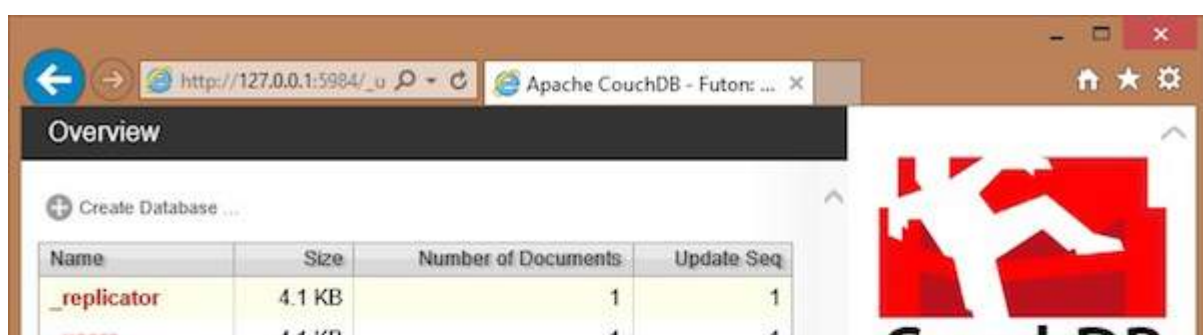
- Creates documents.
- Updates documents.
- Edits documents.
- Deletes documents.

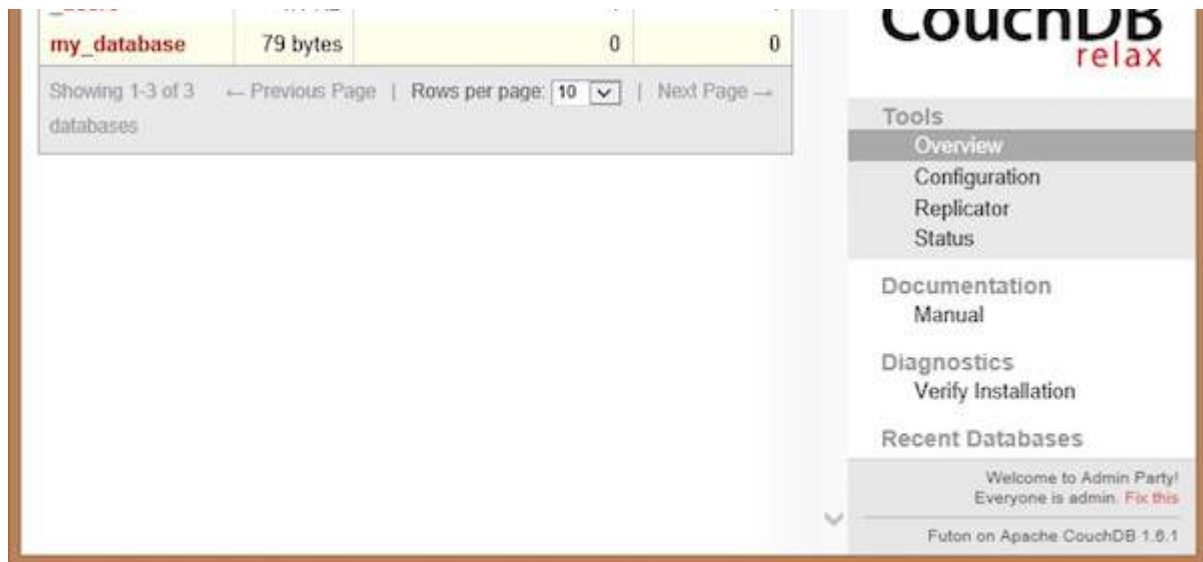
## Starting Futon

Make sure CouchDB is running and then open the following url in browser –

```
http://127.0.0.1:5984/_utils/
```

If you open this url, it displays the Futon home page as shown below –





- On the left hand side of this page you can observe the list of all the current databases of CouchDB. In this illustration, we have a database named **my\_database**, along with system defined databases **\_replicator** and **\_user**.
- On the right hand side you can see the following –
  - **Tools** – In this section you can find **Configuration** to configure CouchDB, **Replicator** to perform replications, and **Status** to verify status of CouchDB and recent modifications done on CouchDB.
  - **Documentation** – This section contains the complete documentation for the recent version of CouchDB.
  - **Diagnostics** – Under this you can verify the installation of CouchDB.
  - **Recent Databases** – Under this you can find the names of recently added databases.

Loading [MathJax]/jax/output/HTML-CSS/jax.js