COUCHDB TUTORIAL

https://www.wisdomjobs.com/e-university/couchdb-tutorial-1522.html

What is CouchDB?

This tutorial facilitates a concise knowledge related to CouchDB, the procedures to set it up, and the ways to communicate with CouchDB server with cURL and Futon. It also says us how to create, update and delete databases and documents.

Audience

This tutorial assists the professionals aspiring to make a career in Big Data and NoSQL databases, particularly the documents store.

Prerequisites

In prior you begin proceeding with this tutorial, we are assuming that you have a concise knowledge on Big Data, Hadoop, and also have the basic knowledge of databases.

CouchDB Tutorial: List of Topics

- Couchdb
  - Couchdb Introduction
  - Couchdb Installation
  - Couchdb Curl & Futon
  - Couchdb Http Api
  - Couchdb Creating A Database
  - Couchdb Deleting A Database
  - Couchdb Creating A Document
  - Couchdb Updating A Document
  - Couchdb Deleting A Document
  - Couchdb Attach Files

What is CouchDB - Introduction?

Database management system facilitates mechanism for storing and retrieval of data. There are three major types of database management systems namely –

- RDBMS
- NoSQL

RDBMS

RDBMS stands for Relational Database Management System. RDBMS is the basis for SQL, and for all modern database systems such as MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.
A (RDBMS) is a database management system (DBMS) that is based on the relational model that was introduced by E. F. Codd.
The data in RDBMS is stored in database objects known astables. The table is a collection of associated data entries and it contains columns and rows. It will store structured data only.

NoSQL Databases

A NoSQL database (also called as Not Only SQL) is a database that facilittaes a mechanism for storing and retrieving of data other than the tabular relations employed in relational databases. These databases are schema-free, support easy replication, have simple API, eventually consistent, and have ability to handle large amounts of data (big data).
The main objective of a NoSQL database is to have the below −

- Simplicity of design,
- Horizontal scaling, and
- Finer control over availability.

NoSQL databases will use different data structures in comparision to relational databases. It makes some operations faster in NoSQL. The suitability of a given NoSQL database merely depends on the problem it should solve. These databases will store both structured data and unstructured data such as audio files, video files, documents, etc.
These NoSQL databases are divided into three types and they are explained in the following sections.
Key-value Store − These databases are designed for storing the data in key-value pairs and these databases wont have any schema. In these databases, each data value contains an indexed key and a value for that key.
Below are the examples :
BerkeleyDB, Cassandra, DynamoDB, Riak.
Column Store− In these databases, data is stored in cells grouped in

columns of data, and these columns are in turn grouped into Column families. These column families can consists of number of columns.
Below are the examples :
BigTable, HBase, and HyperTable.
Document Store – These are the databases developed on the basic idea of key-value stores where "documents"consists of more complex data. Here, every document is assigned a unique key, which is used for retrieving the document. These are designed for storing, retrieving, and managing document-oriented information, also called as semi-structured data.
Below are the examples :
CouchDB and MongoDB.

What is CouchDB?

CouchDB is an open source database and it was developed by Apache software foundation. The mainemphasis is on the simplicity of use, embracing the web. It is a NoSQL document store database.
It employes JSON, to store data (documents), java script as its query language for transforming the documents, http protocol for api for accessing the documents, query the indices with the web browser. CouchDB is a multi master application released in 2005 and it became an apache project in 2008.

Why CouchDB?

- CouchDB have an HTTP-based REST API, which assists to interact with the DB easily. And the easy structure of HTTP resources and methods (GET, PUT, DELETE) are simple to understand and make use of if.
- As we store data in the flexible document-based structure, there is no necessity to botherregarding the structure of the data.
- Users are facilitated with powerful data mapping, which permits querying, combining, and filtering of the information.
- CouchDB facilitatessimple-to-use replication, employing which you can copy, share, and synchronize the data between databases and machines.

What is Data Model?

- Database is the outermost data structure/container in CouchDB.
- Each database is a group of independent documents.
- Each document will maintains its own data and self-contained schema.
- Document metadata consists of revision information, which makes it feasible to combine the differences occurred while the databases were disconnected.
- CouchDB implements multi version concurrency control, to keep away the need to lock the database field during writes.

What are the futures of CouchDB?

Document Storage

- CouchDB is a document storage NoSQL database. Documents are the major unit of data, every field is uniquely named and consists of values of diffeent data types like text, number, Boolean, lists, etc... In these documents there is no set limit to text size or element count.
- CouchDB facilitates an API called RESTful HTTP API for the purpose of reading and updating (add, edit, delete) database documents.

Following is the sample JSON document structure followed by couchDB..

x

```
1
{
2
"field" : "value",
3
"field" : "value",
4
```

```
"field" : "value",
5
}
6
```

ACID Properties

The CouchDB file layout and commitment system features all the ACID properties.

Once the data is entered in to the disc it will not be overwritten.

Document updates (add, edit, delete) will follow Atomicity, i.e., they will be saved completely or not saved at all. The database will not have any partially saved or edited documents.

Most of these update are serialized and any number of clients can read a document without being waiting and also without being interrupted.

Compaction

Compaction is an operation to avail extra disc space for the database by discarding unused data. While performing compaction operation on a given file, a file with the extension. Compaction is created and all the active/actual data is copied (cloned) to that file, when the copying process is completed then the old file is removed. The database remains online during the compaction and all updates and reads are permitted to finish successfully.

Views

Data in CouchDB is stored in semi-structured documents that are adaptable with individual implicit structures, but it is aneasy document model for data storing and sharing. If we wish to see our data in various ways, we require a way to filter, organize and report on data that hasn’t been decomposed into tables.

For solving this problem, CouchDB facilitates a view model. Views are the method of aggregating and reporting on the documents in a database, and are built on-demand to aggregate, join and report on database documents. As views are built dynamically and don’t affect the underlying document, you may have various view representations

of the same data as you wish.

History

A brief of CouchDB's past is as shown below −

- CouchDB was written in Erlang programming language.
- It was beginned by Damien Katz in 2005.
- CouchDB became an Apache project in 2008.

The present version of CouchDB is 1.61.

{
"couchdb":"Welcome",
"uuid":"8f0d59acd0e179f5e9f0075fa1f5e804",
"version":"1.6.1",
"vendor":{
"name":"The Apache Software Foundation",
"version":"1.6.1"
}
}

What is cURL Utility?

cURL utility is a way to interact with CouchDB.

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

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

Using cURL Utility

You can access any website with cURL utility by just typing cURL alongwith the website address as shown below −

x

```
1
curl www.wisdomjobs.com/
2
```

By default, the cURL utility returns the source code of the page which we are requesting. It will display this code on the terminal window.

cURL Utility Options

cURL utility facilitatesdifferent options to work with, and you can view them in cURL utility assistance.
The below code shows some portion of cURL help.

xxxxxxxxxx

```
2
```

```
1
$ curl --help
2
```

During communication with CouchDB, certain options of cURL utility were broadlyused. Below are the brief descriptions of some major options of cURL utility with those that are used by CouchDB −

-X flag

During communication with an HTTP server, -X flag permits to specify

a request method (GET, POST, PUT etc.) for overriding the default method(GET).
During communication with an FTP server, -X flag permits to specify FTP command for overriding the default command(LIST).

-H

This flag will specify a custom header to pass on to the server. For instance, for sending a file to the server, we need to use the "content type"which defines the type of content of the file that we are about to upload.

-d flag

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

Example

For example if there is a website and you wish to login into it or send some data to the website with –d flag of cURL utility as shown here.

xxxxxxxxxx

2

```
1
curl -X PUT http://mywebsite.com/login.html -d userid=001 -d
password=wisdomjobs
2
```

It sends a post chunk that looks like"userid=001&password=wisdomjobs". Similarly, you can also send documents (JSON) with -d flag.

-o flag

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

Example

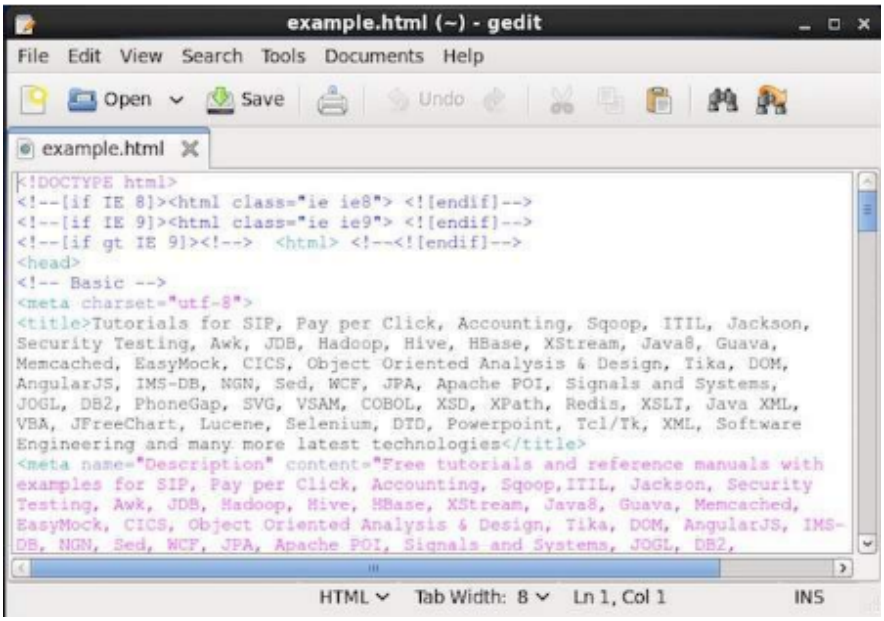
The below example shows the usage of -o flag of cURL utility.

xxxxxxxxxx

6

```
$ curl -o example.html www.wisdomjobs.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 wisdomjobs.com, creates a file with the name example.com and saves the output in the file named example.html.
Below is the snapshot of the **example.html.**



-O

This flag is same as to –o, the only variation 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 below example shows the usage of -O flag of cURL utility.

xxxxxxxxxx


```
1
$ curl -O www.wisdomjobs.com/index.htm
2
% Total % Received % Xferd Average Speed Time Time Time Current
3
Dload Upload Total Spent Left
4
Speed
5
100 81285 0 81285 0 0 49794 0 --:--:-- 0:00:01 --:--:--
```

```
6
60077
7
```

It will create a new file with the name index.htm and will save the source code of the index page of WisdomJobs.com in the file.

Hello CouchDB

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

xxxxxxxxxx

```
2
```

```
1
curl http://127.0.0.1:5984/
2
```

This will give you a JSON document as shown below where CouchDB will specify the details needed such as version number, name of the vendor, and version of the software.

xxxxxxxxxx

```
11
```

```
$ 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, just by sending a get request including the string "_all_dbs string ". Below is the syntax to get the list of all databases in CouchDB.

```
xxxxxxxxxx
```

```
1
curl -X GET http://127.0.0.1:5984/_all_dbs
2
```

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

xxxxxxxxxx

```
3
```

```
1
$ curl -X GET http://127.0.0.1:5984/_all_dbs
2
[ "_replicator" , "_users" ]
3
```

Creating a Database

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

xxxxxxxxxx

```
2
```

```
1
$ curl -X PUT http://127.0.0.1:5984/database_name
2
```

Example

As an example, with the above given syntax create a database with name my_database as shown here.

xxxxxxxxxx

3

```
1
$ curl -X PUT http://127.0.0.1:5984/my_database
2
{"ok":true}
3
```

Verification

We should verify whether the database is created, by listing out all the databases as shown here. In the list you can notice the name of database that was created newly, "my_database" .

xxxxxxxxxx

3

```
1
$ curl -X GET http://127.0.0.1:5984/_all_dbs
2
[ "_replicator " , "_users" , "my_database" ]
3
```

Getting Database Info

You can get the informationrelated to the database with the get request including with the database name. Below 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 see the information related to your database as a response.

xxxxxxxxxx


```
1
$ curl -X GET http://127.0.0.1:5984/my_database
2
{
3
"db_name" : "my_database",
4
"doc_count" : 0,
5
"doc_del_count" : 0,
6
"update_seq" : 0,
7
"purge_seq" : 0,
8
"compact_running" : false,
9
"disk_size" : 79,
```

```
10
"data_size" : 0,
11
"instance_start_time" : "1423628520835029",
12
"disk_format_version" : 6,
13
"committed_update_seq" : 0
14
}
15
```

Futon

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

Databases –

- Creates databases.
- Destroys databases.

Documents –

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

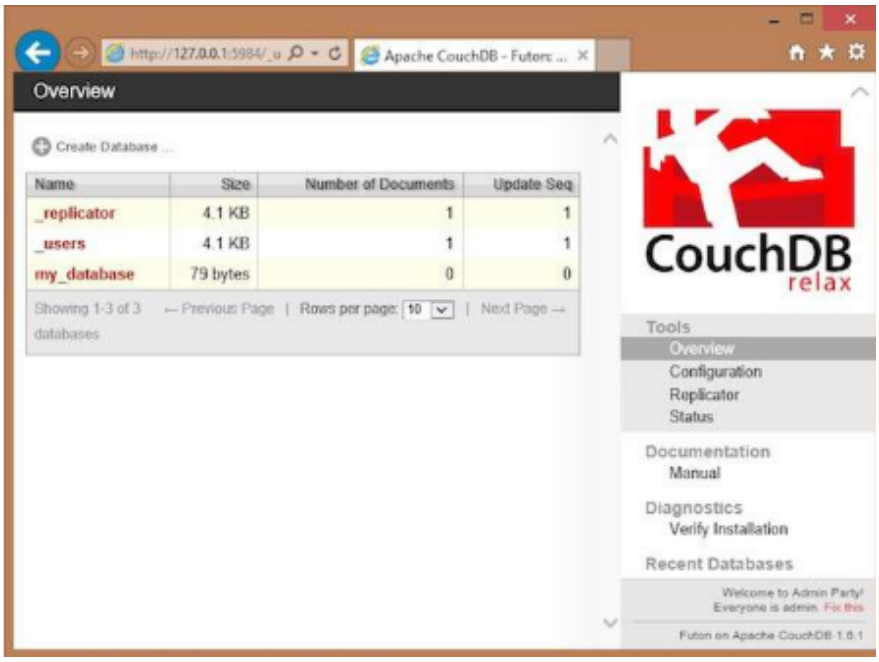
Starting Futon

Ensure that CouchDB is running and then open the below url in browser –

xxxxxxxxxx

```
1
http://127.0.0.1:5984/_utils/
2
```

If you open this url, it displays the Futon home page as shown in the below screenshot−



On the left hand side of this page you can notice the list of all the databases that are available in CouchDB. In this explanation, we have a database named my_database, including the system defined databases _replicator and _user.

On the right hand side you can see the below −

- **Tools** − In this section you can find Configuration for configuring CouchDB, Replicator for performing replications, and Status for verifying the status of CouchDB and recent changes done on CouchDB.
- **Documentation** − This section consists of the entire documentation for the latest version of CouchDB.
- **Diagnostics** − Below this you can verify the installation of

CouchDB.

- **Recent Databases** – Below this you can find the names of recently added databases.

« Previous Topics

Couchdb Installation

Couchdb Introduction

Next Topics »

Couchdb Http Api

Couchdb Creating A Database

Couchdb Deleting A Database

What is CouchDB - HTTP API?

With HTTP request headers, you can interact with CouchDB. Using these requests we can retrieve data from the database, store data in to the database in the form of documents, and we can see as well as format we can the documents that are stored in a database.

HTTP Request Formats

While interacting with the database we will employvarious request formats such as get, head, post, put, delete, and copy. For all operations in CouchDB, both the input datastructures and the output data structures would be in the form of JavaScript Object Notation (JSON) object.

Below are the various request formats of HTTP Protocol employed to interact with CouchDB.

- **GET** – This format is employedto get a specific item. To get different items, you have to send specific url patterns. In CouchDBwith this GET request, we can get static items, database documents and configuration, and statistical information in the form of JSON documents (in most cases).
- **HEAD** – The HEAD method is employedto get the HTTP header of a GET request without the body of the response.
- **POST** – Post request is employed to upload data. In

CouchDBwith POST request, you can set values, upload documents, set document values, and can also beginspecific administration commands.

- **PUT** − With PUT request, you can create new objects, databases, documents, views and design documents.
- **DELETE** − With DELETE request, you can delete documents, views, and design documents.
- **COPY** − With COPY method, you can copy documents and objects.

HTTP Request Headers

HTTP headers must be supplied to get the correct format and encoding. during sending the request to the CouchDB server, you can send Http request headers including with the request. Below are the different Http request headers.

- **Content-type** − This Header is employed for specifying the content type of the data that we provide to the server along with the request. Mostly the type of the content we send including with the request will be MIME type or JSON (application/json). With Content-type on a request is highly recommended.
- **Accept** − Accept header is employedfor specifying the server, the list of data types that client can understand, so that the server will send its response with those data types. Generally here, you can send the list of MIME data types the client accepts, partitioned by colons.

Even though, with Accept in queries of CouchDB is not needed, it is highly recommended to make sure that the data returned can be processed by the client.

Response Headers

Response headers are the headers of the response that is sent by the server. These headers give information regarding the content that is send by the server as response.

- **Content-type** − This header specifies the MIME type of the data returned by the server. For most request, the returned MIME type is text/plain.

- **Cache-control** − This header suggests the client about treating the information sent by the server. CouchDB mostly returns the must-revalidate, which indicates that the information should be revalidated if possible.
- **Content-length** − This header returns the length of the content sent by the server, in bytes.
- **Etag** − This header is used to show the revision for a document, or a view.

Status Codes

Below is the tabular form of the status code that are sent by the http header and the explanation of it.

200 – OK	This status will be issued when a request completed successfully.
201 – Created	This status will be issued when a document is created.
202 – Accepted	This status will be issued when a request is accepted.
404 – Not Found	This status will be issued when the server is unable to find the requested content.
405 – Resource Not Allowed	This status is issued when the HTTP request type used is invalid.
409 – Conflict	This status is issued whenever there is any update conflict.
415 – Bad Content Type	This status indicated that the requested content type is not supported by the server.
500 – Internal Server Error	This status is issued whenever the data sent in the request is invalid.

HTTP URL Paths

There are someurl paths with which, you can communicate with the database directly. Below is the tabular format of such url paths.

URL	Operation
PUT /db	This url is used to create a new database.
GET /db	This url is used to get the information about the existing database.
PUT /db/document	This url is used to create a document/ update an existing document.
GET /db/document	This url is used to get the document.
DELETE /db/ document	This url is used to delete the specified document from the specified database.
GET /db/_design/ design-doc	This url is used to get the definition of a design document.
GET /db/_design/ designdoc/_view/ view-name	This url is used to access the view, view-name from the design document from the specified database.

« Previous Topics

Couchdb Curl & Futon

Couchdb Installation

Couchdb Introduction

Next Topics »

Couchdb Creating A Database

Couchdb Deleting A Database

Couchdb Creating A Document

What is CouchDB - Creating a Database?

Database is the outermost data structure in CouchDBthe place where

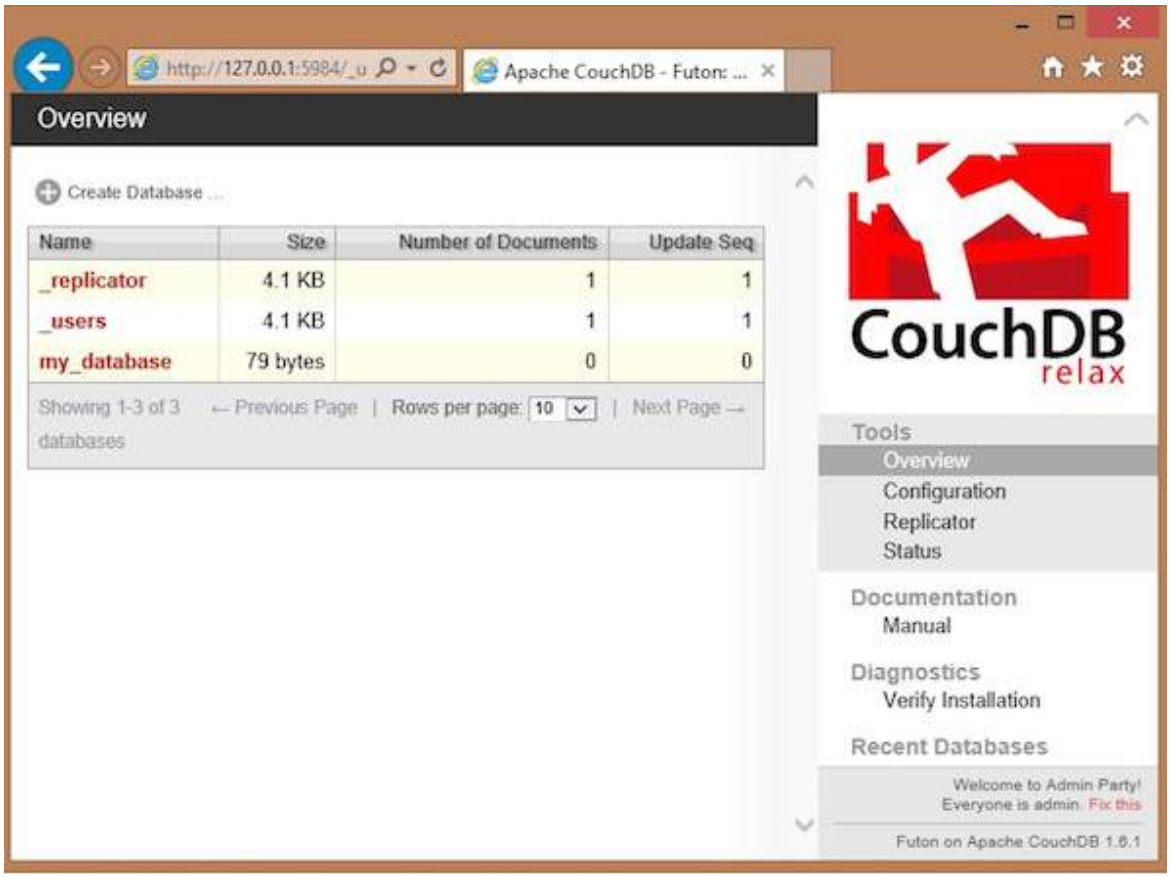
your documents will be stored. You can create these databases withcURL utility facilitated by CouchDB, and also the Futon the web interface of CouchDB.

Creating a Database using cURL Utility

You can create a database in CouchDB by sending an HTTP request to the server with PUT method via thecURL utility. Below is the syntax for creating a database –

x

```
1
$ curl -X PUT http://127.0.0.1:5984/database_name
2
```



With −X we can mention HTTP custom request method to be emoloyed. In this scenario, we are employing PUT method. When we employ the PUT operation/method, the content of the urlmentions the object name we are creating with HTTP request. Here we need to send thedatabase name with put request in the urlfor creating a database.

Example

With the above given syntax if you wish to create a database with name my_database, you can create database as specified below. As a response the server will in return you a JSON document using content "ok" : true saying that the operation was successful.

xxxxxxxxxx

```
curl -X PUT http://127.0.0.1:5984/my_database
2
{
3
 "ok":true
4
}
5
```

Verification

Verify if the database is properly created, by listing out all the databases as displayed below. Here you can notice the name of thedatabase that was newly created, " my_database " in the list.

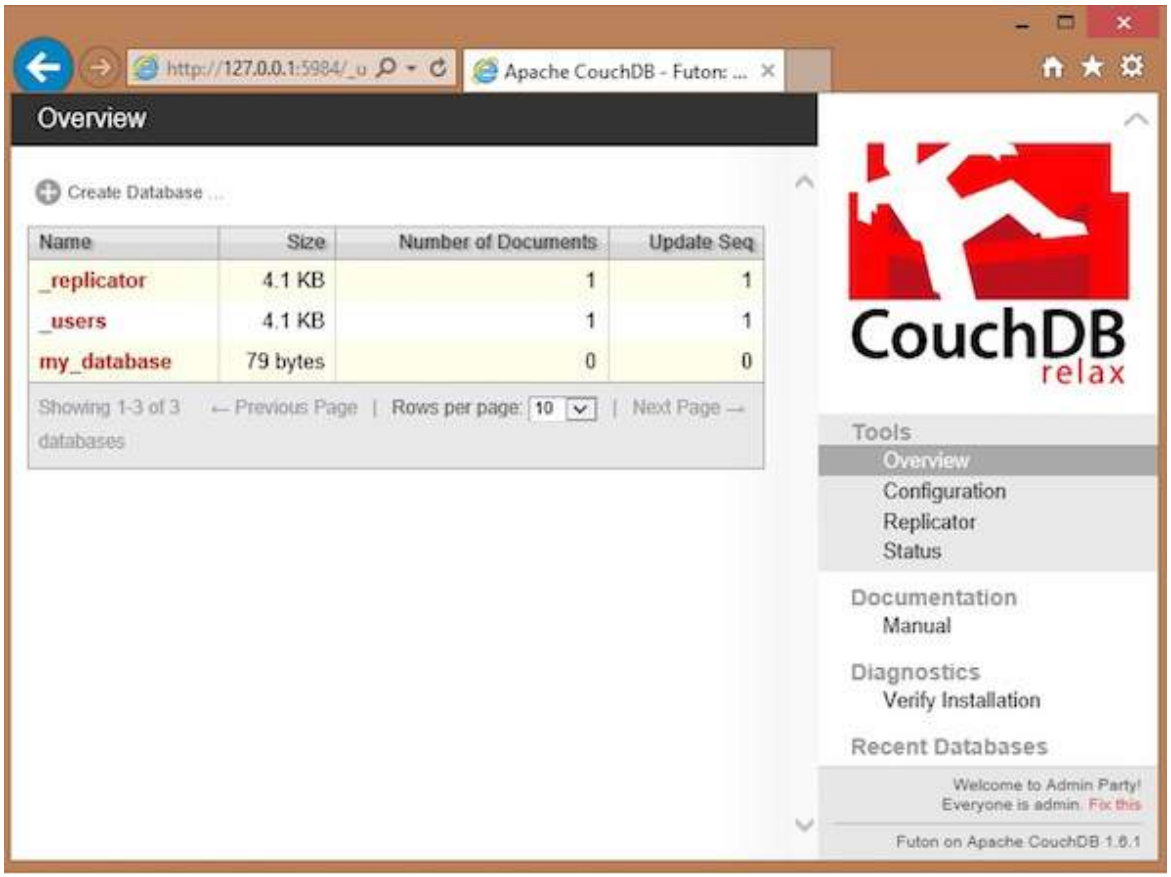
xxxxxxxxxx

3

```
1
$ curl -X GET http://127.0.0.1:5984/_all_dbs
2
[ "_replicator " , " _users " , " my_database " ]
3
```

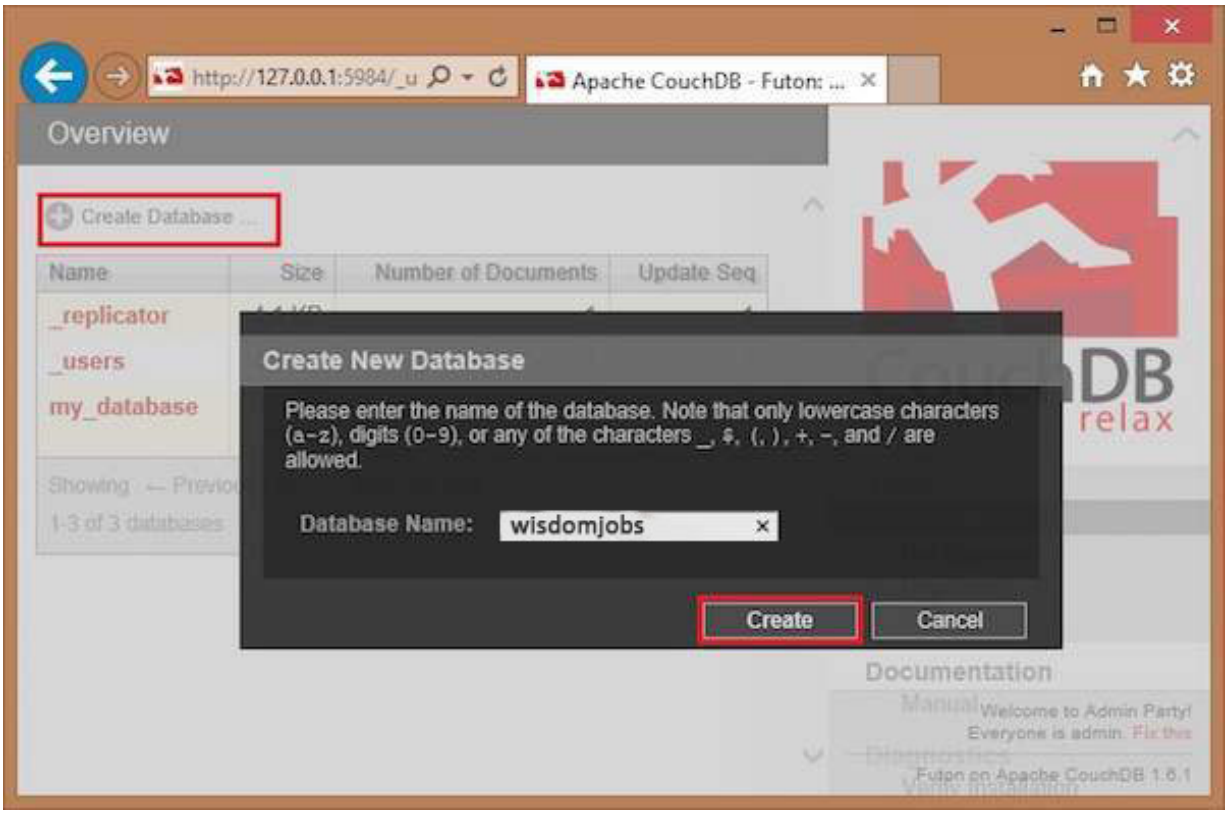
Creating a Database with Futon

For creating a database open the http://127.0.0.1:5984/_utils/. You can get an Overview/index page of CouchDB as shown in below screenshot.



In this page, you can view all the list of databases in CouchDB, an option button Create Database on the LHS.

Now click on the create database link. You can view a popup window Create New Databases asking for the database name for the new database. Select any name following the specified criteria. Here we are creating other database with the name wisdomjobs. Click on the create button as shown in the below screenshot.



« Previous Topics

Couchdb Http Api

Couchdb Curl & Futon

Couchdb Installation

Next Topics »

Couchdb Deleting A Database

Couchdb Creating A Document

Couchdb Updating A Document

How to Delete a Database with cURL Utility?

One can delete a database in CouchDB simply sending a request to the server with DELETE method via cURL utility. Below is the syntax for creating a database –

x

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

with −X we can mention a custom request method of HTTP we are using, while interacting with the HTTP server. In this scenario, we are employing the DELETE method. Send the url to the server by mentioning the database that need to be deleted in it.

Example

Let us assume there exists a database named my_database2 in CouchDB. Execute the command as specified below. Server must respond you with a JSON document with "ok" : true saying the operation was successful.

xxxxxxxxxx

```
$ curl -X DELETE http://127.0.0.1:5984/my_database2
{
 "ok" : true
}
```

Verification

Verify of the database is deleted by listing out all the databases as displayed below. Here you can notice the name of the database that was deleted, "my_database" will not be displayed in the list.

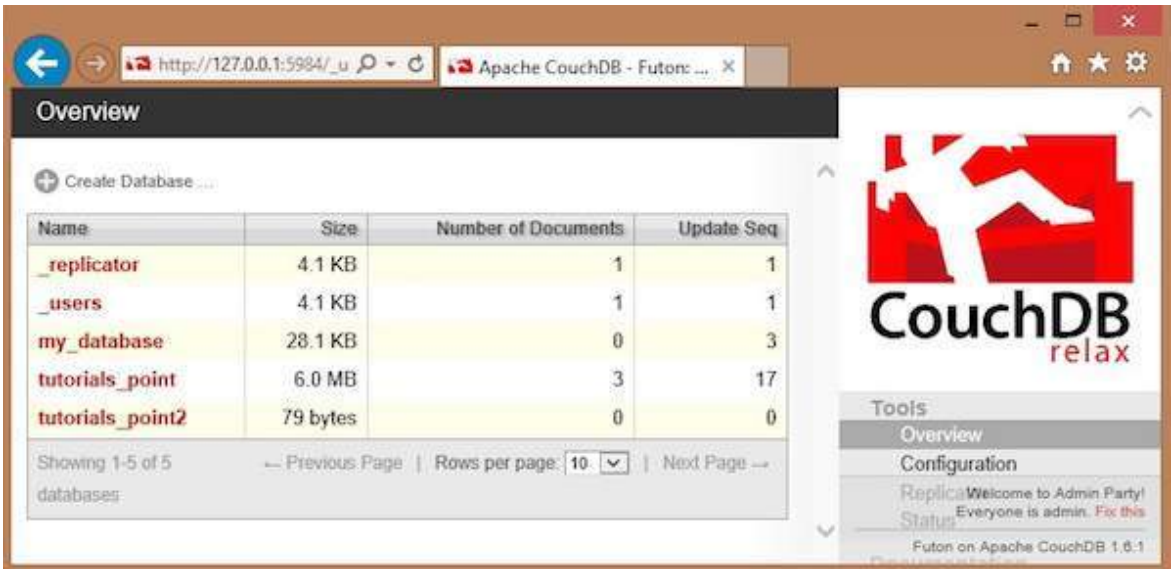
xxxxxxxxxx

3

```
1
$ curl -X GET http://127.0.0.1:5984/_all_dbs
2
[ "_replicator " , " _users " ]
3
```

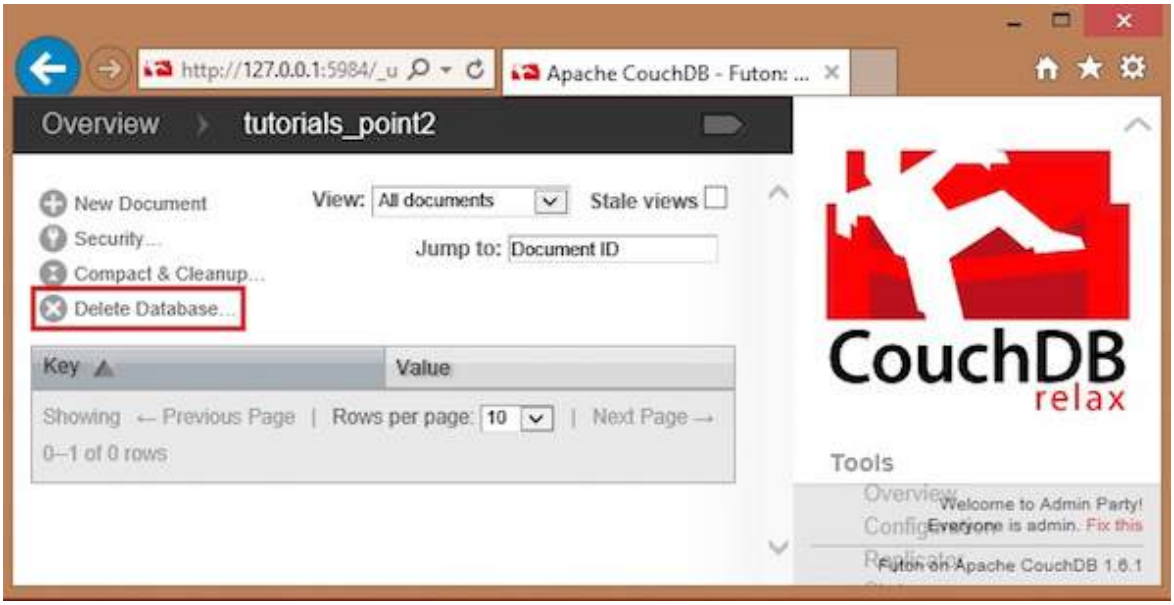
Deleting a Database with Futon

For deleting a database, open the url shown below http:// 127.0.0.1:5984/_utils/ which will take you to an Overview/index page of CouchDB as shown below.

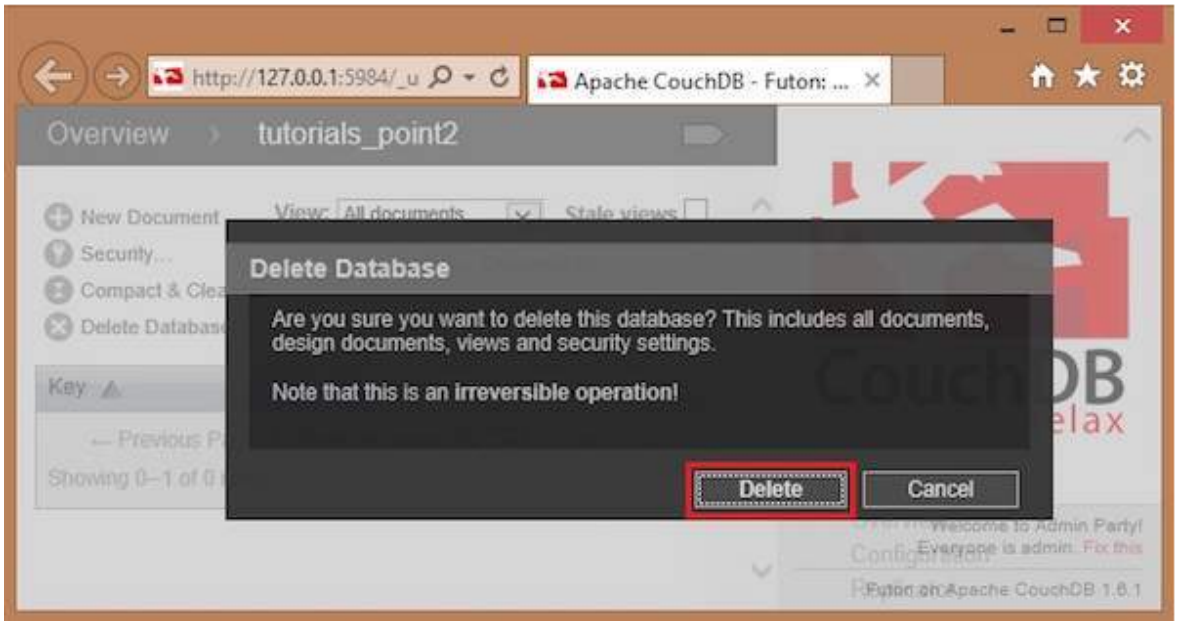


Here you can see three user created databases. Let us delete the database that was named WisdomJobs. For deleting a database,

choose one from the list of databases, and click on it, which will take you to the overview page of the choosen database where you can view the different operations on databases. The below screenshot displays the same −



Among them you can see Delete Database option. By clicking on it you can see a popup window, to confirm Click on delete, to delete the database that was selected.



« Previous Topics

Couchdb Creating A Database

Couchdb Http Api

Couchdb Curl & Futon

Next Topics »

Couchdb Creating A Document

Couchdb Updating A Document

Couchdb Deleting A Document

What is CouchDB - Creating a Document?

Documents are CouchDB's central data structure. Database Contents will be stored in the form of Documents rather than of tables. You can create these documents with cURL utility facilitated by CouchDB, and also the Futon. This chapter explains the ways for creating a document in a database.

Each document in CouchDB will have a unique ID. You can select your own ID that must be in the form of a string. Usually, UUID (Universally Unique IDentifier) is employed, which are random numbers that will have very least chances of creating a duplicate. These are preferred in order to avert collisions.

How to create a Document withcURL Utility?

One can create a document in CouchDB by sending an HTTP request to the server with PUT method viacURL utility. Belowis the syntax for creating a document.

x

```
1
$ curl -X PUT http://127.0.0.1:5984/database name/"id" -d
' { document} '
2
```

With −X, we can mention a custom request method of HTTP we are employing, while interacting with the HTTP server. In this scenario, we are employing PUT method. When we employ the PUT method, the content of the urlmentions the object name we are creating with the HTTP request. Here we have to send the below things −

- The name of the database name in which we are creating the document.
- The document id.
- The data of the document. −d option is employedfor sending the data/document via HTTP request. While writing a document just enter your Field-Value pairs partitioned by colon, within flower brackets as shown here–

xxxxxxxxxx

6

```
1
{
2
Name : Raju
3
age : 23
4
Designation : Designer
5
}
6
```

Example

With the given syntax if you wish to create a document with id 001 in a database with the name my_database, you can create it as shown here.

xxxxxxxxxx

4

```
$ curl -X PUT http://127.0.0.1:5984/my_database/"001" -d
'{ " Name " : " Raju " , " age " :" 23 " , " Designation " : " Designer " }'
{"ok":true,"id":"001","rev":"1-1c2fae390fa5475d9b809301bbf3f25e"}
```

The response of CouchDB to this request consists of three fields −

- "ok", indicating the operation was successful.
- "id", which stores the id of the document and
- "rev", this indicates the revision id. A _rev value will be generated by CouchDB each time when you modify or alter the document. If you wish to update or delete a document, CouchDB expects you to include the _rev field of the revision you want to alter. When CouchDB accepts the alteration, it will then generate a new revision number. This mechanism will make sure concurrency control.

Verification

If you wish to see the created document you can get it with the document as shown here.

xxxxxxxxxx

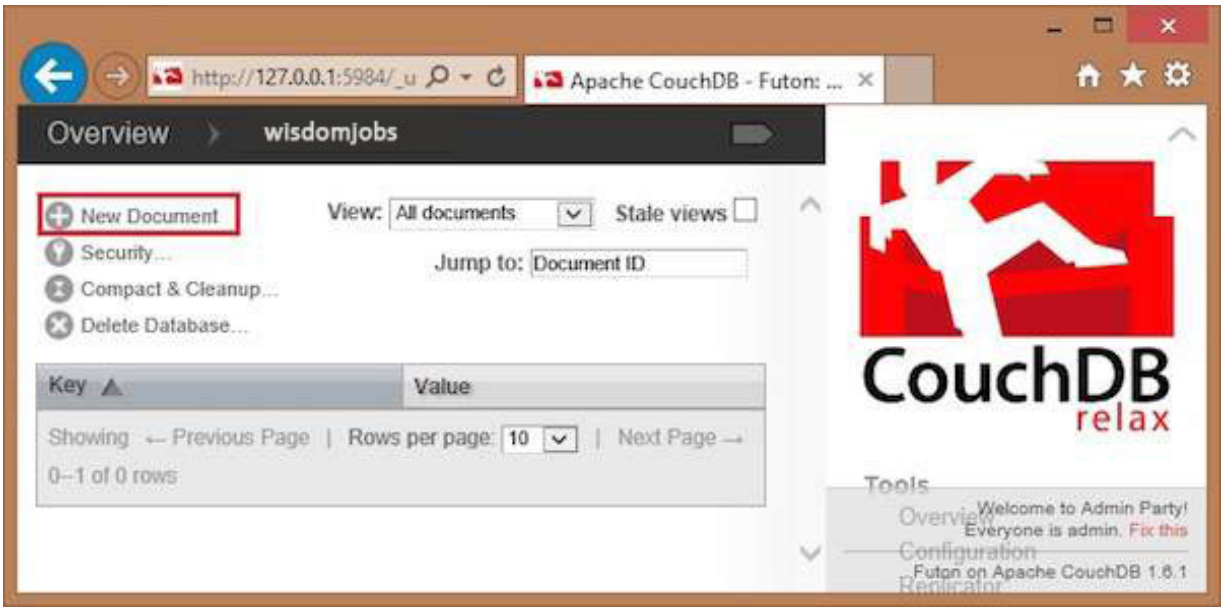
9

```
$ curl -X GET http://127.0.0.1:5984/my_database/001
{
 "_id": "001",
 "_rev": "1-3fcc78daac7a90803f0a5e383f4f1e1e",
 "Name": "Raju",
 "age": 23,
 "Designation": "Designer"
}
```

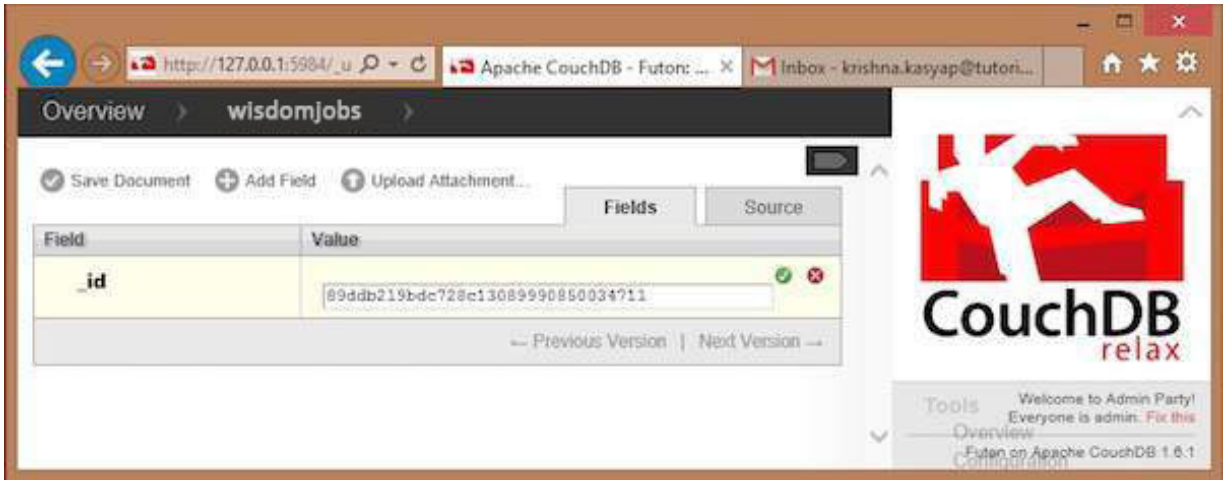
How to Create a Document using Futon?

For Creating a document open the mentioned url http://127.0.0.1:5984/_utils/ to get an Overview/index page of CouchDB as shown in the below screenshot.

Choose the database in which you wish to create the document. Open the Overview page of the database and choose New Document option as shown in the below screenshot.



When you choose the New Document option, CouchDB creates a new database document, byassigning it a new id. You can edit the value of the id and you can assign your own value in the form of a string. In the below illustration, we have created a new document with an id 001.



In this page, you can notice three options – save Document, Add Field and Upload Attachment.

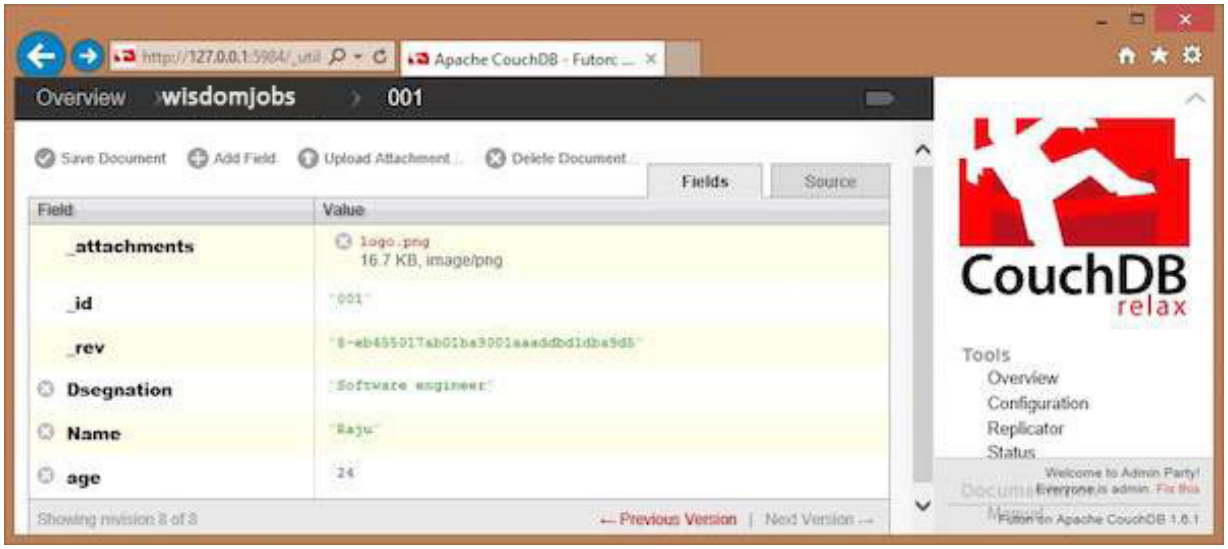
How to Add Field to the Document?

In order to add field to the document click on the Add Field option.Soon after creating a database, you can add a field to it with this option. Clicking on it will get show you a pair of text boxes,

namely, Field, value. You can edit these values by clicking on them. Edit those values and type the Field-Value pair which you desire. Click on the green button for saving these values.
In the below illustration, we have created three fields Name, age and, Designation of the employee.

How to Save Document?

You can save the modifications made to the document just by clicking on this option. After saving, a new id _rev will be generated as shown in the below screenshot



« Previous Topics

Couchdb Deleting A Database

Couchdb Creating A Database

Couchdb Http Api

Next Topics »

Couchdb Updating A Document

Couchdb Deleting A Document

Couchdb Attach Files

What is CouchBD - Updating Documents with cURL?

One can update a document in CouchDB just by sending an HTTP request to the server with via PUT method through cURL utility. Below is the syntax for updating a document.

x

```
1
curl -X PUT http://127.0.0.1:5984/database_name/document_id/ -d
'{ "field" : "value", "_rev" : "revision id" }'
2
```

Example

Assume there is a document with id 001 in the database with the namemy_database.

First of all, get the revision id of the document that requires to be updated. You can find out the _rev of the document in the document itself, as shown here–

xxxxxxxxxx

```
$ curl -X GET http://127.0.0.1:5984/my_database/001
{
 "_id" : "001",
 "_rev" : "2-04d8eac1680d237ca25b68b36b8899d3 " ,
 "age" : "23"
}
```

Make use of revision id _rev from the document. In the given example, we are updating the age from 23 to 24.

xxxxxxxxxx

4

```
$ curl -X PUT http://127.0.0.1:5984/my_database/001/ -d
' { " age " : " 24 " , " _rev " : " 1-1c2fae390fa5475d9b809301bbf3f25e " } '
{ " ok " : true , " id " : " 001 " , " rev " : "
2-04d8eac1680d237ca25b68b36b8899d3 " }
```

Verification
In order to verify the document, get the document again with GET request as shown here–

xxxxxxxxxx
7

```
$ curl -X GET http://127.0.0.1:5984/my_database/001
{
" _id " : " 001 ",
" _rev " : " 2-04d8eac1680d237ca25b68b36b8899d3 " ,
" age " : " 23 "
}
```

Note –
Below are some major points to be noted during updating a document.

- The URL we send in the request consisting of database name and the document id.
- Updating an existing document is similar to as updating the complete document. You cannot add a field to a document

that is existing. You can only write an entirely new version of the document into the database with the same document ID.

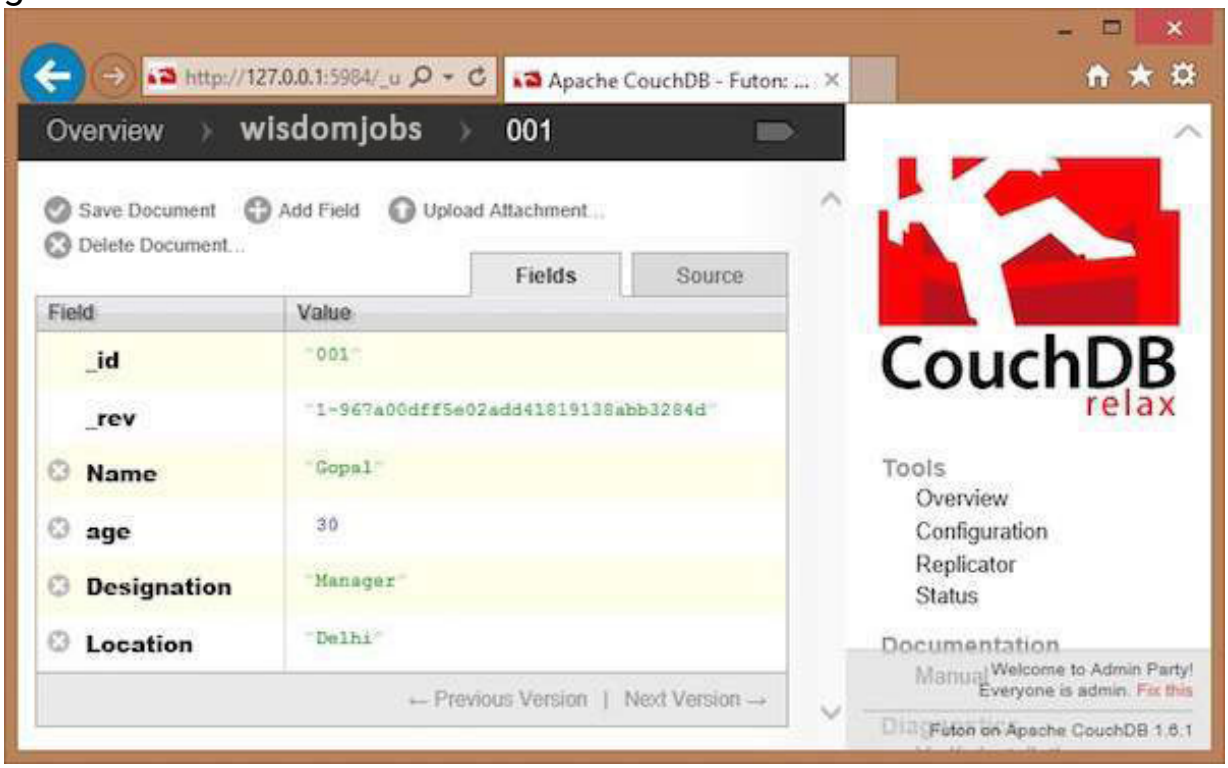
- We need to supply the revision number as a part of the JSON request.
- In return JSON consists of the success message, the ID of the document that was being updated, and the new revision information. If you wish to update the new version of the document, you requireto quote this latest revision number.

Updating Documents with Futon

For deleting a document open the url http://127.0.0.1:5984/_utils/ to get an Overview/index page of CouchDB as shown in the below screenshot.

Choose the database in which the document to be updated is present and click on it. Here we are updating a document in the database with the name wisdomjobs. You will get the documentslist in the database as shown in the below screenshot.

Choose a document that you wish to update and click on it. You will get the contents of the documents as shown in the below screenshot.



Here, for updating the location from Delhi to Hyderabad, click on the text box, and edit the field, and click the green button for saving the modifications as shown in the below screenshot

« Previous Topics

Couchdb Creating A Document

Couchdb Deleting A Database

Couchdb Creating A Database

Next Topics »

Couchdb Deleting A Document

Couchdb Attach Files

What is CouchDB - Deleting a Document using cURL Utility?

One can delete a document in CouchDB just by sending an HTTP request to the server with DELETE method viacURL utility. Below is the syntax for deleting a document.

x

```
1
curl -X DELETE http://127.0.0.1:5984/database_name/database_id?
_rev
2
```

With −X, we can mention a custom request method of HTTP we are employing, duringinteracting with the HTTP server. In this situation, we are employing Delete method. For deleting a database / database_name/database_id/ is not sufficient. You need to pass the recent revision id viathe url. To specify attributes of any data structure "?" is employed.

Example

Assume there is a document in database with the name my_database with document id 001. For deleinge this document, you need to get the rev id of the document. Get the document data as shown here.

xxxxxxxxxx

7

```
1
$ curl -X GET http://127.0.0.1:5984/my_database/001
2
{
3
" _id " : " 001 ",
4
" _rev " : " 2-04d8eac1680d237ca25b68b36b8899d3 " ,
5
" age " : " 23 "
6
}
```

7

Now mention the revision id of the document that needs to be deleted, id of the document, and database name the document belongs to, as shown here–

xxxxxxxxxx

4

```
1
$ curl -X DELETE http://127.0.0.1:5984/my_database/001?rev=1-
2
3fcc78daac7a90803f0a5e383f4f1e1e
3
{"ok":true,"id":"001","rev":"2-3a561d56de1ce3305d693bd15630bf96"}
4
```

Verification

In order to verify if the document is deleted, try to bring the document by employing the GET method. As you are bringing a deleted document, this will display you an error message as shown here–

xxxxxxxxxx

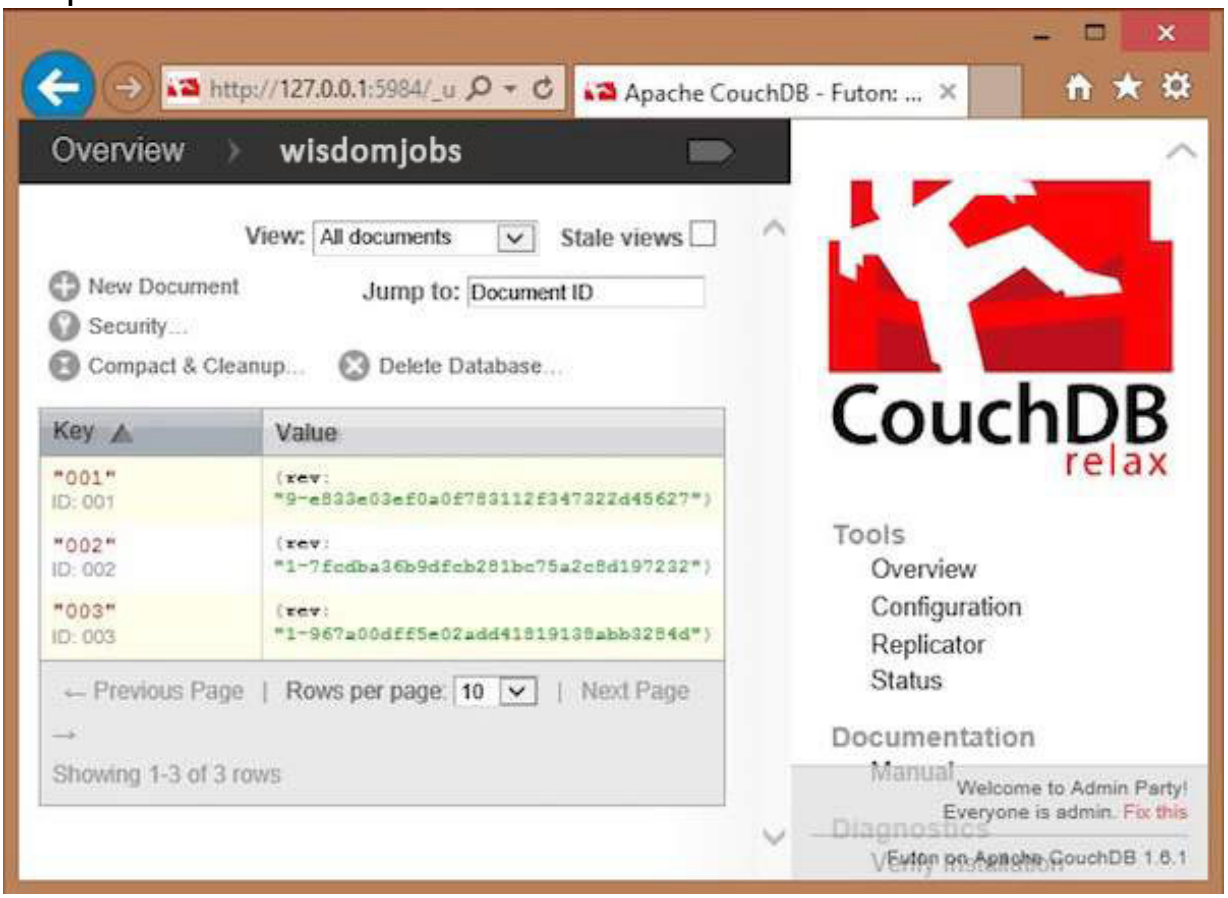
3

```
1
```

```
$ curl -X GET http://127.0.0.1:5984/my_database/001
2
{"error":"not_found","reason":"deleted"}
3
```

Deleting a Document with Futon

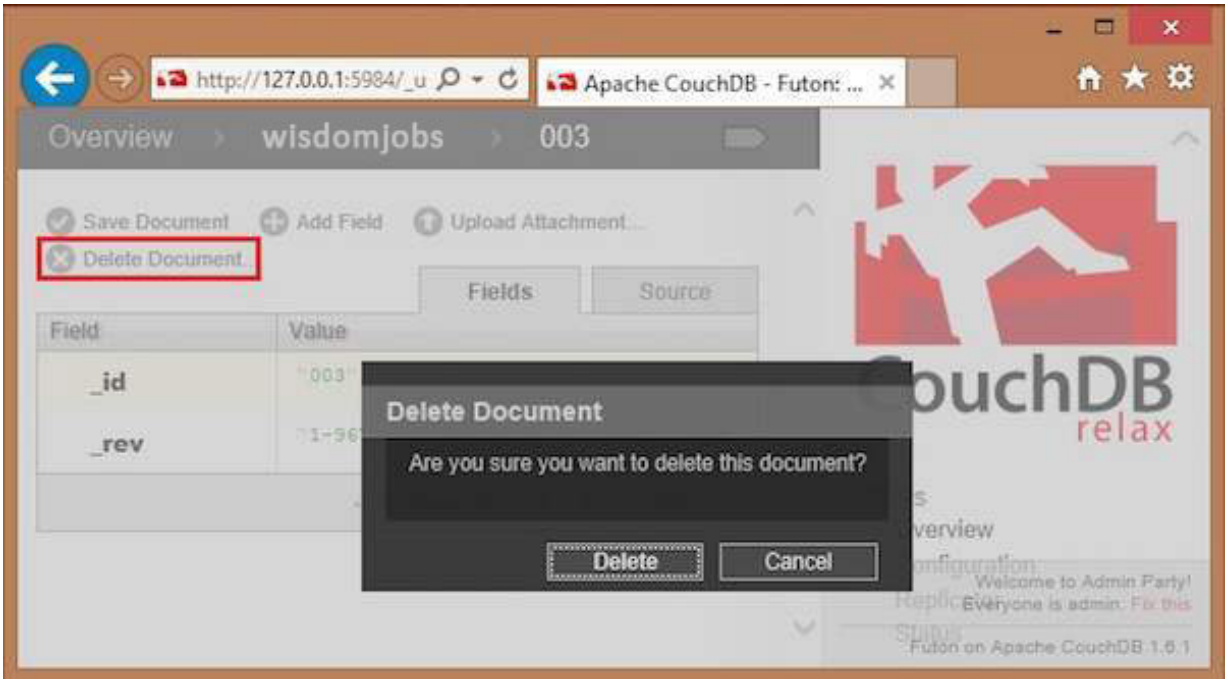
First of all, verify the documents in the database. Below is the snapshot of the database with the name WisdomJobs.



Here you can notice, the database contains three documents. For deleting any of the documents say 003, do the below things −

- Click on the document, it will display a page showing the contents of selected document in the form of field-value pairs.
- This page also consists of four options. They are Save Document, Add Field, Upload Attachment, Delete Document.

- Click on Delete Document option.
- You will obtain a dialog box saying "Are you sure you want to delete this document?" Click on delete, for deleting the document.



« Previous Topics

Couchdb Updating A Document

Couchdb Creating A Document

Couchdb Deleting A Database

How to Attach Files with cURL?

You can also attach files to CouchDB, similarly you do in email. The file consists of metadata such as name and includes its MIME type, and the number of bytes the attachment consists of. For attaching files to a document you need to send PUT request to the server. Below is the syntax for attaching files to the document –

x

```
1
$ curl -vX PUT http://127.0.0.1:5984/database_name/database_id
2
/filename?rev=document rev_id --data-binary @filename -H "Content-
Type:
3
type of the content"
4
```

The request has different options that are explained in the following:

- **--data-binary@** - This option tells cURL to read a file's contents into the HTTP request body.
- **-H -** This option is used to specify the content type of the file we are about to upload.

Example

Let us attach a file with the name boy.jpg, to the document with id 001, in the database with the namemy_database by sending PUT request to CouchDB. In prior to that, you need to bring the data of the document with id 001 to have its present rev id as shown here.

xxxxxxxxxx

```
10
```

```
1
$ curl -X GET http://127.0.0.1:5984/my_database/001
2
{
```

3
"_id": "001",
4
"_rev": "1-967a00dff5e02add41819138abb3284d"
5
}
6
Now with the _rev value, send the PUT request to the CouchDB server as shown below.
7
$ curl -vX PUT http://127.0.0.1:5984/my_database/001/boy.jpg?rev=1-
8
967a00dff5e02add41819138abb3284d --data-binary @boy.jpg -H "ContentType:
9
image/jpg"
10

Verification

To verify if the attachment has been uploaded, bring the document content as shown here−

xxxxxxxxxx
15

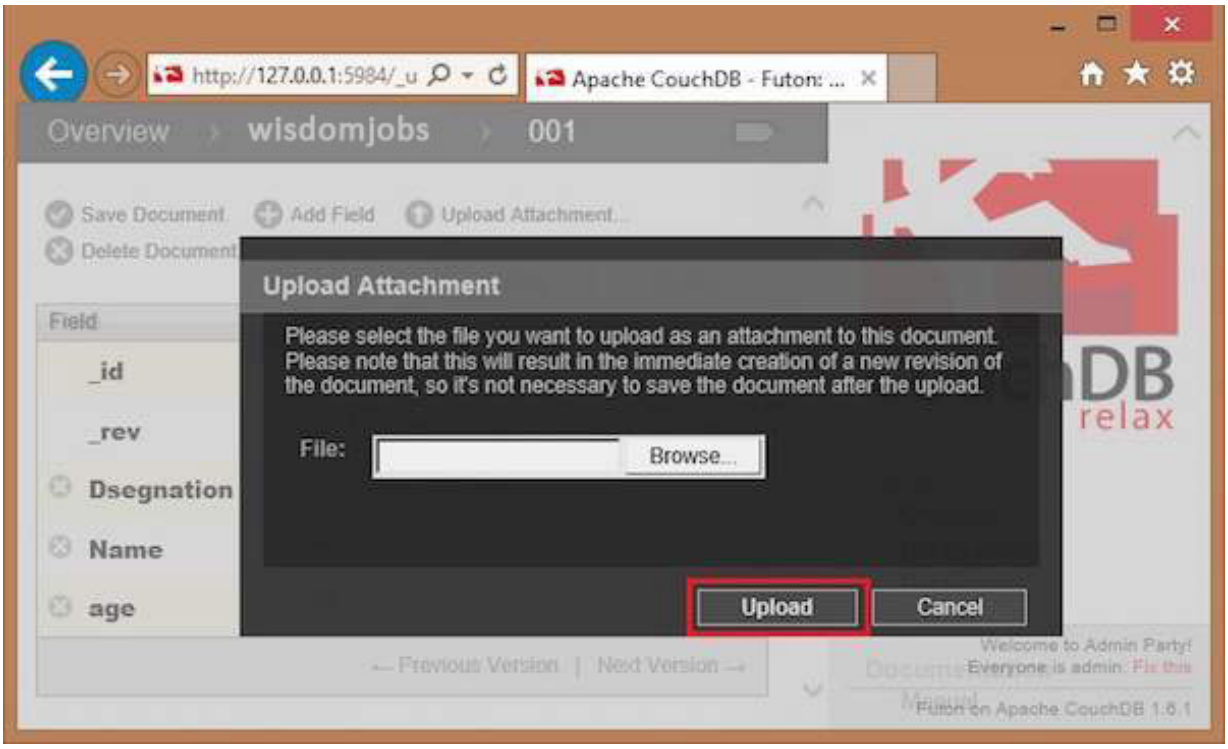
1
$ curl -X GET http://127.0.0.1:5984/my_database/001
2
{
3

```
 "_id": "001",
 "_rev": "2-4705a219cdcca7c72aac4f623f5c46a8",
 "_attachments": {
   "boy.jpg": {
    "content_type": "image/jpg",
    "revpos": 2,
    "digest": "md5-9Swz8jvmga5mfBIsmCxCtQ==",
    "length": 91408,
    "stub": true
   }
 }
}
```

How to Attach Files with Futon?

Upload Attachment

With this option, you can upload a new attachment like file, image, or document, to the database. In order to do this, click on the Upload Attachment button. A dialog box will display where you can choose the file that needs to be uploaded. Choose the file and then click on the Upload button.



The file uploaded will be shown below _attachments field. Later you can view the file by clicking on it.