PouchDB

tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com

About this Tutorial

PouchDB is an open source **in-browser database API** written in JavaScript. It is modelled after CouchDB – a NoSQL database that powers npm. Using this API, we can build applications that work offline and online. PouchDB uses WebSQL and IndexedDB internally to store the data.

This tutorial discusses the basics of PouchDB along with relevant examples for easy understanding.

Audience

This tutorial has been prepared for beginners to help them understand the basic concepts of PouchDB. It will aid you to build applications which will work offline and online alike using PouchDB and CouchDB.

Prerequisites

The reader should have a basic knowledge of databases. It would be better to have a good command on programming languages, which are compatible with node.js such as JavaScript and CoffeeScript.

Copyright & Disclaimer

Table of Contents

About this Tutorial i

Audience i

Prerequisites i

Copyright & Disclaimer i

Table of Contents ii

1. POUCHDB – OVERVIEW 1

What is PouchDB? 1

How Does it Work? 1

Features of PouchDB 1

Advantages of PouchDB 1

Browsers that Support PouchDB 2

2. POUCHDB – ENVIRONMENT 3

Installing PouchDB 3

Installing Pouch Using Node.js 4

Downloading CouchDB 4

Installing CouchDB 6

3. POUCHDB – CREATE DATABASE 7

4. POUCHDB – DATABASE INFO 8

Remote Database Info 9

5. POUCHDB – DELETE DATABASE 11

Deleting a Remote Database 12

6. POUCHDB – CREATE DOCUMENT 14

Inserting a Document in a Remote Database 15

7. POUCHDB – READ DOCUMENT 18

Reading a Document from a Remote Database 19

8. POUCHDB – UPDATE DOCUMENT 22

Updating a Document in a Remote Database 24

9. POUCHDB – DELETE DOCUMENT 27

Deleting a Document from a Remote Database 28

10. POUCHDB – CREATE BATCH 31

Inserting a Batch in a Remote Database 32

11. POUCHDB – FETCH BATCH 35

Reading a Batch from a Remote Database 37

12. POUCHDB – UPDATE BATCH 40

Updating Batch from a Remote Database 42

13. POUCHDB – DELETE BATCH 47

Deleting Batch from a Remote Database 49

14. POUCHDB – ADDING ATTACHMENT 52

Adding Attachment to an Existing Document 54

Adding Attachment to a Remote Document 55

15. POUCHDB – RETRIEVING ATTACHMENT 59

Retrieving Attachment from a Remote Document 60

16. POUCHDB – DELETING ATTACHMENT 63

Removing Attachment from a Remote Document 65

17. POUCHDB – REPLICATION 69

Replicating LocalDB to CouchDB **69**

Replicating CouchDB to PouchDB **71**

18. POUCHDB – SYNCHRONIZATION 74

19. POUCHDB – MISCELLANEOUS 79

Compaction **79**

BulkGet Method **79**

1. PouchDB – Overview

This chapter provides a brief introduction to PouchDB along with its features and how it works.

What is PouchDB?

PouchDB is an open source **in-browser database API** written in JavaScript. It is modelled after Couch DB – a NoSQL database. Using this API, we can build applications that work offline and online. It internally uses WebSQL and IndexedDB to store data.

How Does it Work?

In PouchDB, when the application is offline, the data is stored locally using WebSQL and IndexedDB in the browser. When the application is back online, it is synchronized with CouchDB and compatible servers.

Using PouchDB, you can communicate with both local and remote databases seamlessly without noticing any difference.

Features of PouchDB

Following are the features of PouchDB -

- **Cross Browser:** The API provided by PouchDB works the same in every environment, therefore, we can run a PouchDB application in various browsers.
- **Light Weight:** PouchDB is a very light-weight API, it is also included easily just using a script tag.
- **Easy to Learn:** If you have a prior knowledge of any programming language, it is easy to learn PouchDB.
- **Open Source:** PouchDB is an Open Source Application and is available on GitHub.

Advantages of PouchDB

Following are the advantages of PouchDB –

- Since PouchDB resides inside the browser, there is no need to perform queries over the network, this results in faster execution of queries.
- You can synchronize the data with any of the supported server and by doing so you can run apps both online and offline.

Browsers that Support PouchDB

Following are the browsers that support PouchDB -

- Firefox 29+ (Including Firefox OS and Firefox for Android)
- Chrome 30+
- Safari 5+
- Internet Explorer 10+
- Opera 21+
- Android 4.0+
- iOS 7.1+
- Windows Phone 8+

2. PouchDB – Environment

This chapter explains how to download and install PouchDB in your system.

Installing PouchDB

In order to work with PouchDB, you need to download the file **.js** file and include it in your script. Following are the steps to install PouchDB.

Step 1

Visit the homepage of PouchDB website, by clicking the following link –

https://PouchDB.com/

Step 2

Click the Download button on the top right hand side of the web page as shown in the above screenshot. This will download **PouchDB-5.3.0.min.js** in your system.

Step 3

Copy and paste the **PouchDB-5.3.0.min.js** to your working directory and include it in your JavaScript as shown below.

```
<script src="PouchDB-5.3.0.min.js"></script>
```

Installing Pouch Using Node.js

You can also install PouchDB as Node.js module. Following are the steps to install PouchDB using Node.js.

Step 1

Install Node.js by following the steps given in the Installing Node.js section of our coffee script tutorial.

Step 2

Open the command prompt and execute the following command. This will install PouchDB node module in your system.

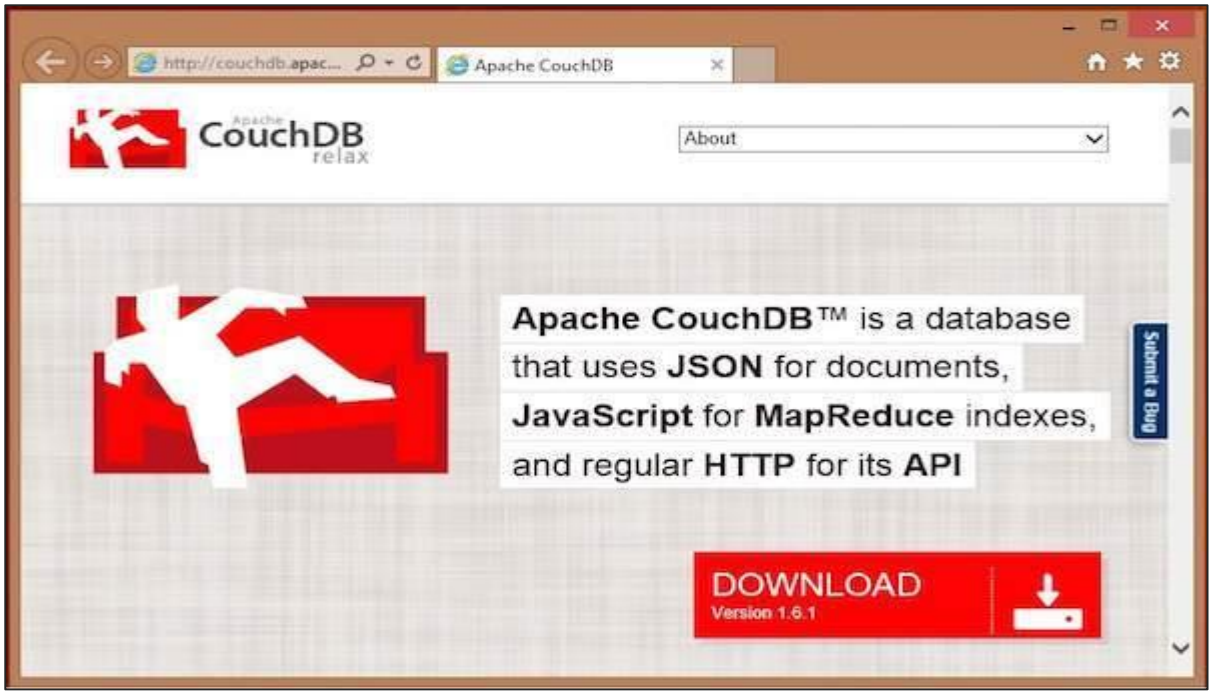
```
npm install --save PouchDB
```

Downloading CouchDB

When offline, PouchDB stores data locally and works like an app. You can access it online by connecting with compatible servers. As we know PouchDB can be connected to CouchDB, so, lets install CouchDB too. Following are the steps to install CouchDB.

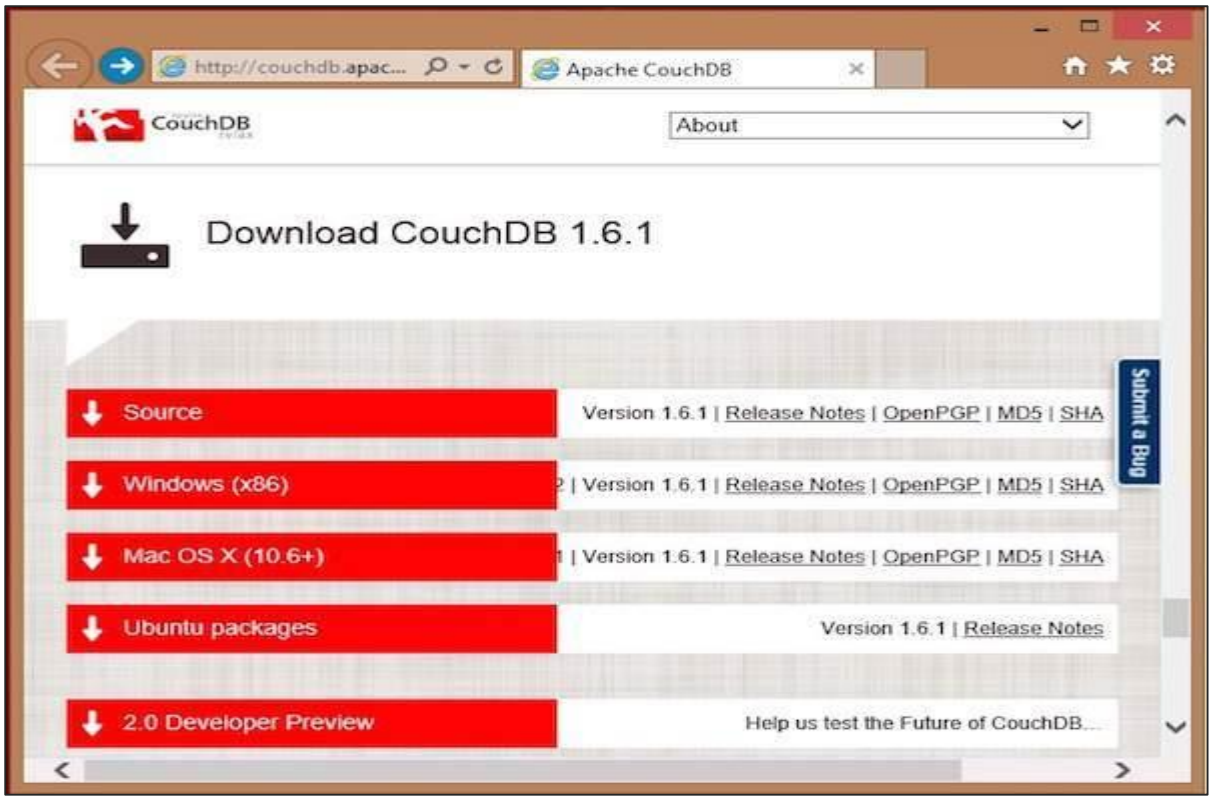
Step 1

The official website for CouchDB is http://couchdb.apache.org. If you click the given link, you can get the home page of CouchDB official website as shown in the following screenshot.



Step 2

If you click on the download button that will lead to a page where the download links of CouchDB in various formats are provided. The following snapshot illustrates the same.



Step 3

Choose the download link for Windows Systems and select one of the provided mirrors to start your download.

Installing CouchDB

A windows executable **setup-couchdb-1.6.1_R16B02.exe** file will be downloaded on your system. Run the setup file and proceed with the installation.

After installing CouchDB in your system successfully, open the folder where CouchDB was installed, go to the bin folder, and start the server by running a script file named **couchdb.bat.**

After installation, open built-in web interface of CouchDB by visiting the following link: http://127.0.0.1:5984/. If everything goes fine, this will give you a web page, which will have the following output.

```
{   "couchdb":"Welcome","uuid":"c8d48ac61bb497f4692b346e0f400d60",
   "version":"1.6.1",
   "vendor":{
      "version":"1.6.1","name":"The Apache Software Foundation"
   } }
```

You can interact with CouchDB web interface by using the following URL –

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

This shows you the index page of Futon, which is the web interface of CouchDB.

3. PouchDB – Create Database

You can create a database in PouchDB using the PouchDB constructor.

Syntax

Following is the syntax of using the PouchDB constructor. To this, you need to pass the name of the database as a parameter.

```
new PouchDB(Database_name)
```

Example

To create a database in PouchDB using **node**, first of all, you need to require the PouchDB package using the **require()** method and then you can create a database as shown in the following example.

```
//Requiring the package
var PouchDB = require('PouchDB');

//Creating the database object
var db = new PouchDB('my_database');
console.log ("Database created Successfully.");
```

Save the above code in a file with the name **Create_Database.js**. Open the command prompt and execute the JavaScript file using **node** as shown below.

```
C:\PouchDB_Examples>node Create_Database.js
```

This will create a database locally (you can see the folder in the current directory) displaying the following message.

```
Database created Successfully.
```

4. PouchDB – Database Info

You can get the basic information about the database using the method named **info()**

Syntax

Following is the syntax of using the **info()** method of PouchDB. This method accepts a callback function.

```
db.info([callback])
```

Example

Following is an example of retrieving database information using the **info()** method. Here, we are displaying the information of the database named **my_database**. In case of error, the error will be displayed on the console.

```
//Requiring the package
var PouchDB = require('PouchDB');

//Creating the database object
var db = new PouchDB('my_database');

//Database information
db.info(function(err, info) {
  if (err) {
    return console.log(err);
    }
  else {
    console.log(info);
  }
});
```

Save the above code in a file with the name **Database_info.js**. Open the command prompt and execute the JavaScript file using **node** as shown below.

```
C:\PouchDB_Examples>node Database_info.js
```

This will display the info of the specified database as follows.

```
{ doc_count: 0,
  update_seq: 0,
  backend_adapter: 'LevelDOWN',
  db_name: 'my_database',
  auto_compaction: false,
  adapter: 'leveldb' }
```

Remote Database Info

In the same way, you get the information of a database that is saved remotely on the server (CouchDB). To do so, instead of database name, you need to pass the path to the required database in CouchDB.

Example

Following is an example of retrieving information of a database that is saved in the CouchDB server. This code gives you information of a database named **my_database**.

```
//Requiring the package
var PouchDB = require('PouchDB');

//Creating the database object
var db = new PouchDB('http://localhost:5984/my_database');

//Database information
db.info(function(err, info) {
  if (err) {
    return console.log(err);
    }
  else {
    console.log(info);
  }
});
```

Save the above code in a file with the name **Database_ Remote_info.js**. Open the command prompt and execute the JavaScript file using **node** as shown below.

```
C:\PouchDB_Examples>node Database_remote_info.js
```

This will display the info of the specified database as follows.

```
{ 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: '1458209191708486',
  disk_format_version: 6,
  committed_update_seq: 0,
  host: 'http://localhost:5984/my_database/',
  auto_compaction: false,
  adapter: 'http' }
```

5. PouchDB – Delete Database

You can delete a database in PouchDB using the **db.destroy()** method.

Syntax

Following is the syntax of using the **db.destroy()** method. This method accepts a callback function as a parameter.

```
db.destroy()
```

Example

Following is an example of deleting a database in PouchDB using the **destroy()** method. Here, we are deleting the database named **my_database,** created in the previous chapters.

```
//Requiring the package
var PouchDB = require('PouchDB');

//Creating the database object
var db = new PouchDB('my_database');

//deleting database
db.destroy(function (err, response) {
  if (err) {
    return console.log(err);
  } else {
    console.log ("Database Deleted”);
  }
});
```

Save the above code in a file with the name **Delete_Database.js**. Open the command prompt and execute the JavaScript file using **node** as shown below.

```
C:\PouchDB_Examples >node Delete_Database.js
```

This will delete the database named **my_database** which is stored locally displaying the following message.

```
Database Deleted
```

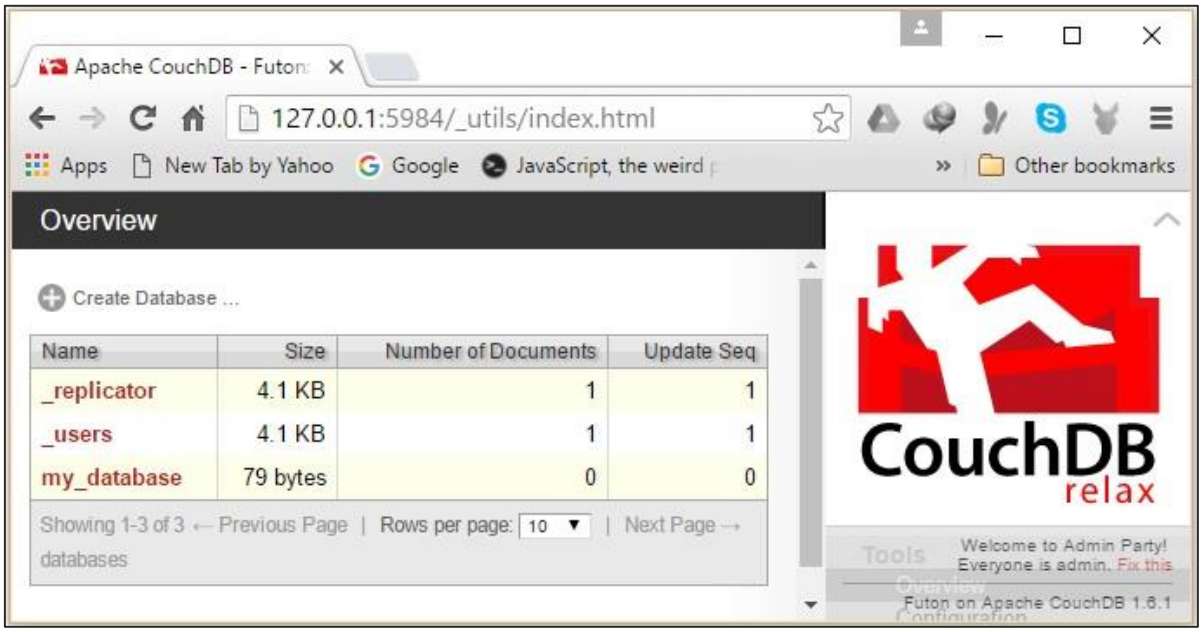
Deleting a Remote Database

In the same way, you can delete a database that is stored remotely on the server (CouchDB).

To do so, instead of database name, you need to pass the path to the database that is required to be deleted, in CouchDB.

Example

Suppose there is a database named **my_database** in the CouchDB server. Then, if you verify the list of databases in CouchDB using the URL http://127.0.0.1:5984/_utils/index.html you will get the following screenshot.



Following is an example of deleting a database named **my_database** that is saved in the CouchDB server.

```
//Requiring the package
var PouchDB = require('PouchDB');

//Creating the database object
var db = new PouchDB('http://localhost:5984/my_database');

//Database information
db.info(function(err, info) {
  if (err) {
    return console.log(err);
    }
  else {
    console.log(info);
```

```
  }
});
```

Save the above code in a file with the name **Database_Remote_info.js**. Open the command prompt and execute the JavaScript file using **node** as shown below.

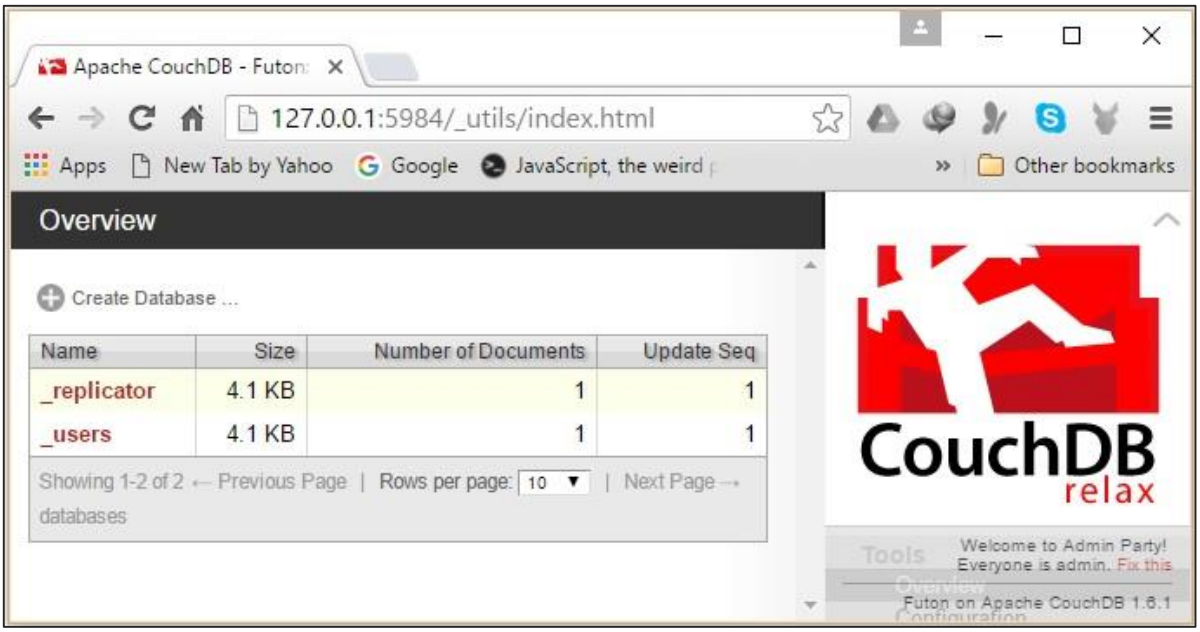
```
C:\PouchDB_Examples >node Remote_Dtabase_Delete.js
```

This deletes the specified database from PouchDB displaying the following message.

```
Database Deleted
```

Verification

After executing the above program, if you visit the URL again, you will get the following screenshot. Here you can observe only two databases since **my_database** was deleted.



End of ebook preview

If you liked what you saw…

Buy it from our store @ https://store.tutorialspoint.com