



# CouchDB

Jerry, Nic, and Omer

# CouchDB – Relax

- CouchDB is an open-source document store DB that uses JSON for document storage.
- Queries are done through JavaScript, API is MapReduce + HTTP

# CouchDB

- Designed to handle changes from multiple sources.
  - Revision control exists for this purpose; each document has a revision number which can be used to compare to another document's revision number.

# Introduction to JSON

- All JSON documents start with { and end with }
- Strings are always enclosed with double-quotes
- Items inside JSON objects are separated by commas
- Example:

```
- {  
  "userName": "Mark",  
  "age": 33,  
  "gender": "M"  
}
```

# JSON Constructs

More Examples:

```
{  
  "fruitName": "apple",  
  "color": "green",  
  "numberOfSeeds": 5,  
  "parentTypes": [  
    "Granny Smith",  
    "Ambrosia"  
  ]  
}
```

String data  
String data  
Integer data  
Array

# GET

- Retrieve database information or documents.

```
curl -X GET host:port/db_name
```

Gets the database information

```
curl -X GET host:port/db_name/doc_id
```

Gets the documents in the database

```
curl -X GET host:port/_all_dbs
```

Get all the databases

```
curl -X GET host:port/db_name/_all_docs
```

Get all the documents in certain database

# HEAD

- Head returns the basic information of the document.

```
curl -X HEAD host:port/db_name/doc_id
```

# PUT and POST

- PUT and POST adds databases or documents.

```
curl -X PUT host:port/db_name
```

This creates a database

```
curl -X PUT host:port/db_name/doc_id -d "{\ "key\":  
\ "value\", "key\": \"value\".....}"
```

To store number: \ "key": 1

To store array: \ "key": [ \ "value1\", \ "value2\"... ]



# DELETE and UPDATE

- When you want to delete or update a document in couchDB you have to provide `_rev` at the end of the query.

```
curl -X DELETE host:port/db_name/doc_id?rev=.....
```

It will only delete the document if you provided with the correct rev

```
curl -X PUT host:port/db_name/doc_id?rev=...../ -d  
“{.....}”
```

This will overwrite the original document with the same id.

[http://wiki.apache.org/couchdb/HTTP\\_Document\\_API](http://wiki.apache.org/couchdb/HTTP_Document_API)

# VIEW

- After you created the database and the documents you can then create views (used like queries).
- We can do views in cURL, but it is difficult to PUT, UPDATE, DELETE views through cURL. It is easier to do it through Futon.
- Two main types: *permanent* (via GET) and *temporary* (via POST)
- *Temporary* views are computationally expensive, so should not be used in deployed systems.
- CouchDB was designed with frequent 'inserts' in mind, along with infrequent changes to 'selects' (views)
  - You can make *Temporary Views* through Futon, then save them to make them *Permanent Views*

- Under database you can create view

The screenshot shows the Apache CouchDB Futon Utility Client interface. The browser address bar displays `http://localhost:5984/_utils/`. The main content area is titled "hello-world" and includes navigation links for "Overview", "Create Document ...", "Compact Database", and "Delete Database". A "Select view:" dropdown menu is set to "All documents", with an arrow pointing to it. Below this is a table with columns "Key" and "Value", showing "Showing 0--1 of 0 rows". The table has navigation controls for "Previous Page", "Rows per page: 10", and "Next Page". On the right sidebar, there is a CouchDB logo with the text "CouchDB relax" and a "Tools" section with links for "Overview", "Replicator", "Configuration", and "Test Suite". Below that is a "Recent Databases" section listing "hello-world". The footer of the sidebar indicates "Futon on Apache CouchDB 0.9.0".

Apache CouchDB: Futon Utility Client

http://localhost:5984/\_utils/

Overview **hello-world**

+ Create Document ... ⓘ Compact Database ✕ Delete Database

Select view: All documents

Key	Value
Showing 0--1 of 0 rows	

← Previous Page | Rows per page: 10 | Next Page →

**CouchDB**  
relax

Tools

- Overview
- Replicator
- Configuration
- Test Suite

Recent Databases

- hello-world

Futon on Apache CouchDB 0.9.0

Done

Apache CouchDB - Futon: Browse Database

http://127.0.0.1:5984/\_utils/database.html?hello-world/\_temp\_view

Overview > hello-world

[+ Create Document ...](#)
[+ Compact Database](#)
[x Delete Database](#)
Select view:

View Code


Map Function: <pre>function(doc) {   emit(null, doc); }</pre>	Reduce Function (optional):
--	-----------------------------

Language: 
=

**Warning:** Please note that temporary views are not suitable for use in production, as they are really slow for any database with more than a few dozen documents. You can use a temporary view to experiment with view functions, but switch to a permanent view before using them in an application.

Key	Value
<b>null</b> ID: 16701272c45cd9c9b8f19b16dbec3598	<pre>{_id: "16701272c45cd9c9b8f19b16dbec3598", rev: "1-1691925970", item: "apple", prices: {Fresh Mart: 1.59, Price Max: 5.99, Apples Express: 0.79}}</pre>
<b>null</b> ID: de1a66fec97dce70ecab177f2596b007	<pre>{_id: "de1a66fec97dce70ecab177f2596b007", rev: "1-567524839", item: "banana", prices: {Fresh Mart: 1.99, Price Max: 0.79, Banana Montana: 4.22}}</pre>
<b>null</b> ID: e7dba71e9885d4eadb1c8fd01ba8d5ad	<pre>{_id: "e7dba71e9885d4eadb1c8fd01ba8d5ad", rev: "1-4255393789", item: "orange", prices: {Fresh Mart: 1.99, Price Max: 3.19, Citrus Circus: 1.09}}</pre>

Showing 1-3 of 3 rows ← Previous Page | Rows per page:  | Next Page →



**CouchDB**  
relax

**Tools**

- [Overview](#)
- [Configuration](#)
- [Replicator](#)
- [Status](#)
- [Test Suite](#)

**Recent Databases**

- asdasd
- couchio\_production
- couchio\_test
- hello-world
- ploklight

Futon on Apache CouchDB  
0.9.0a753629

# Views

- *Views* are essentially the *map* component of MapReduce in CouchDB.
  - Because of this, views are written in JavaScript as functions
- A few necessary components:
  - `function (doc) {...}`
  - *doc* is the current document being processed; each document goes through the map one by one
- Examples:

```
function(doc) {  
    emit(doc.name, doc.age);  
}
```

```
function(doc) {  
    if(doc.scores.name) {  
        for(var i in soc.scores)  
            emit(doc.scores[i].name, doc.scores[i].score)  
        }  
    }  
}
```

You can do sum or count by using the reduce function

MAP:

```
Function(doc){  
  If(doc.scores.name){  
    For(var i in soc.scores)  
      emit(doc.scores[i].name,doc.scores[i].score)  
  }  
}
```

REDUCE:

`_sum` or `_count` or `_stats`