

# 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 \"{.....}\"
```

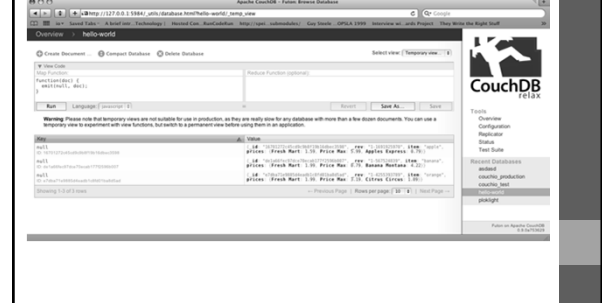
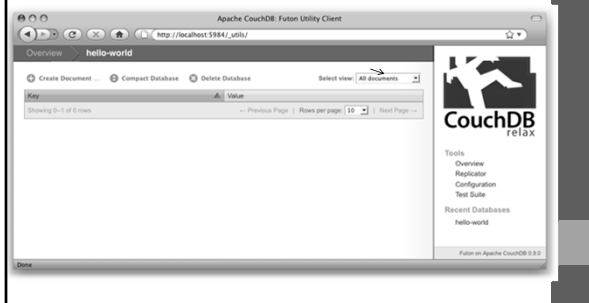
This will overwrite the original document with the same id.

[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



## 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`