



CouchDB  
relax

# Who am I?

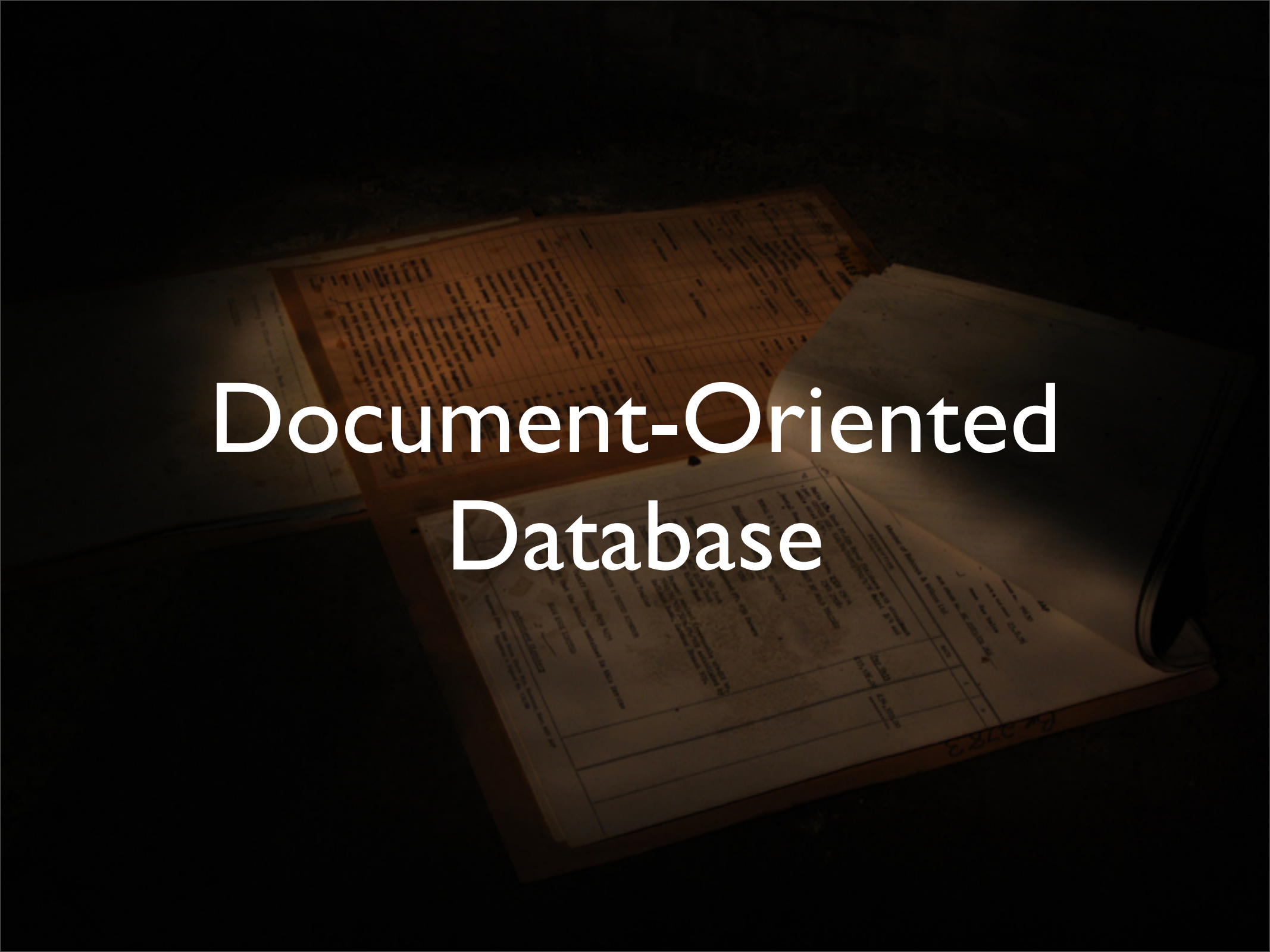
```
{  
  "name": "Myles Braithwaite",  
  "job": "Web Developer",  
  "url": "http://mylesbraithwaite.com",  
  "twitter": "http://twitter.com/mylesb",  
  "xmpp": "me@mylesbraithwaite.com",  
  "email": "me@mylesbraithwaite.com"  
}
```

# Monkey in your Soul

```
$ curl -X GET --head monkeyinyoursoul.com
HTTP/1.1 200 OK
X-Name: Monkey in your Soul
X-URL: monkeyinyoursoul.com
X-Email: info@monkeyinyoursoul.com
X-Twitter: http://twitter.com/miys
X-Owner: Myles Braithwaite
X-Owner-Email: myles@monkeyinyoursoul.com
```

A photograph of a dilapidated room with graffiti on the wall and a couch in the foreground. The room is in a state of decay, with peeling paint, exposed wooden beams, and a window with broken panes. The wall is covered in graffiti, including a crown and various abstract shapes. A grey couch is positioned in the center of the room, and the floor is covered in debris and dirt. The text "What is CouchDB?" is overlaid in white on the couch.

What is CouchDB?



# Document-Oriented Database

No Rows or Columns

But

# Collection of JSON Documents



This is know as  
“Schema-Free”

```
{  
  "fn": "Myles Braithwaite",  
  "emails": [  
    "me@myles.tk",  
    "myles@miys.net"  
  ],  
  "company": "Monkey in your Soul"  
}
```

```
<person>
  <fn>Myles Braithwaite</fn>
  <emails>
    <email>me@myles.tk</email>
    <email>myles@miys.net</email>
  </emails>
  <company>Monkey in your Soul</company>
</person>
```

# Data types

```
"number": 1
"floating_point": 1.11035
"string": "Hello, World!"
"boolean": true
"array": [1, 2, 3]
"object": {
  1: "one",
  2: "two",
  3: "three"
}
>null": null
```



ADAM KOFORD



ADAM KOFORD



ADAM KOFORD



ADAM KOFORD

400 Bad Request

401 Unauthorized

402 Payment Required

403 Forbidden



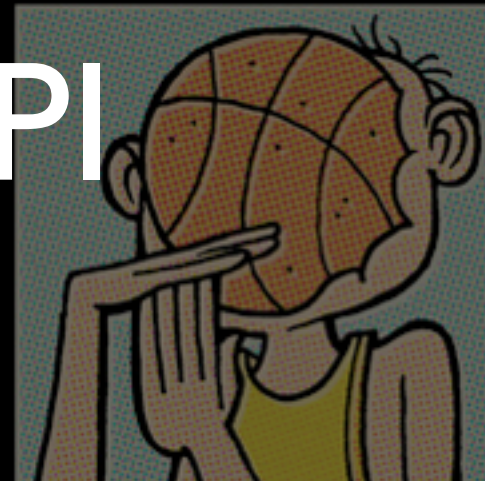
ADAM KOFORD



ADAM KOFORD



ADAM KOFORD



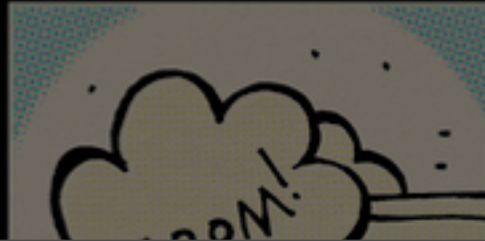
ADAM KOFORD

404 Not Found

405 Method Not Allowed

406 Not Acceptable

408 Request Timeout



# HTTP RESTful API

*“CouchDB is built of the  
Web”*

- Jacob Kaplan-Moss

# Read the RFC 2616

<http://bit.ly/rfc2616>



# REST

- **R**epresentational **S**tate **T**ransfer
- The foundation of Web
  - basic HTTP methods like:
    - POST, GET, PUT, and DELETE

Think low level  
XML-RPC and SOAP

HTTP Methods	CRUD
POST	Create and Update
GET	Read
PUT	Create and Replace
DELETE	Delete
<a href="http://bit.ly/rfc2616-sec9"><u>http://bit.ly/rfc2616-sec9</u></a>	

201 Created

Created a Document

200 OK

Succeeded

404 Object Not  
Found

Document does not  
Exist

<http://bit.ly/rfc2616-sec10>

A wide-angle landscape photograph showing a deep fjord valley. In the foreground, there is a rocky, grassy hillside. The middle ground features a large, calm body of water (a fjord) that reflects the surrounding mountains and the sky. The mountains are rugged and covered in green vegetation. The sky is filled with large, white, fluffy clouds. The word "Views" is written in a large, white, sans-serif font across the center of the image.

# Views

# Aggregation and Reporting on Documents

But how do you  
aggregate  
non-structured data?

Instead of using SQL



JavaScript



Distributed

# Using Replication

You have your  
Database.



You

You want to share.



You



Friend

Full Control



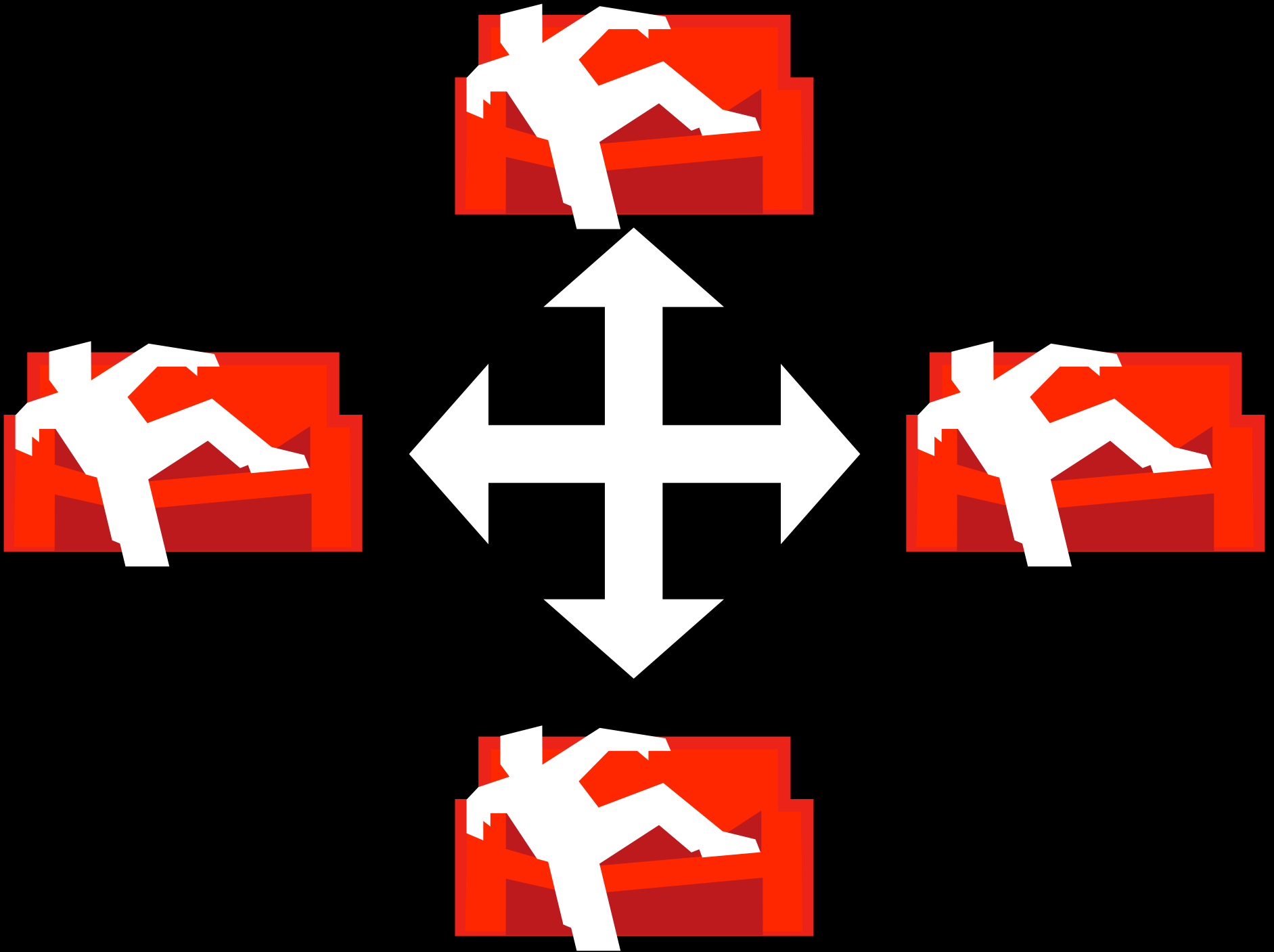
# Bi-Directional Changes



You



Friend



# Built in Conflict Detection



# Using the API

Database

# Create a Database

```
$ curl -v -X PUT $COUCHDB/address_book/
```

```
PUT /address_book/
```

```
Host: 127.0.0.1:5984
```

HTTP/1.1 201 Created

Server: CouchDB/0.9.1 (Erlang OTP/R12B)

```
{"ok":true}
```



HTTP/1.1 412 Precondition Failed  
Server: CouchDB/0.9.1 (Erlang OTP/R12B)

```
{"error": "file_exists", "reason": "The  
database could not be created, the file  
already exists."}
```

```
CREATE TABLE "people" (  
    "id" SERIAL PRIMARY KEY,  
    "fn" VARCHAR(50),  
    "company" VARCHAR(50)  
);
```

```
CREATE TABLE "person_emails" (  
    "id" SERIAL PRIMARY KEY,  
    "email" VARCHAR(50),  
    "person_id" INTEGER REFERENCES people(id)  
);
```

# Delete a Database

```
$ curl -v -X DELETE $COUCHDB/address_book/
```

```
DELETE /address_book/ HTTP/1.1
```

```
Host: 127.0.0.1:5984
```

HTTP/1.1 200 OK

Server: CouchDB/0.9.1 (Erlang OTP/R12B)

Date: Thu, 08 Oct 2009 15:11:31 GMT

Content-Type: text/plain; charset=utf-8

Content-Length: 12

Cache-Control: must-revalidate

{"ok":true}

# Fetching all Rows

```
$ curl -v -X GET $COUCHDB/address_book/  
_all_docs
```

```
GET /address_book/_all_docs HTTP/1.1  
Host: 127.0.0.1:5984
```

HTTP/1.1 200 OK

Server: CouchDB/0.9.1 (Erlang OTP/R12B)

Date: Thu, 22 Oct 2009 13:12:05 GMT

Content-Type: text/plain; charset=utf-8

Content-Length: 12

Cache-Control: must-revalidate

```
{"total_rows":1,"offset":0,"rows":[  
{"id":"9c118d4acfcdd1bc0687bcaa53c53674", "  
key":"9c118d4acfcdd1bc0687bcaa53c53674", "v  
alue":{"rev":"1-180621101"}}  
]}
```

```
{
  "total_rows": 1,
  "offset": 0,
  "rows": [
    {
      "id": "9c118d4acfcdd1...",
      "key": "9c118d4acfcdd1...",
      "value": {
        "rev": "1-180621101"
      }
    }
  ]
}
```

Document



# Creating a Document

```
curl -v -X POST $COUCHDB/address_book -d  
'{"fn": "Myles Braithwaite", "emails":  
["me@myles.tk", "myles@miys.net"], "company":  
"Monkey in your Soul"}'
```

```
POST /address_book HTTP/1.1  
Host: 127.0.0.1:5984  
Content-Length: 103
```

HTTP/1.1 201 Created

Location: [http://127.0.0.1:5984/  
address\\_book/](http://127.0.0.1:5984/address_book/)

9c118d4acfcdd1bc0687bcaa53c53674

```
{"ok":true,"id":"9c118d4acfcdd1bc0687bcaa5  
3c53674","rev":"1-180621101"}
```

```
INSERT INTO people ("fn", "company") VALUES  
( 'Myles Braithwaite', 'Monkey in your Soul' );  
INSERT INTO person_emails ("email", "person_id")  
VALUES ( 'me@myles.tk', 1 );  
INSERT INTO person_emails ("email", "person_id")  
VALUES ( 'myles@miys.net', 1 );
```

# Get a Document

```
curl -v -X GET $COUCHDB/address_book/  
9c118d4acfcdd1bc0687bcaa53c53674
```

```
GET /address_book/  
9c118d4acfcdd1bc0687bcaa53c53674 HTTP/1.1  
Host: 127.0.0.1:5984
```

HTTP/1.1 200 OK

Server: CouchDB/0.9.1 (Erlang OTP/R13B)

Etag: "1-180621101"

```
{"_id": "9c118d4acfcdd1bc0687bcaa53c53674",  
  "_rev": "1-180621101", "fn": "Myles  
Braithwaite", "emails":  
  ["me@myles.tk", "myles@miys.net"], "company":  
  "Monkey in your Soul"}
```

```
{
  "_id":
  "9c118d4acfcdd1bc0687bcaa53c53674",
  "_rev": "1-180621101",
  "fn": "Myles Braithwaite",
  "emails": [
    "me@myles.tk",
    "myles@miys.net"
  ],
  "company": "Monkey in your Soul"
}
```

# Update a Document

```
curl -v -X PUT $COUCHDB/address_book/  
9c118d4acfcdd1bc0687bcaa53c53674 -d  
'{"_id":"9c118d4acfcdd1bc0687bcaa53c53674"  
, "_rev":"1-180621101", "fn":"Myles  
Braithwaite", "emails":  
["me@myles.tk", "myles@miys.net"], "company"  
:"Monkey in your Soul", "urls":["http://  
twitter.com/mylesb", "http://  
mylesbraithwaite.com/"]}'
```

# Update a Document

```
curl -v -X PUT $COUCHDB/address_book/  
9c118d4acfcdd1bc0687bcaa53c53674 -d  
'{"_id": "9c118d4acfcdd1bc0687bcaa53c53674"  
, "_rev": "1-180621101", "fn": "Myles  
Braithwaite", "emails":  
["me@myles.tk", "myles@miys.net"], "company"  
: "Monkey in your Soul", "urls": ["http://  
twitter.com/mylesb", "http://  
mylesbraithwaite.com/"]}'
```



PUT /address\_book/

9c118d4acfcdd1bc0687bcaa53c53674 HTTP/1.1

Host: 127.0.0.1:5984

HTTP/1.1 201 Created

Server: CouchDB/0.9.1 (Erlang OTP/R13B)

Etag: "2-2080422110"

```
{"ok":true,"id":"9c118d4acfcdd1bc0687bcaa53c53674", "rev":"2-2080422110"}
```

# Attaching a File to a Document

```
curl -v -X PUT $COUCHDB/address_book/  
215422c46959ffc1bd875b025e8d1360/  
photo.jpg?rev=5-2484899332 --data-binary  
@photo.jpg -H 'Content-Type: image/jpeg'
```

```
PUT /address_book/  
215422c46959ffc1bd875b025e8d1360/  
photo.jpg?rev=5-2484899332 HTTP/1.1  
Host: 127.0.0.1:5984  
Content-Type: image/jpeg  
Content-Length: 25565  
Expect: 100-continue
```

HTTP/1.1 201 Created

Server: CouchDB/0.9.1 (Erlang OTP/R13B)

Content-Type: text/plain;charset=utf-8

```
{"ok":true,"id":"215422c46959ffc1bd875b025  
e8d1360","rev":"6-3885229966"}
```

```
curl -v -X GET $COUCHDB/address_book/  
215422c46959ffc1bd875b025e8d1360/photo.jpg
```

```
GET /address_book/  
215422c46959ffc1bd875b025e8d1360/photo.jpg  
HTTP/1.1  
Host: 127.0.0.1:5984
```



# Copy a Document

```
curl -v -X COPY $COUCHDB/address_book/  
215422c46959ffc1bd875b025e8d1360 -H  
'Destination: myles-braithwaite'
```

```
COPY /address_book/  
215422c46959ffc1bd875b025e8d1360 HTTP/1.1  
Host: 127.0.0.1:5984  
Destination: myles-braithwaite
```

HTTP/1.1 201 Created

Server: CouchDB/0.9.1 (Erlang OTP/R13B)

Etag: "1-4208662590"

{"rev": "1-4208662590"}



# Delete a Document

```
curl -v -X DELETE $COUCHDB/address_book/  
9c118d4acfcdd1bc0687bcaa53c53674?  
rev=2-2080422110
```

```
DELETE /address_book/  
9c118d4acfcdd1bc0687bcaa53c53674?  
rev=2-2080422110 HTTP/1.1  
Host: 127.0.0.1:5984
```

HTTP/1.1 200 OK

Server: CouchDB/0.9.1 (Erlang OTP/R13B)

Etag: "3-3181309232"

```
{"ok":true,"id":"9c118d4acfcdd1bc0687bcaa53c53674","rev":"3-3181309232"}
```

Demo Futon



# Libraries

# Python

- <http://code.google.com/p/couchdb-python/>

# Ruby

- <http://github.com/jchris/couchrest>

# Erlang

- <http://bitbucket.org/benoitc/couchbeam/>

# Java

- <http://code.google.com/p/couchdb4j/>



I have created a Library  
in an afternoon.

```
"""
```

```
Copyright 2009 Myles Braithwaite <me@mylesbraithwaite.com
```

```
Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at
```

```
http://www.apache.org/licenses/LICENSE-2.0
```

```
Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the license for the specific language governing permissions and  
limitations under the License.
```

```
"""
```

```
import urllib, base64, string
```

```
import simplejson  
import httplib2
```

```
class Couch(object):
```

```
    """
```

```
    :param host: The full URI to the CouchDB host.  
    :params username password: The username and password to login.
```

```
    """
```

```
    def __init__(self, uri='http://127.0.0.1:5984', username=None, password=None):  
        self.uri = uri  
        self.username = username  
        self.password = password
```

```
    def _connect(self):
```

```
        """Connect to the CouchDB server.  
        """
```

```
        h = httplib2.Http()  
        if self.username and self.password:  
            h.add_credentials(self.username, self.password)
```

```
        return h
```

```
    def _http(self, path, method, headers={}, body=None):
```

```
        c = self._connect()  
        return c.request('%s%s' % (self.uri, path),  
                        method,  
                        headers=headers,  
                        body=body)
```

```
    # Database operations
```

```
    def list_databases(self):
```

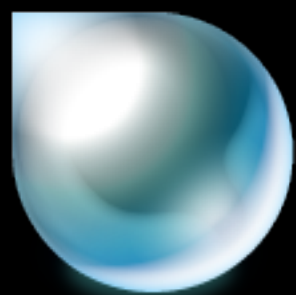
```
        """To get a list of databases on a CouchDB server.
```

```
        :return: A list of database names.  
        :rtype: list  
        """
```

```
        headers = { "Accept": "application/json" }  
        response, content = self._http('/_all_dbs', "GET", headers=headers)
```

Just shy of 300 lines

- Python: `httpLib`
- Perl: `Net::HTTP`
- Java: `java.net.HttpURLConnection`
- Ruby: `net/http`



mozilla LABS  
**Raindrop**

# Store, sync and share

Back up your life with Ubuntu One

Sign in

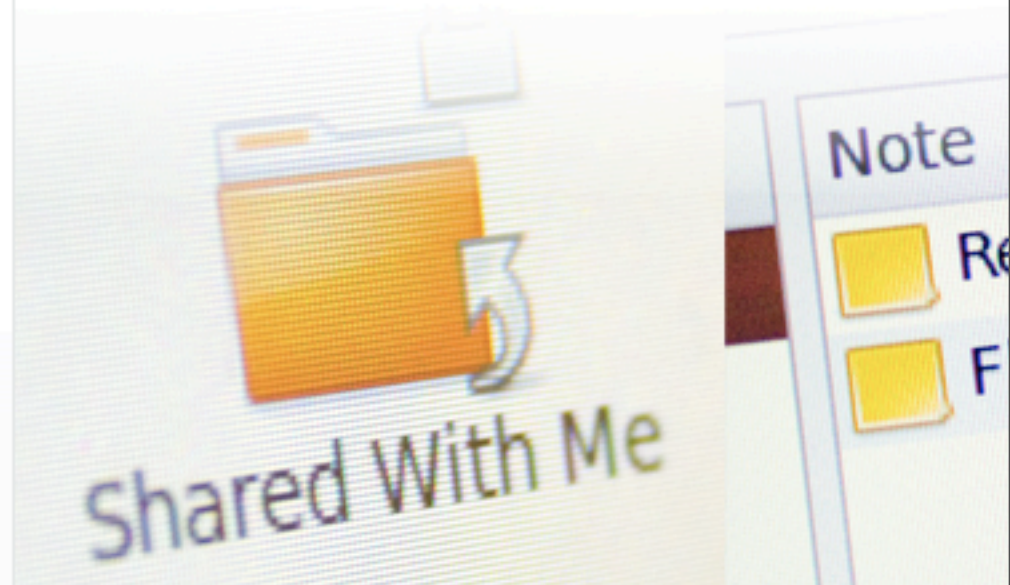
Subscribe »

## What is Ubuntu One?

Ubuntu One is your personal cloud. You can use it to back up, store, sync and share your data with other Ubuntu One users.

Ubuntu One gives all features and 2 GB of essential storage to everyone. If you need more space, choose 50 GB for just \$10 a month.

[Subscribe to Ubuntu One »](#)



## Sync your files and share them with others

With Ubuntu One, your projects are as public (and your secrets as safe) as you want them to be.

[More about syncing and sharing files »](#)

## Edit Tomboy notes online and in sync

Tomboy notes always available wherever you are.

[More about syncing »](#)



I will be at the GTALUG  
Booth at:  
4:00 p.m.

Questions



Other  
Document-Oriented  
Database Systems

# MongoDB

<http://mongodb.org>

# Amazon SimpleDB

<http://aws.amazon.com/simpledb>

# Photo Credits

- <http://www.flickr.com/photos/83737641@N00/2377691249/>
- <http://www.flickr.com/photos/dragon2309/1490657223/>
- <http://www.flickr.com/photos/vincealongi/354450827/>
- <http://www.flickr.com/photos/stewart/99129170/>
- <http://www.flickr.com/photos/matthigh/1817222239/>
- <http://www.flickr.com/photos/garibaldi/300430036/>