

# CouchDB is sacrilege... mmm, delicious sacrilege

Dan Scott, Laurentian University

code4lib 2008

February 28, 2008

*(Thanks Aaron!)*

# A wonderful, awful idea

- Damien Katz wanted to recreate Lotus Notes
  - with a good API
  - and a good data format
  - under an open source license (Apache)
  - without wasting effort on making it a groupware email / calendaring application
- See <http://lotusnotessucks.4t.com/>

# Sacrilege break

**Mmm, delicious sacrilege**

Mangoat | 05/22/2006 - 6:36am

We have Jesus on a piece of toast.. they have holy  
Arabic phrases on a tuna

Put those 2 together and you've got the holiest sandwich  
ever. You'd just need some Buddhist mayo.

*<http://mangoat.net>*

# What is CouchDB?

- A document database
  - with a RESTful API
  - and versioning
  - and replication
- Sounds a lot like ThingDB or Amazon S3, eh?
- Written in Erlang for concurrency

# HACK IS NOT A CRIME

- Just for fun – assuming the router is working, that we're both on the code4lib access point, and that firewalls won't get in the way – you can play with CouchDB on my laptop during the presentation
  - Base URI: <http://192.168.3.??? :5984/>
  - Admin interface: [http://192.168.3.??? :5984/\\_utils/](http://192.168.3.??? :5984/_utils/)

# It has an API...

- And that API is RESTful, and talks JSON. Yay.
  - GET – select
  - POST - update
  - PUT - insert
  - DELETE - drop
  - HEAD and OPTIONS? Not so much.

# Creating, listing, and deleting databases

- List databases:

```
GET /_all_dbs HTTP/1.0
```

- Create database:

```
PUT /newdatabase/ HTTP/1.0
```

- DELETE database:

```
DELETE /newdatabase/ HTTP/1.0
```

- Database metadata:

```
GET /newdatabase/ HTTP/1.0
```

# Document properties

- Every document has:
  - 1 `_id` attribute (unique, can be supplied or generated automaticall)
  - 1 `_rev` attribute
  - $n$  arbitrary key-value pairs (where each value is a JSON object)
    - keys beginning with `_` are reserved for CouchDB



# Sacrilege break

## Shrimp Pho with Lemongrass, Chili, and Ginger

...

I didn't have lemongrass, so I used the zest of a few lemons. I also didn't want to buy a serrano chili, so used chili powder. I love garlic but I pretty much left it out.

Sacrilege, I know. Delicious, delicious sacrilege.

*(<http://sugaredharpy.com>, filed under "Flaming ovaries")*

# Creating documents

- **PUT** to a named location to give it a specific ID

```
PUT /database/documentID HTTP/1.0
```

```
HEADER junk
```

```
{...}
```

- **POST** to the database to generate an ID

```
POST /database/ HTTP/1.0
```

```
HEADER junk
```

```
{...}
```

# Bulk document creation

- POST to the database an array of JSON records:

```
POST /database/ HTTP/1.0
```

```
HEADER junk
```

```
[  
  {"_id": "1", "name": "OSS Endeca"},  
  {"_id": "11", "name": "Fac-Back-OPAC"},  
  {"_id": "33", "name": "kobold chieftain"},  
]
```

# Retrieving documents

- GET a named location returns JSON

```
GET /database/doc_name HTTP/1.0
```

- optional *rev* param for a specific revision

```
GET /database/doc_name?rev=9467819C HTTP/1.0
```

- To get a list of all revisions stuffed in a *revs* field, ask for  
*?revs=true*

```
GET /database/doc_name?revs=true HTTP/1.0
```

# Documents can have attachments

- (Currently): Insert them into an `_attachments` field, with filename mapped to type and data subfields

```
_attachments: {  
  "citation.txt": {  
    "type": "text/html",  
    "data": "<html>\r\n<head>\r\n<title>How I learned to  
relax and love CouchDB</title>"  
  }  
}
```

# Retrieving document attachments

- Not working in trunk, friends!
- (Currently): The document itself just lists the attached file names, with *stub*, *type*, and *length* attributes
- (Currently): Ask for the attachment as a GET param:

```
GET /database/doc_name?attachment=citation.txt HTTP/1.0
```

# Updating and deleting documents

- PUT with the revision attribute specified in the document body

```
PUT /database/doc_name HTTP/1.0
```

```
HEADER junk
```

```
{ ...
```

```
  "_rev": "9467819C"
```

```
}
```

- DELETE with the revision specified

```
DELETE /database/doc_name?rev=9467819C HTTP/1.0
```

# Sacrilege break

Listening to Jenny Lewis on Rabbit Fur Coat is what I imagine tasting the water turned to wine would be like. That statement alone may give you a taste of some of the delicious sacrilege that is also on the album.

*<http://kiteflyersociety.blogspot.com/2007/01/year-in-review.html>*



# It ain't SQL, but it has Views

- CouchDB supports both on-the-fly views and persistent views:

```
{
  "_id": "design/work",
  "language": "text/javascript",
  "views": {
    "accepted": "function(doc) { if (doc.accepted == true) { map(doc.name, {desc: doc.description}) } }",
    "all": "function(doc) { map(null, doc) }"
  }
}
```

# ACID?

- Apparently so:
  - Data and structures are never overwritten
  - MVCC is used to ensure a consistent view of the data during reads, while avoiding locking
  - Consistency checks are never required after a crash

# Replication

- This was one of the great features of Notes
  - Bi- or multi-directional incremental replication with automatic conflict resolution and ability to resolve conflicts after replication
  - Replication includes documents and “design documents” (views)
  - Partial replication is also supported

# Authentication

- Current documentation (“technical overview”) suggests “full administrator” vs. “reader” is already implemented, but it lies:
  - There is currently no implemented security model
  - Firewall your CouchDB server and use HTTP auth
- LDAP is “a priority feature” according to the roadmap

# Ready for prime time?

- Attachment API is changing
- Search is not really there yet
  - Experiments with integrating Lucene as a full-text search engine are ongoing but not yet part of the build or granular (see `src/fulltext/lucene`)
  - So, just suck all the data into Solr if you need full-text search
- Once `couchdbwiki.com` is built on CouchDB, then we can talk

# Gratuitous demo

- Don't get too excited
- We'll import some basic data and show off the CouchDB admin interface

# Be sure to wear some flowers in your hair

- Wouldn't it be nice if OpenLibrary did something like replication for bibliographic data?
  - Easy holdings updates
  - LOCKSS
  - Local caching of reviews, images, table of contents, abstracts, tags, comments
  - Hey - we could build our own OpenSocialBiblioCat!

# See...

- CouchDB home - <http://couchdb.org>
- Damien Katz - <http://damienkatz.net>
- Christopher Lenz - <http://www.cmlenz.net>
- Documentation - <http://www.couchdbwiki.com>