

# Upholstering Apache CouchDB



# Who am I?

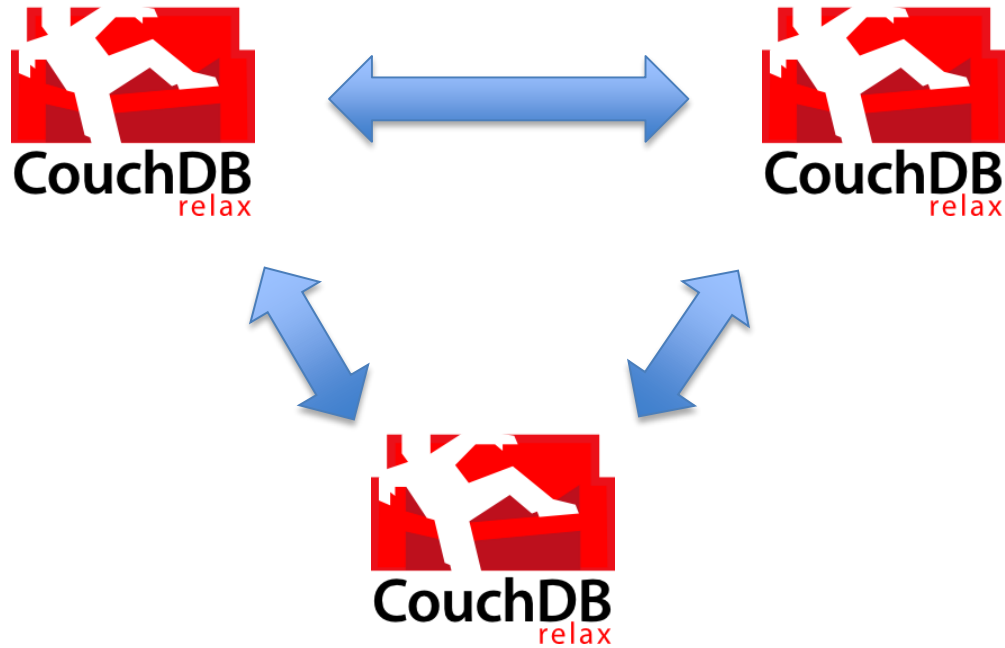


**We're  
hiring!!**

# Apache CouchDB

- Document Storage Database
- Best at Replication
- Also
  - Schemaless
  - Uses Map/Reduce for indexing
  - HTTP for querying
  - JSON for everything else
  - Can store binaries as “attachments”

# Eventually Things Match



because you can...and they replicate!

# **WEB APPS ON COUCHDB**

# Apache CouchDB HTTP API

- GET `{db}/`
  - Database info
- GET `{db}/_all_docs`
  - List of all document's info in the database
  - Primary Index
- GET `{db}/{doc_id}`
  - JSON doc stored here
- GET `{db}/{doc_id}/index.html`
  - Get the index.html file “attached” to the doc

# Apache CouchDB HTTP API

- PUT `/{db}/`
  - Create a database
- PUT `/{db}/{doc_id}`
  - Send in JSON to create a Document
- PUT `/{db}/{doc_id}/index.html`
  - Send in “anything” to create an attachment on that Document
  - ``Content-Type: text/html``

# Web Apps on CouchDB

- CouchDB is a web server
- Good at Etags (304 Not Modified FTW!)
- Virtual Host
- (optionally) CORS enabled
- Basic User Management
- Simple Security Model
- Flexible URL Space



# CouchApps / Design Docs

- Extra Awesome
- MapReduce indexing
- Transform indexes
- Transform documents
- Handle special inbound formats
  - XML, form encoding, CSV
- Rewrite URLs

# GET /{db}/\_design/{app}/

- **\_view**/{index}
  - MapReduce index access
- **\_list**/{transform}/{index}
  - Transform the index output
- **\_show**/{transform}/{doc\_id}
  - Transform a single document
- **\_update**/{parser}[/{doc\_id}]
  - Handle non-JSON (XML, form encoding, etc) updates
- **\_rewrite**/{path}
  - Rewrite URLs

# In JSON

```
{ "_id": "_design/app",
  "views": {
    "by_name": { "map": "function(doc) { emit(doc.name); }", "reduce": "_count" }
  },
  "lists": {
    "toRSS": "function(head, req) { provides('application/rss+xml', ...)"
  },
  "shows": {
    "toHTML": "function(doc, req) { return '<pre>' + doc.toString() + '</pre>'; }"
  },
  "updates": {
    "fromForm": "function(doc, req) { return [req.form, 'Saved the form!']; }"
  },
  "rewrites": [
    { "from": "/",
      "to": "index.html" },
    { "from": "/:doc",
      "to": "_show/toHTML/:doc" },
    { "from": "/feed/:index",
      "to": "_list/feed/:index" }
  ]
}
```

# On your file system

```
/_design/app/  
  views/  
    by_name/  
      map.js  
      reduce.js  
  lists/  
    toRSS.js  
  shows/  
    toHTML.js  
  updates/  
    fromForm.js  
  rewrites.json
```

# `_view/by_name?reduce=false`

```
function (doc) {  
  if ('name' in doc) {  
    emit(doc.name, 1);  
  }  
}
```

ID	Key	Value
kwefsda	"Benjamin Young"	1
23jkhgt	"Glynn Bird"	1
j4hp6f8	"Glynn Bird"	1
p4369zh	"Michelle Phung"	1

# \_view/by\_name

\_count

ID	Key	Value
null	“Benjamin Young”	1
null	“Glynn Bird”	2
null	“Michelle Phung”	1

# \_list/toHTML/by\_name?reduce=false

```
function (head, reg) {  
  provides('html', function () {  
    send('<ul>');  
    while(row = getRow()) {  
      send('<li>' + doc.name + '</li>');  
    }  
    send('</ul>');  
  });  
}
```

```
<ul>  
<li>Benjamin Young</li>  
<li>Glynn Bird</li>  
<li>Michelle Phung</li>  
</ul>
```

# \_show/toHTML/kwefsd

```
function (doc, reg) {  
  if (doc) {  
    return '<pre>' + doc.toString() + '</pre>';  
  } else {  
    // output a form to create the doc...maybe  
  }  
}
```

```
<pre>  
{“doc”:{“name”: “Benjamin Young”}}  
</pre>
```



# POST \_update/fromForm

```
function (doc, req) {  
  if (!doc && req.form) {  
    return [req.form, 'Saved the form!'];  
  } else if (doc) {  
    // update the doc with the contents  
    // of req.form  
  }  
}
```

Saved the form!

- New doc created matching form contents

`_rewrite/*`

- **GET /**  
< index.html contents
- **GET /kwefsd**  
< <pre>  
< {“doc”:{“name”: “Benjamin  
Young”}}  
< </pre>  
– The result of `_show/toHTML/kwefsd`

# Virtual Hosts

- Configuration settings to map domains to `_rewrite/` URLs

```
[vhosts]
example.com = /example-com/_design/app/_rewrite
```

- Magic

```
[vhosts]
*.couchdb.local = /*
:ddocname.:dbname.example.com =
/:dbname/_design/:ddocname/_rewrite
```

Get on CouchDB

# **TOOLS OF THE TRADE**

# Command Line for CouchDB

- Couchapp.py (the granddaddy)
- Erica
- Node.couchapp.js
- Gulp.js + couchdb-push
- Kan.so
- Casters
- Reupholster

# couchapp.py

- `pip install couchapp`
- `couchapp push . \`  
<http://USER:PASS@localhost:5984/db>
- Or
  - `autopush`
  - `pushdocs`
  - `pushapps`

# couchdb-push

- JS-based & useable from node.js build scripts (gulp, etc)
- `couchdb-push \`  
<http://localhost:5984/my-app> \  
`project/couchdb --watch`

# couchdb-push in gulp.js

```
var push = require('couchdb-push');

gulp.task('push', function() {
  push(couch_url, doc,
    function(err, resp) {
      if (err) {
        throw err;
      }
    }
  );
});
```



# UI for CouchDB

- Futon (in CouchDB =< 1.6.1)
- Fauxton (in CouchDB >= 2.0)
  
- <http://ddoc.me>
  
- Make one!


Single Document Web App on CouchDB

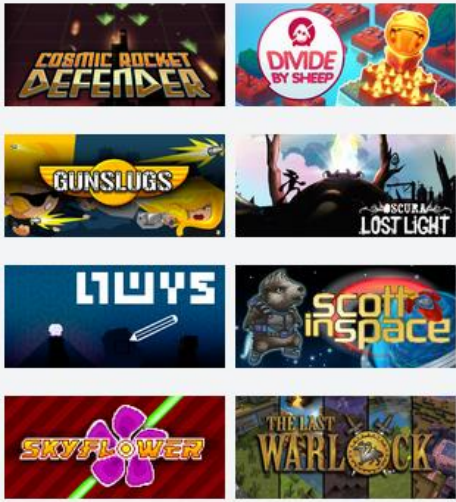
**BUNDLED.IN**

# bundled.in on Phone & Desktop

bundled in?

Currently 31 Bundles [AFAWK](#)


 Bundle Stars - Cosmic Bundle

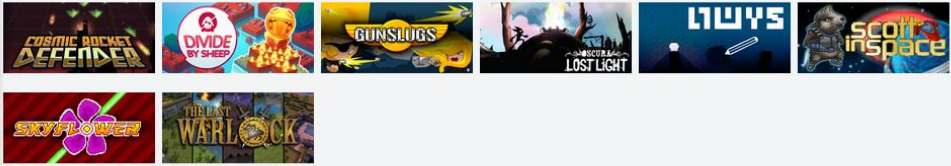



built by [BigBlueHat](#) | powered by Steam

bundled in?


Currently 31 Bundles [AFAWK](#)

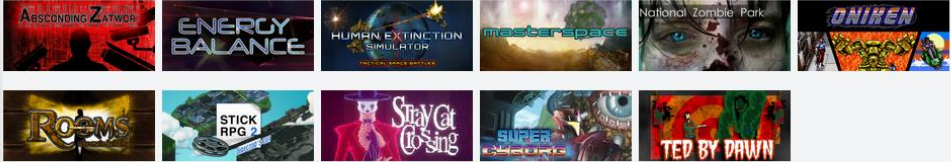
 Bundle Stars - Cosmic Bundle



**QUICKBOOKS INVENTORY TOOL** 

Inventory Management Software Free 14 Day Trial.

 Indie Gala - Friday Special Bundle 21



built by [BigBlueHat](#) | powered by Steam

# [bundled.in](http://bundled.in)

- Video Game Bundle Buying Bonanza!
- Single document drives UI
  - Holds latest bundles + games list
- Static HTML UI hosted from Cloudant
  - Vue.js, superagent, Semantic-UI
- No CouchApp stuff for this one (yet!)
- Wishlist checker uses “external” Python host
  - Filters main list by a screen scraped wishlist

Markdown Editor on CouchDB


# **PILLOW NOTES**

# Pillow Notes

- Yet Another Markdown Editor Thing
- JSON looks like:
  - “\_id”: “...title of the note...”
  - “markdown”: “...the note...”
  - “created”: “...iso8601...”
  - “updated”: “...iso8601...”

# Pillow Notes

new doc name +

2015 ApacheCon: Big Data 

# 2015 ApacheCon: Big Data 0000001

Speaking on:

- \* Decentralized Document Delivery
- \* Upholstering Apache CouchDB
- \* Finding Your Way Around Apache

## 2015 ApacheCon: Big Data 0000001

Speaking on:

- Decentralized Document Delivery
- Upholstering Apache CouchDB
- Finding Your Way Around Apache

x

username password <http://example.com/database> Sync

# Pillow Notes Implementation

- HTML5, CSS, JS
- PouchDB
  - Persistence in browser
  - Replication out to CouchDB, Cloudant, etc
    - For backup, sharing, publication?
- Vue.js
  - Interaction



# Static Hosting Pillow Notes

- On GitHub Pages
  - `git push origin gh-pages`
  - <http://bigbluehat.github.io/pillow-notes/>
- On Cloudant
  - `couchapp push . \`  
`http://{KEY}:{SECRET}@bigbluehat.cloudant.com/pillow-notes/`
  - <http://bigbluehat.cloudant.com/pillow-notes/design/pillow-notes/rewrite/>
- On CouchDB locally
  - `couchapp push . \`  
`http://{user}:{pass}@localhost:5984/pillow-notes/`

# Pillow Notes & Replication

new doc name +

# 2015 ApacheCon: Big Data 0000001

2015 ApacheCon: Big Data 0000001

Username, Password, URL of Database  
Click "Sync"

```
PouchDB.sync(local, remote)  
  .then(function(resp) {...});
```

Bi-directional Replication MAY create conflicts

username

password

http://example.com/database

Sync

# CORS & Single Origin Pain

- Cross Origin Resource Sharing
  - Disables a core feature of the Web
  - Makes moving JSON with Browsers painful
- (re?)Enable CORS
  - `npm install -g add-cors-to-couchdb`
  - Cloudant has some UI, but only works over HTTPS
- Can't share without CORS being **enabled**
- OK...it's actually the Single Origin Policy...

Building a CouchApp

# COUCHMAIL

# CouchMail

- Email archival on CouchDB
- IMAP or Context.io
- Cloudant Full-Text Search
  - If hosted there
- Binary attachment storage
- Original message preservation
  - Lawyers like this one.

# CouchMail Import

- IMAP => JSON
- Starts with newest
- Can skip archived mail (or re-archive)
- Stores “body parts” as attachments
  - text/plain
  - text/html
  - ...actual attachments based on media type

Thinking on CouchDB

# **DEVELOPING WITH DOCUMENTS**

# Development Process

- Craft your JSON
- Craft your Map/Reduce
- ...repeat...
- Design some UI
  - if you didn't before you started
- Implement that in HTML5, JS, and CSS
- Use command line tools to publish into CouchDB



# CouchMail the CouchApp

- `_design/couchapp`
  - `_attachments/`
    - `index.html`
    - `css/*.css`
    - `Js/*.js`
  - `views/`
    - `by_attachment_name`
    - `by_date`
      - `Map.js`
      - `Reduce.js`
    - ...
- `shows`
  - `atom.js`
- `lists`
  - `rss.js`
  - `atom.js`
- `updates`
  - `notes.js`
- `validate_doc_update.js`
- `rewrites.json`

PLANNED

# The Filesystem Mapping

- ddoc/
  - \_attachments/
    - index.html
    - bundle.js
    - styles.css
  - \_docs
    - fixture.json
    - settings.json
  - Views
    - by\_date
      - map.js
      - reduce.js
- shows
  - xml.js
- lists
  - rss.js
  - atom.js
- updates
  - message.js
- validate\_doc\_update.js
- rewrites.json

# Thanks!



- [bigbluehat.com](http://bigbluehat.com)
- [@bigbluehat](https://twitter.com/bigbluehat)
- [github.com/BigBlueHat](https://github.com/BigBlueHat)
- bigbluehat on irc.freenode.net
  - #couchdb #pouchdb #hypothes.is
- [bigbluehat@apache.org](mailto:bigbluehat@apache.org)
- [byoung@bigbluehat.com](mailto:byoung@bigbluehat.com)
- [bigbluehat@hypothes.is](mailto:bigbluehat@hypothes.is)