

# PHPillow & CouchDB

Kore Nordmann <kore@php.net>

May 31, 2009

- ▶ Kore Nordmann, <kore@php.net>, <kn@ez.no>
- ▶ Long time PHP developer
- ▶ Regular speaker, author, etc.
- ▶ Studies computer science in Dortmund
- ▶ Active open source developer:
  - ▶ eZ Components (Graph, WebDav, Document), Arbit, PHPUnit, Torii, *PHPillow*, KaForkL, Image 3D, WCV, ...

Introduction

CouchDB

PHPillow

- ▶ Who uses an RDBMS (relational database management system)?

- ▶ Who uses an RDBMS (relational database management system)?
- ▶ Who uses a hash based “database”? (MemcacheDB, ...)

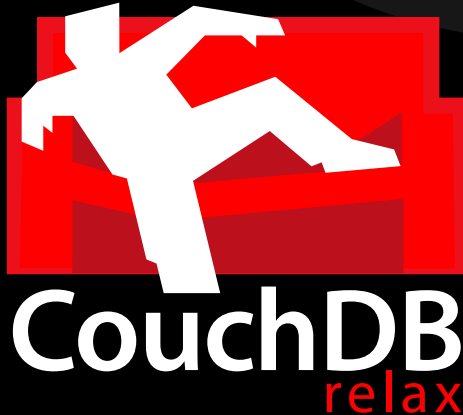
- ▶ Who uses an RDBMS (relational database management system)?
- ▶ Who uses a hash based “database”? (MemcacheDB, ...)
- ▶ Who uses a document based database?
  - ▶ Amazon S3 (SimpleDB)
  - ▶ StrokeDB (Ruby)
  - ▶ FeatherDB (Java port of CouchDB)
  - ▶ CouchDB

Introduction

CouchDB

PHPillow

- ▶ An object orientated client for CouchDB





- ▶ Document based database

- ▶ Document based database
  - ▶ Schema-less

- ▶ Document based database
  - ▶ Schema-less
  - ▶ Store anything you want, even deep structures

- ▶ Document based database
  - ▶ Schema-less
  - ▶ Store anything you want, even deep structures
  - ▶ Arbitrary JSON objects, attachments

- ▶ Document based database
  - ▶ Schema-less
  - ▶ Store anything you want, even deep structures
  - ▶ Arbitrary JSON objects, attachments
- ▶ Apache Project

- ▶ Document based database
  - ▶ Schema-less
  - ▶ Store anything you want, even deep structures
  - ▶ Arbitrary JSON objects, attachments
- ▶ Apache Project
- ▶ ACID compliant

- ▶ Document based database
  - ▶ Schema-less
  - ▶ Store anything you want, even deep structures
  - ▶ Arbitrary JSON objects, attachments
- ▶ Apache Project
- ▶ ACID compliant
- ▶ Trivial replication

- ▶ Document based database
  - ▶ Schema-less
  - ▶ Store anything you want, even deep structures
  - ▶ Arbitrary JSON objects, attachments
- ▶ Apache Project
- ▶ ACID compliant
- ▶ Trivial replication
- ▶ MVCC (Multiversion Concurrency Control)



- ▶ Document based database
  - ▶ Schema-less
  - ▶ Store anything you want, even deep structures
  - ▶ Arbitrary JSON objects, attachments
- ▶ Apache Project
- ▶ ACID compliant
- ▶ Trivial replication
- ▶ MVCC (Multiversion Concurrency Control)
- ▶ Communicates with clients using HTTP

- ▶ Written in/for Erlang/OTP

- ▶ Written in/for Erlang/OTP
  - ▶ Rocks! ... why?

- ▶ Written in/for Erlang/OTP
  - ▶ Rocks! ... why?
- ▶ Scales nearly with amount of processors

- ▶ Written in/for Erlang/OTP
  - ▶ Rocks! ... why?
- ▶ Scales nearly with amount of processors
- ▶ Highly fault tolerant (9 nines)
  - ▶ Live updates of application code

- ▶ Written in/for Erlang/OTP
  - ▶ Rocks! ... why?
- ▶ Scales nearly with amount of processors
- ▶ Highly fault tolerant (9 nines)
  - ▶ Live updates of application code
- ▶ Developed by Ericsson

```
1 $ curl -I -X PUT http://localhost:5984/vikinger
2 HTTP/1.1 201 Created
3 Server: CouchDB/0.10.0 a773833 (Erlang OTP/R12B)
4 Content-Type: text/plain; charset=utf-8
5
6 {"ok": true}
```

```
1 $ curl -I -X GET http://localhost:5984/vikinger
2 HTTP/1.1 200 OK
3 Server: CouchDB/0.10.0a773833 (Erlang OTP/R12B)
4 Content-Type: text/plain; charset=utf-8
5
6 {
7   "db_name": "vikinger",
8   "doc_count": 0,
9   "doc_del_count": 0,
10  "update_seq": 0,
11  "purge_seq": 0,
12  "compact_running": false,
13  "disk_size": 4096,
14  "instance_start_time": "1243673303593969",
15  "disk_format_version": 2
16 }
```



```
1 $ curl -I -X GET http://localhost:5984/unknown
2 HTTP/1.1 404 Object Not Found
3 Server: CouchDB/0.10.0a773833 (Erlang OTP/R12B)
4 Content-Type: text/plain; charset=utf-8
5
6 {"error": "not_found", "reason": "Missing"}
```

► Map-reduce implementations for views

```
1 return array_reduce(  
2     array_map(  
3         function ( $character )  
4             {  
5                 return ord( $character );  
6             },  
7         str_split( $string , 1 )  
8     ),  
9     function ( $sum, $character )  
10        {  
11            return $sum + $character;  
12        }  
13 ) % 16;
```

Introduction

CouchDB

PHPillow

- ▶ Object-orientated client for CouchDB
- ▶ PHP  $\geq$  5.2 since last release (5.3 only before)
- ▶ 96% test coverage

- ▶ Object-orientated client for CouchDB
- ▶ PHP  $\geq$  5.2 since last release (5.3 only before)
- ▶ 96% test coverage
  - ▶ Which I can't show, because PHPUnit is not yet installed :)

- ▶ Object-orientated client for CouchDB
- ▶ PHP  $\geq$  5.2 since last release (5.3 only before)
- ▶ 96% test coverage
  - ▶ Which I can't show, because PHPUnit is not yet installed :)
- ▶ Still in alpha state

- ▶ Object-orientated client for CouchDB
- ▶ PHP  $\geq$  5.2 since last release (5.3 only before)
- ▶ 96% test coverage
  - ▶ Which I can't show, because PHPUnit is not yet installed :)
- ▶ Still in alpha state
  - ▶ Since CouchDB is still "alpha"

- ▶ Issue tracker views from arbit
  - ▶ *The next generation issue tracking system ;)*

```
1 // Create a document
2 $doc = new phpillowUserDocument();
3 $doc->login = 'kore';
4 $doc->name = 'Kore_Nordmann';
5 $doc->data = array(
6     'mail' => "kore@php.net",
7     // ...
8 );
9 $id = $doc->save();
10
11 // Fetch a document by ID
12 $doc = new phpillowUserDocument( $id );
```



- ▶ Issue tracker views from arbit
  - ▶ *The next generation issue tracking system ;)*

- ▶ Apache CouchDB: <http://couchdb.org/>
- ▶ PHPillow:  
<http://kore-nordmann.de/projects/phpillow/>
- ▶ CouchDB use case: [http://kore-nordmann.de/blog/phpillow\\_php\\_couchdb\\_wrapper.html](http://kore-nordmann.de/blog/phpillow_php_couchdb_wrapper.html)
  - ▶ Implementing a user permission system using CouchDB

- ▶ Open questions?
- ▶ Further remarks?
- ▶ Contact
  - ▶ Mail: <kore@php.net>
  - ▶ Web: <http://kore-nordmann.de/> (Slides will be available here soonish)
  - ▶ Twitter: <http://twitter.com/koredn>