



CouchDB:

HTTP and JSON Persistence

Scott Davis, ThirstyHead.com



ThirstyHead.com

training done right.



ThirstyHead.com

training done right.

The
Pragmatic
Programmers

GIS For Web Developers

Adding *Where* to Your
Web Applications



The
Pragmatic
Programmers

Groovy Recipes

*Greasing
the Wheels
of Java*



Scott Davis
Jason Rudolph



Getting Started with Grails Second Edition

A modern web framework for the Java platform

ENTERPRISE SOFTWARE
DEVELOPMENT SERIES

InfoQ



Scott Davis

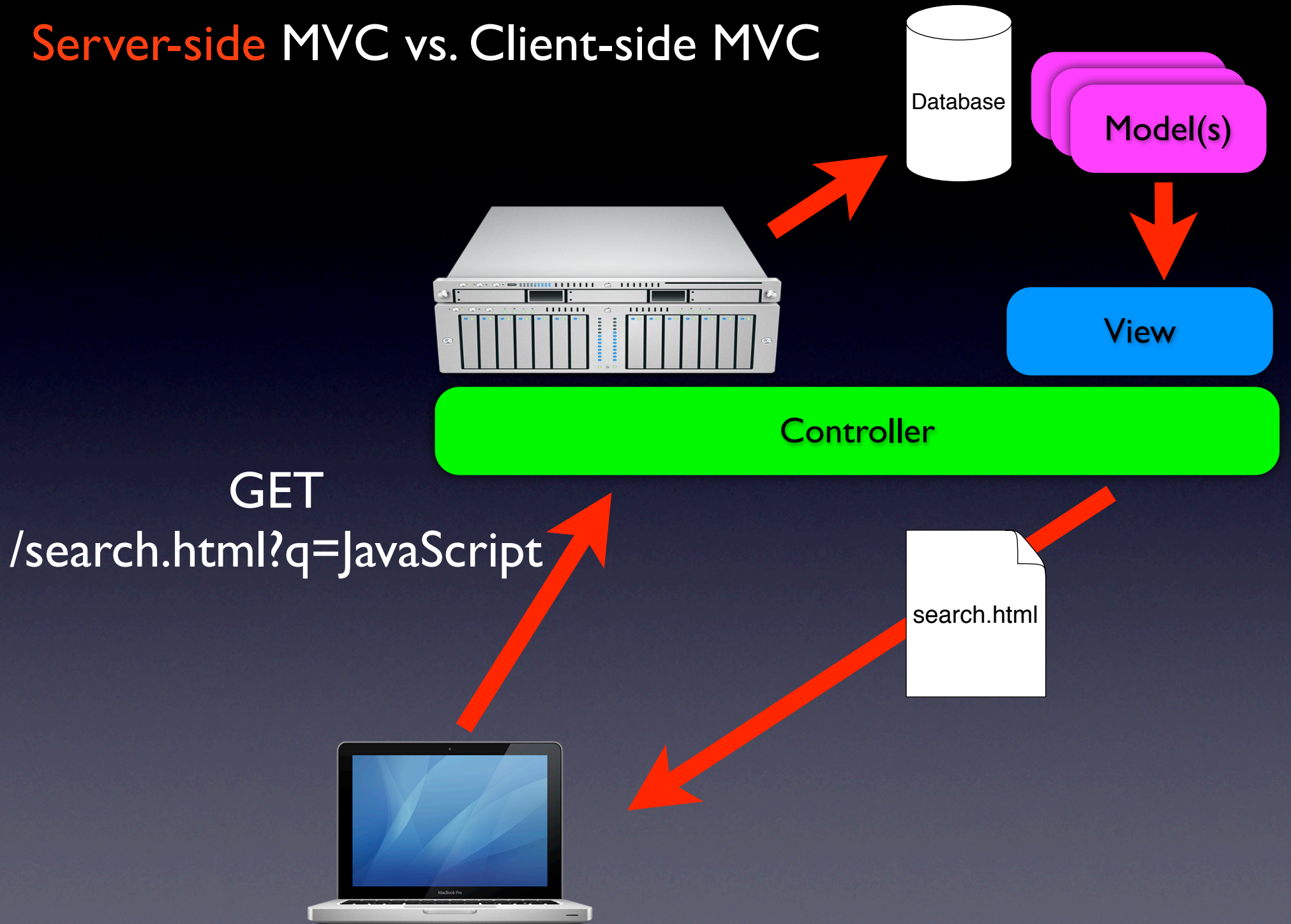
@scottdavis99

HTML

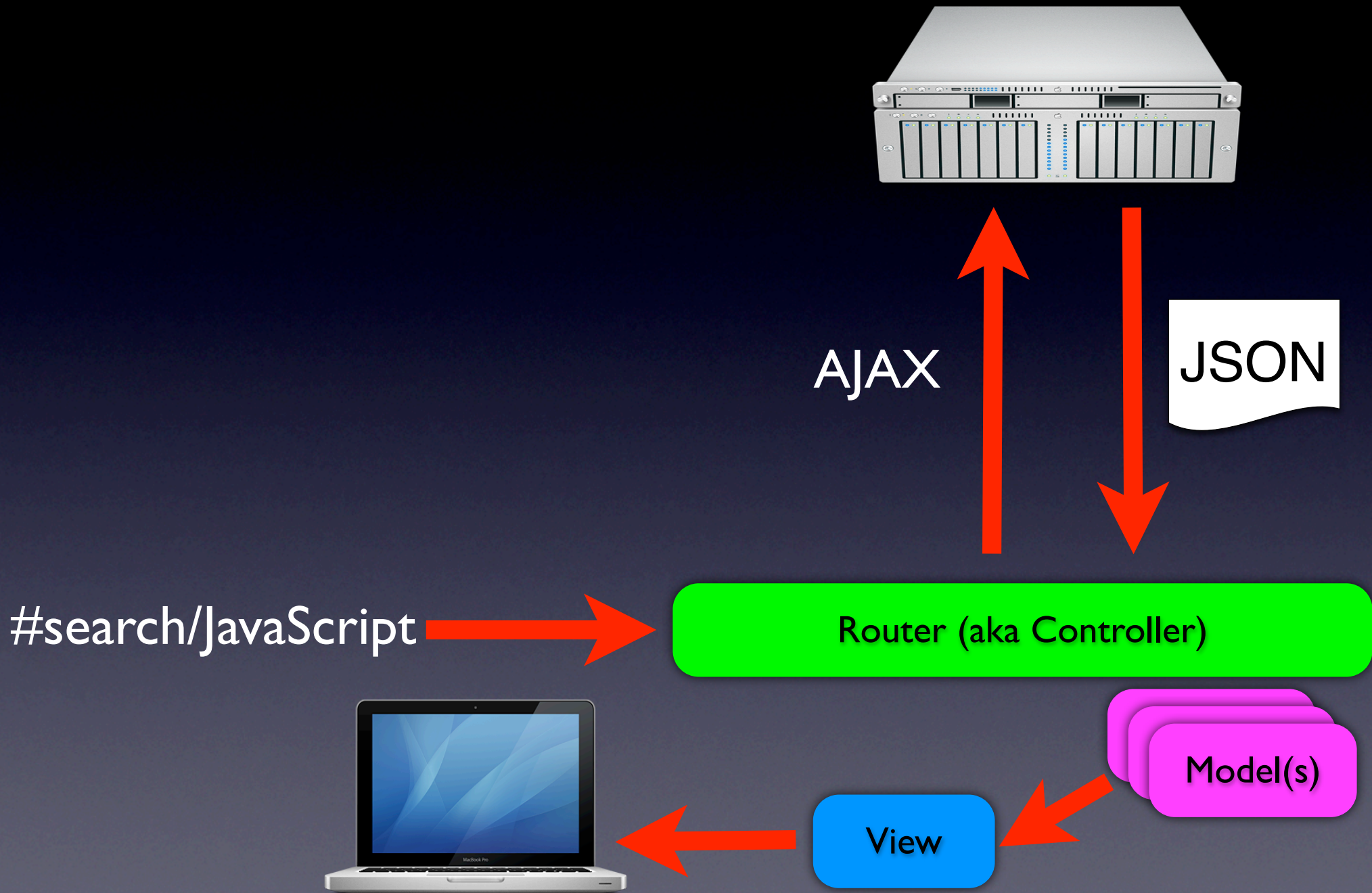


<html>

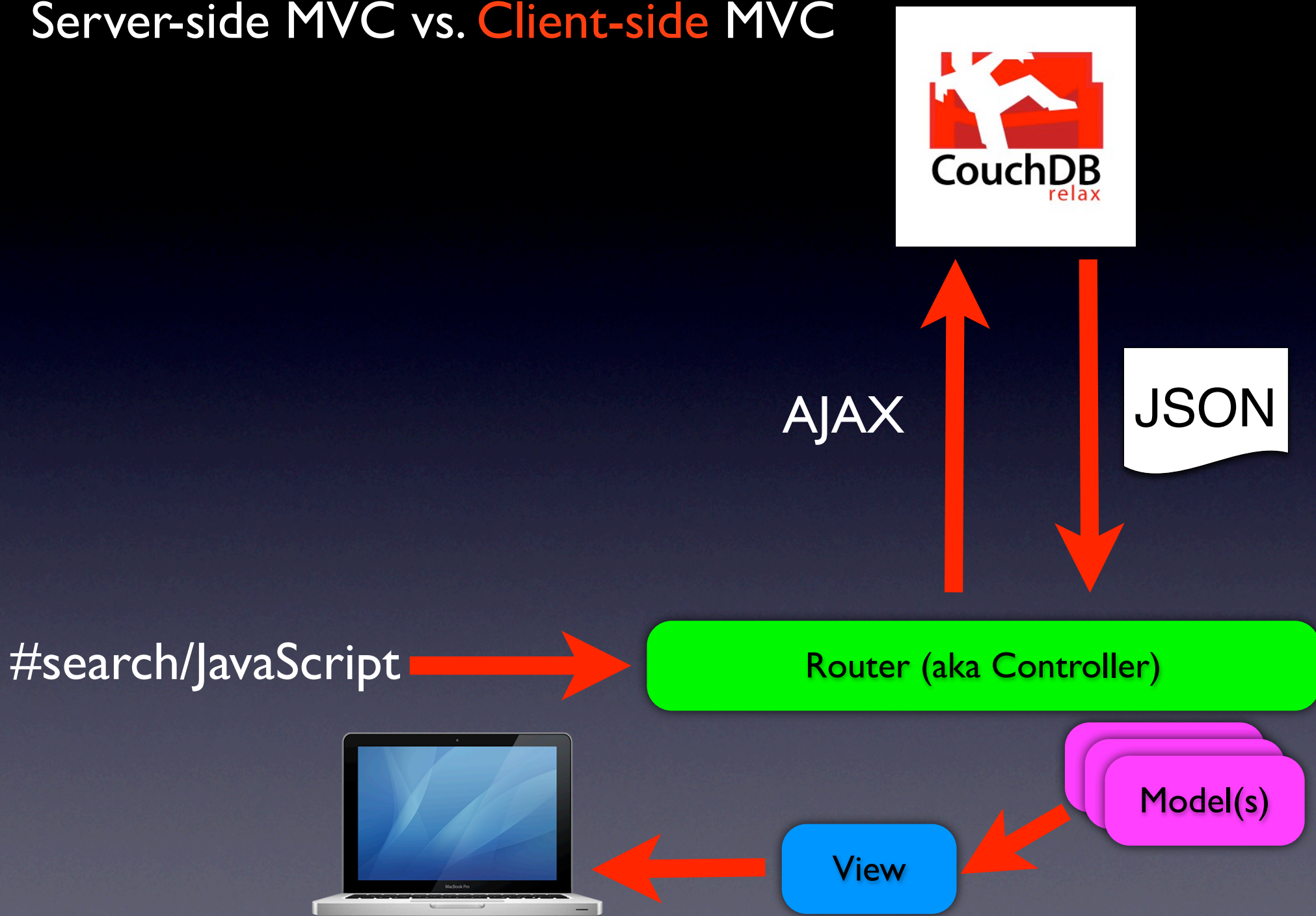
Server-side MVC vs. Client-side MVC

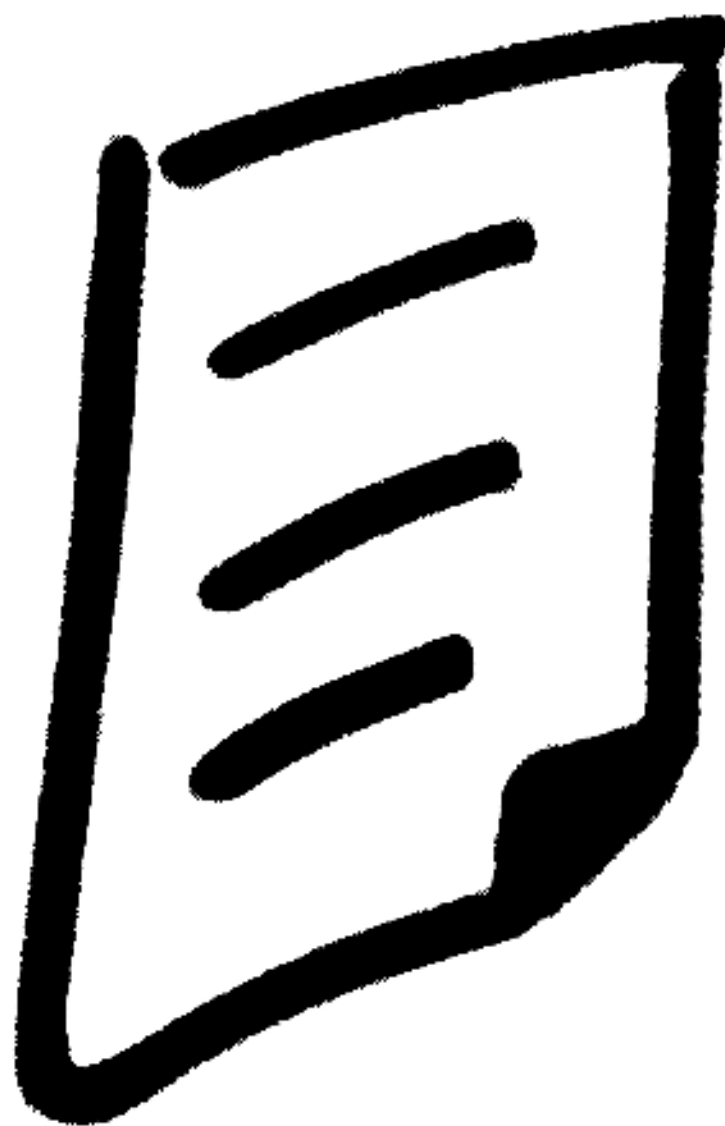
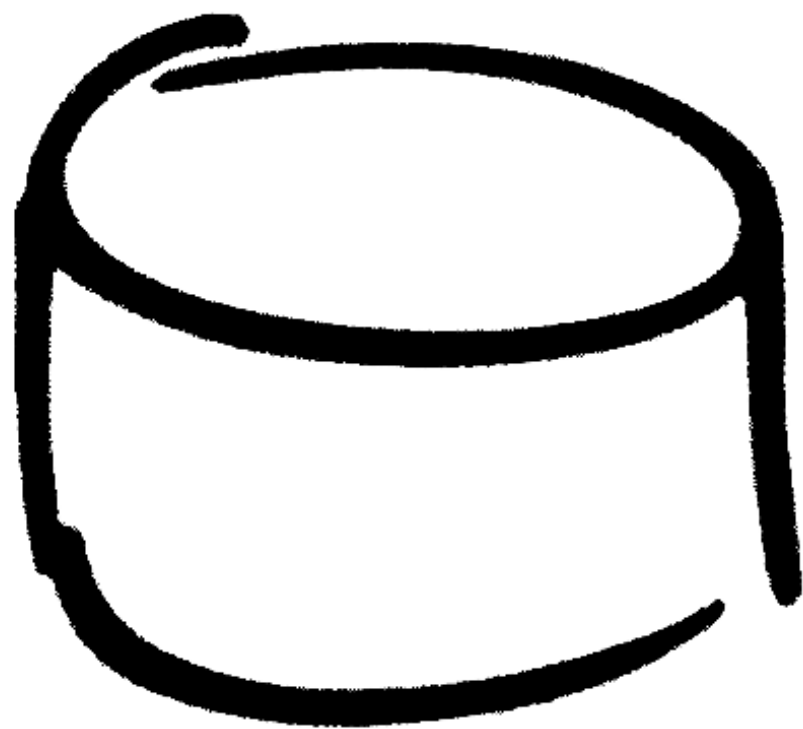


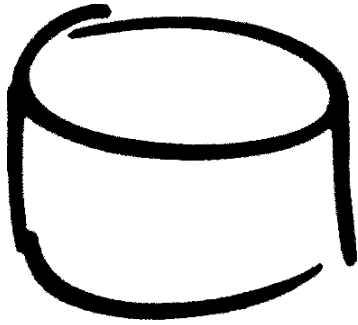
Server-side MVC vs. Client-side MVC



Server-side MVC vs. **Client-side** MVC







Row
oriented

SQL

INSERT, SELECT,
UPDATE, DELETE



Document
oriented

JavaScript

PUT, GET,
POST, DELETE

(REST)



[NoSQL]

The
Pragmatic
Programmers

Seven Databases in Seven Weeks

A Guide to Modern Databases
and the NoSQL Movement

Eric Redmond
and Jim R. Wilson

Edited by Jacquelyn Carter



Time to Relax



CouchDB

The Definitive Guide

O'REILLY®

*J. Chris Anderson,
Jan Lehnardt & Noah Slater*

New Open Source Tools for Location Data

Getting Started with

GEO, CouchDB & Node.js



Indexing and Querying Documents

Writing and Querying

MapReduce Views in CouchDB



O'REILLY®

Bradley Holt

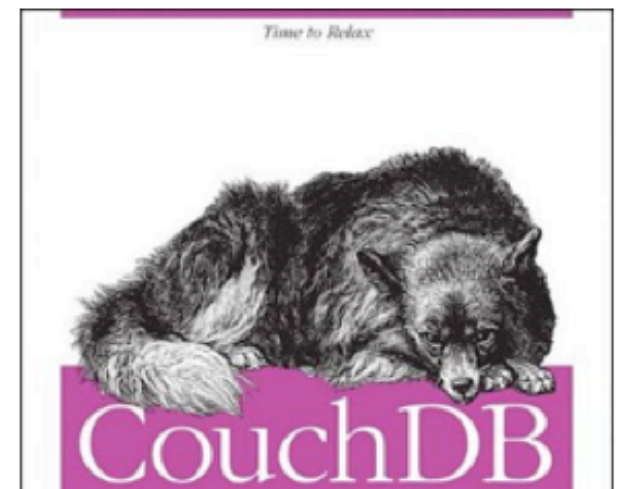
<http://guide.couchdb.org>

CouchDB: The Definitive Guide

About the Book

This is the home of a free [O'Reilly Media](#) book about [Apache CouchDB](#).

The book is designed to guide you gently through using CouchDB with clear but practical scenarios. We progressively showcase key features, starting with simple document CRUD, working through to advanced MapReduce, and culminate with deployment tuning for performance and reliability.



Installation



[About](#)

[Contribute](#)

[Mailing List](#)

[Download](#)

[Quick Links](#)



Apache CouchDB™ is a database
that uses **JSON** for documents,
JavaScript for **MapReduce** queries,
and regular **HTTP** for an **API**

DOWNLOAD
Version 1.2



A Database for the Web

CouchDB is a database that completely embraces the web. Store your data with JSON documents. Access your documents with your web browser, via HTTP. Query, combine, and transform your documents with JavaScript. CouchDB works well with modern web and mobile apps. You can even serve web apps

Installing CouchDB

On Linux:

```
$ sudo apt-get install couchdb
```

On Mac OS X:

```
$ sudo brew install couchdb
```

On Windows:

[download the installer]

Starting CouchDB

```
$ sudo couchdb
```

```
Apache CouchDB 1.0.0 (LogLevel=info) is  
starting.
```

```
Apache CouchDB has started. Time to relax.
```

```
[info] [<0.31.0>] Apache CouchDB has  
started on http://127.0.0.1:5984/
```

Kicking the tires...

Ping

```
curl http://127.0.0.1:5984/  
{"couchdb": "Welcome", "version": "1.0.0"}
```

Get all databases

```
curl -X GET http://127.0.0.1:5984/_all_dbs  
["_users"]
```

Overview

[+ Create Database ...](#)

Name	Size	Number of Documents	Update Seq
_users	4.1 KB	1	1
albums	6.6 KB	3	6

Showing 1-2 of 2 databases

[← Previous Page](#) | Rows per page: | [Next Page →](#)

Tools

- [Overview](#)
- [Configuration](#)
- [Replicator](#)
- [Status](#)
- [Test Suite](#)

Recent Databases

Welcome to Admin Party!
Everyone is admin. [Fix this](#)

Futon on Apache CouchDB 1.0.0

The full

CRUD

experience...

Create a new database (PUT)

Create a database

```
curl -X PUT http://127.0.0.1:5984/albums  
{"ok":true}
```

Create a new database (cont.)

CONFLICT: tried to create an existing database

```
-----  
curl -X PUT http://127.0.0.1:5984/albums  
{ "error": "file_exists", "reason": "The  
database could not be created, the file  
already exists." }
```

Delete a database

```
-----  
curl -X DELETE http://127.0.0.1:5984/albums  
{ "ok": true }
```


Create a document

```
curl -X PUT http://127.0.0.1:5984/albums/2  
-d '{"title": "Revolver", "artist": "The  
Beatles"}'
```

```
{"ok": true, "id": "2", "rev": "1-5e963faf3d2591  
4faecaf772455b0aac"}
```

Create a document from a file

```
curl -X PUT http://127.0.0.1:5984/albums/3  
-d @abbeyRoad.json
```

```
{"ok": true, "id": "3", "rev": "3-  
a94e7594c59864efcb3a7d4afe4aad4f"}
```

Select (GET)

Retrieve a document

```
curl -X GET http://127.0.0.1:5984/albums/2
```

```
{"_id": "2", "_rev": "1-5e963faf3d25914faecaf7  
72455b0aac", "title": "Revolver", "artist": "Th  
e Beatles"}
```

update (PUT)

Update a document

```
curl -X GET http://127.0.0.1:5984/albums/2
```

NOTE: you must know both the id and the rev
to update a document

```
curl -X PUT http://127.0.0.1:5984/albums/2?  
rev=1-5e963faf3d25914faecaf772455b0aac
```

Delete (DELETE)

Delete a document

```
curl -X GET http://127.0.0.1:5984/albums/2
```

NOTE: you must know both the id and the rev
to delete a document

```
curl -X DELETE http://127.0.0.1:5984/  
albums/2?  
rev=1-5e963faf3d25914faecaf772455b0aac
```

Get all docs

Get all docs (id only)

```
curl -X GET http://127.0.0.1:5984/_all_docs
```

Get all docs (full data)

```
curl -X GET http://127.0.0.1:5984/_all_docs?include_docs=true
```


Get UUID(s)

Get a UUID

```
curl -X GET http://127.0.0.1:5984/_uuids  
{ "uuids":  
  ["1e1b132da27ff00f8860f47df60013ad"] }
```

Get multiple UUIDs

```
curl -X GET http://127.0.0.1:5984/_uuids?  
count=3  
{ "uuids":  
  ["1e1b132da27ff00f8860f47df600205a", "1e1b13  
2da27ff00f8860f47df60027e4", "1e1b132da27ff0  
0f8860f47df60033c3"] }
```


Map / Reduce

SQL

```
select * from albums  
where artist = "The Beatles";
```

Map



Reduce



Overview > albums

+ New Document ⓘ Security...
⊗ Compact & Cleanup... ⊗ Delete Database...

Jump to: View: Stale views ☐

▼ View Code

Map Function:

```
function(doc) {  
  if ('artist' in doc) {  
    emit(doc.artist, doc._id);  
  }  
}
```

Reduce Function (optional):

Language:

Warning: Please note that temporary views are not suitable for use in production, as they are really slow for any database with more than a few dozen documents. You can use a temporary view to experiment with view functions, but switch to a permanent view before using them in an application.

Key ▲	Value
"Grateful Dead" ID: 5	"5"
"Led Zeppelin" ID: 4	"4"
"The Beatles" ID: 2	"2"
"The Beatles" ID: 3	"3"

Showing 1-4 of 4 rows

← Previous Page | Rows per page: | Next Page →



Tools

[Overview](#)
[Configuration](#)
[Replicator](#)
[Status](#)
[Test Suite](#)

Recent Databases

[albums](#)
[pixeltarget](#)

Overview > albums

New Document

Security...

Compact & Cleanup...

Delete Database...

Jump to:

View:

Stale views ☐

View Code

Map Function:

function(doc) {
 if ('artist' in doc) {
 emit(doc.artist, doc._id);
 }
}

Reduce Function (optional):

Run

Language:

=

Revert

Save As...

Save

Warning: Please note that temporary views are not suitable for use in production, as they are really slow for any database with more than a few dozen documents. You can use a temporary view for testing and debugging, but not for serving data in an application.

Key ▲

"Grateful Dead"

ID: 5

"Led Zeppelin"

ID: 4

"The Beatles"

ID: 2

"The Beatles"

ID: 3

Showing 1-4 of 4 rows

← Previous Page

Rows per page:

Next Page →

CouchDB

relax

Tools

Overview

Configuration

Replicator

Status

Test Suite

Recent Databases

albums

pixeltarget

Welcome to Admin Party!

Everyone is admin. [Fix this.](#)

Save View As...

You can save this function code as a permanent view in the database. Just enter or select the design document and the name of the view below. Note that if you choose an existing view, it will be overwritten!

Design Document:

View Name:

Save

Cancel


```
{
  total_rows: 4,
  offset: 0,
  rows: [
    - {
      id: "5",
      key: "Grateful Dead",
      value: "5"
    },
    - {
      id: "4",
      key: "Led Zeppelin",
      value: "4"
    },
    - {
      id: "2",
      key: "The Beatles",
      value: "2"
    },
    - {
      id: "3",
      key: "The Beatles",
      value: "3"
    }
  ]
}
```

By key:

By key:

```
curl http://localhost:5984/albums/_design/  
artist/_view/by_name?key="Grateful Dead"
```

Limit returned results:

```
Limit returned results
```

```
-----
```

```
curl http://localhost:5984/albums/_design/  
artist/_view/by_name?key="The Beatles"&limit=1
```

Return a range:

Return a range:

```
curl http://localhost:5984/albums/_design/  
artist/_view/by_name?  
startkey="F"&endkey="M"
```




CouchDB:

HTTP and JSON Persistence

Scott Davis, ThirstyHead.com



ThirstyHead.com
training done right.

scott@thirstyhead.com



Questions?

Thanks for your time.

source and slides:

<http://scottdavis99.github.com/couchdb/>

Image Credits

All images courtesy of <http://morgueFile.com>