

CouchDB

CS 5800
Utah State University

Outline

- Introduction
- JSON
- Creating a data collection
- Queries

CouchDB

R-2

Features

- Apache project, begun in 2005, now version 1.5
- Open source, supported on many platforms
- Document store, JSON documents
- NoSQL – Map/Reduce views (queries)
- For Big Data – distributed, replicated data store
- REST API – access via URIs and HTTP
- Futon – Browser-based GUI
- Used by: Ubuntu, BBC, Credit Suisse, ...
- Similar: MongoDB

CouchDB

R-3

JavaScript Object Notation (JSON)

- Text-based representation of objects
- Object is a key-value set
 - Key – a string (name of an instance variable in object)
 - Value
 - Number
 - String
 - Boolean
 - Array
 - Object

CouchDB

R-4

Example – Person Object

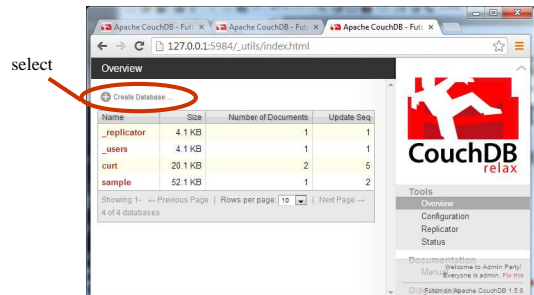
```
{  
  "name" : "Jose",  
  "age" : 39,  
  "aliases" : ["Joe", "Josinho" ]  
  "best friend" : {  
    "name" : "Mark",  
    "address" : "23 Wildflower Way"  
  }  
}
```

CouchDB

R-5

Creating a Database

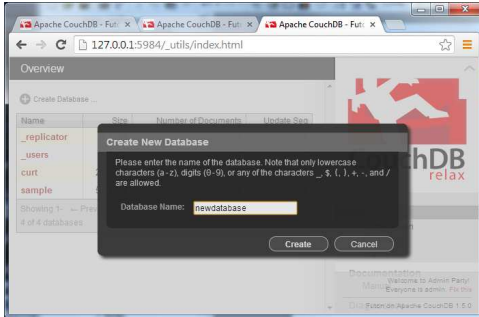
- A database is a collection of JSON documents
- Can use HTTP, see documentation
- Using Futon



CouchDB

R-6

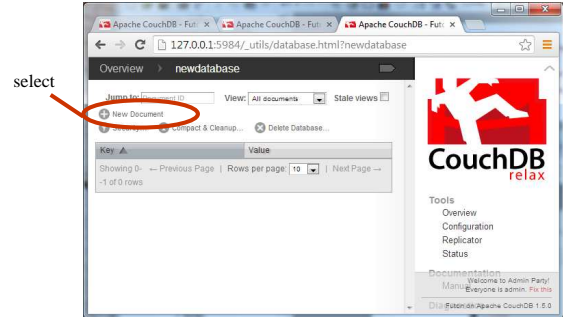
Futon – Create Database continues



CouchDB

R-7

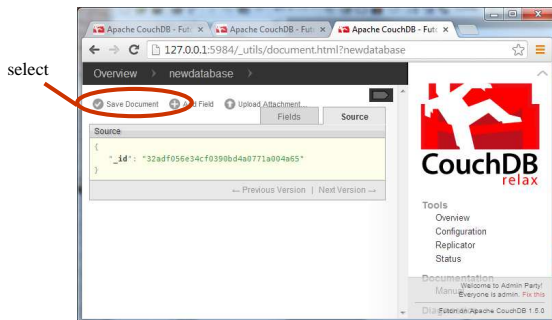
Futon – Create Document



CouchDB

R-8

Futon – Auto Generated ID



CouchDB

R-9

Futon – Revision 1, first time stored



CouchDB

R-10

Editing Data

- In Futon
 - Source tab– JSON source, can be edited, adding fields
 - Field tab – GUI for entering a field at a time
- Can delete documents, databases

CouchDB

R-11

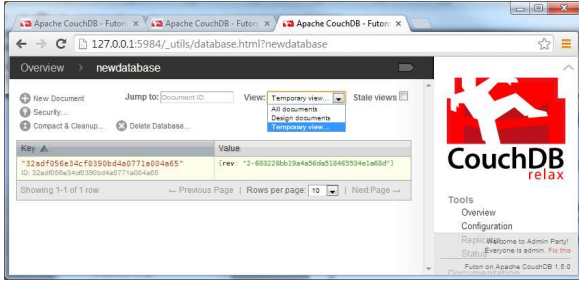
Views

- A view is an “intensional document”
- Generated by JavaScript Map/Reduce functions
 - Map – Convert data to a list
 - Reduce – Apply function to values in list
- Stored in the database as JSON

CouchDB

R-12

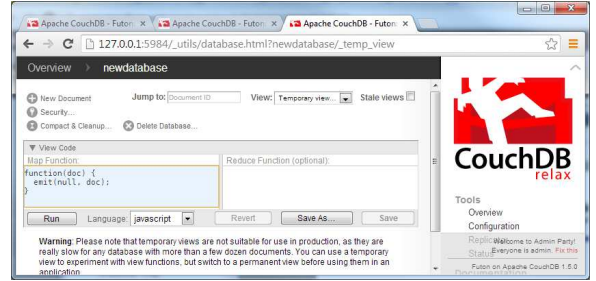
Creating a View in Futon



CouchDB

R-13

View had Map and Reduce Code

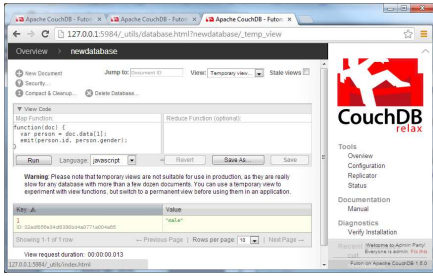


CouchDB

R-14

Emit First Person Object in Data Field

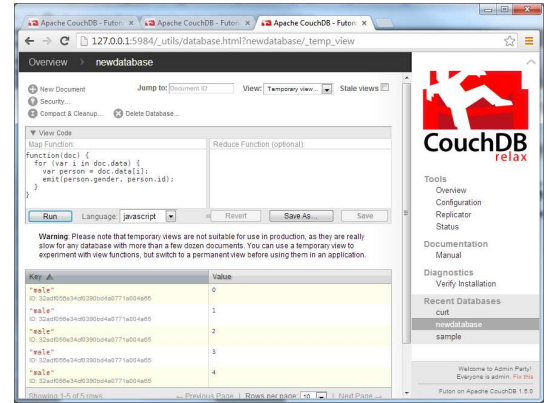
- Use “dot” notation to navigate doc.data[1].id
- Use “emit” function to emit key,value pair



CouchDB

R-15

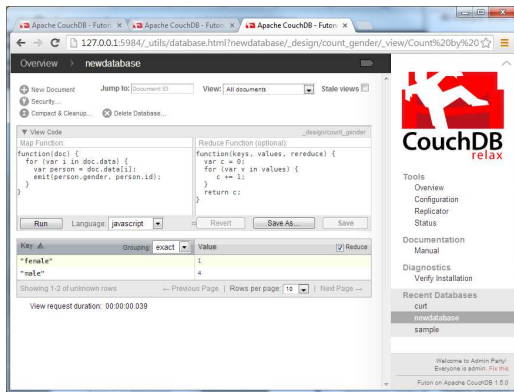
Emit Person (Gender, ID) Pairs



CouchDB

R-16

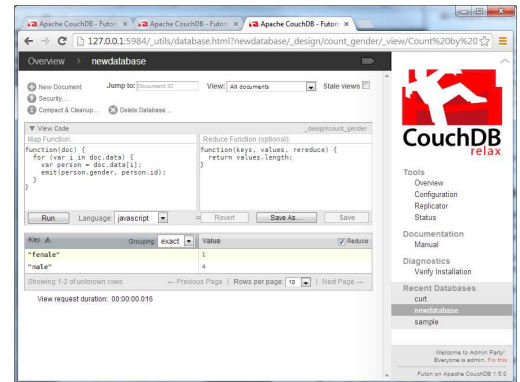
Count Number of People per Gender



CouchDB

R-17

Count Number of People per Gender



CouchDB

R-18

Count Number of People per Gender

The screenshot shows the Apache CouchDB Futon interface for a database named 'newdatabase'. A MapReduce job is configured with the following code:

```

Map Function:
function(doc) {
  for (var i in doc.data) {
    var person = doc.data[i];
    emit(person.gender, person.id);
  }
}

Reduce Function (optional):
function(keys, values, reduce) {
  return values.length;
}
    
```

The results table shows the following data:

Key	Value
"female"	1
"male"	4

CouchDB

R-19

Count Active People per Gender

The screenshot shows the Apache CouchDB Futon interface for a database named 'newdatabase'. A MapReduce job is configured with the following code:

```

Map Function:
function(doc) {
  for (var i in doc.data) {
    var person = doc.data[i];
    emit(person.gender, person.isActive);
  }
}

Reduce Function (optional):
function(keys, values, reduce) {
  return values.length;
}
    
```

The results table shows the following data:

Key	Value
["female", true]	1
["male", false]	3
["male", true]	1

CouchDB

R-20

Filter non-Active People, then count

The screenshot shows the Apache CouchDB Futon interface for a database named 'newdatabase'. A MapReduce job is configured with the following code:

```

Map Function:
function(doc) {
  for (var i in doc.data) {
    var person = doc.data[i];
    if (person.isActive) {
      emit(person.gender, 1);
    }
  }
}

Reduce Function (optional):
function(keys, values, reduce) {
  return sum(values);
}
    
```

The results table shows the following data:

Key	Value
"female"	1
"male"	1

CouchDB

R-21

View Stored as a Document

The screenshot shows the Apache CouchDB Futon interface for a document view of the filtered data. The document content is as follows:

```

{
  "_id": "_design/count_gender_filter",
  "_rev": "2-78ec7fc939f2d9102237f28899e041",
  "language": "javascript",
  "views": {
    "filter then count": {
      "map": "function(doc) {\n  for (var i in doc.data) {\n    var person = doc.data[i];\n    if (person.isActive) {\n      emit(person.gender, 1);\n    }\n  }\n}\n",
      "reduce": "function(keys, values, reduce) {\n  return sum(values);\n}"
    }
  }
}
    
```

CouchDB

R-22

Summary

- Big data
- Document store
- Map/Reduce
- REST API

CouchDB

R-23