Couchbase

Using the Couchbase C/C++ Client Library

Workshop Day 2

https://github.com/dufrenoyl/cb-workshop-2d

Before we begin

- Make sure that Couchbase Server is installed on the Dev Machine!

Document Modelling Basics

JSON

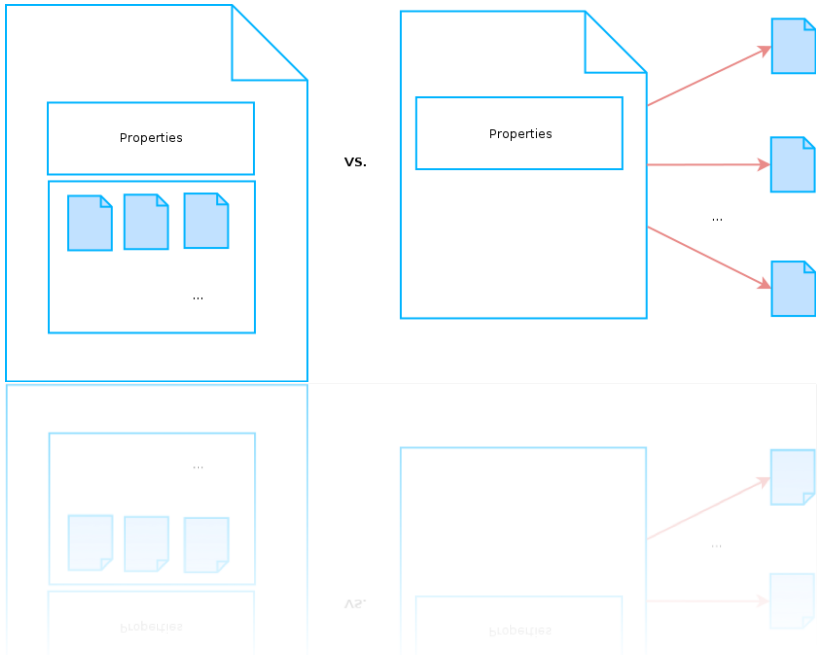
- Java Script Object Notation
  - Meta data
  - Document Value

```
"meta" :
{
  "id" : "person::david",
  "rev" : "1-0002bce0000000000",
  "flags" : 0,
  "expiration":0,
  "type":"json"
}

"doc" :
{
  "type" : "person",
  "uid":"david",
  "firstname":"David",
  "lastname":"Maier",
  "birthday": 330004800000,
  "email":"david.maier@couchbase.com"
}
```

Normalization vs. De-Normalization

- Normalized
  - Uses key references for 1-many relationships
  - Reduces data duplicates
  - Smaller document size
- De-Normalized
  - Uses nested documents
  - Aggregated view of data
  - Allows atomic access
  - No client side joins



Normalization vs. De-Normalization

DE-NORMALIZED

```
{
  "type" : "organization",
  "oid" : "CB",
  "name" : "Couchbase",
  "street" : "2440 West El Camino Real Suite 101",
  "city" : "Mountain View",
  "state" : "California"
  "employees" :
    [
      {
        "uid":"david",
        "firstname":"David",
        "lastname":"Maier",
        "birthday": 1402920000000,
        "email":"david.maier@couchbase.com"
      },
      ...
    ]
}
```

NORMALIZED

```
{
  "type" : "organization",
  "name" : "Couchbase",
  "street" : "2440 West El Camino Real Suite 101",
  "city" : "Mountain View",
  "state" : "California"
  "employees" : ["person::david", "person::perry", "person::dipti", ... ]
}
```

Atomic Counters

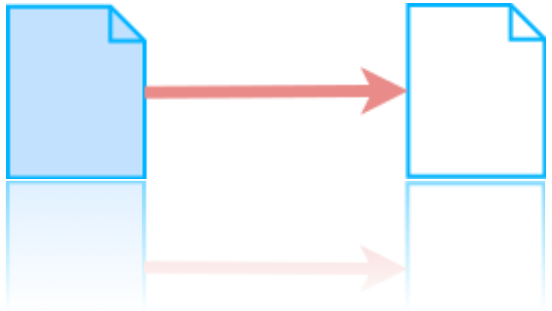
- Similar to sequences / auto-incrementing columns from the relational world
- Initialize and increment a counter value
- Use the counter as part of the key


```
id = client.incr("count::person");
```

```
client.add("person::" + id, doc);
```

Reference Documents for Lookups

- Second document which references the primary one
- Needs to be maintained by the application



```
"email::david.maier@couchbase.com" : { "ref" : "person::david" }
```

Managing Connections

Exercise 7

libcouchbase Installation

Perform the following steps in order to install libcouchbase

- Perl needs to be installed

http://developer.couchbase.com/documentation/server/current/sdk/c/start-using-sdk.html

```
su root
wget http://packages.couchbase.com/releases/couchbase-release/couchbase-release-1.0-2-x86_64.rpm
rpm -iv couchbase-release-1.0-2-x86_64.rpm
yum install libcouchbase-devel libcouchbase2-bin
```

- The described setup procedure adds the the Couchbase package repository (/etc/yum.repos.d) and then installs the packages 'libcouchbase2-bin' and 'libcouchbase-devel'.

Get the Workshop Sources

Perform the following steps in order to check out the latest source code

- New installation

```
git clone https://github.com/dufrenoyl/cb-workshop-2d.git
```

- Preinstalled workshop machine

```
cd ~/Git/cb-workshop-2d
git rebase
```

Before we begin

Open the documentation for libcouchbase!

- https://developer.couchbase.com/documentation/server/current/sdk/c/sample-app-backend.html
- https://developer.couchbase.com/documentation/server/current/sdk/c/start-using-sdk.html

- Open the TravelAppSample project also => This is the solution.
- Open the TravelAppSample-Empty project => This is where you start.
- Check the provided helper classes:

  CouchbaseDocument
  CBCookie*
  CBQStringConvert

Connecting to Couchbase

Implement the following methods in CBDataSource:

- void Connect(QString connectionString, QString password);

Implement the following methods in CBDataSourceFactory:

- static void Create(QString connectionString, QString password);

Test your implementation by executing:

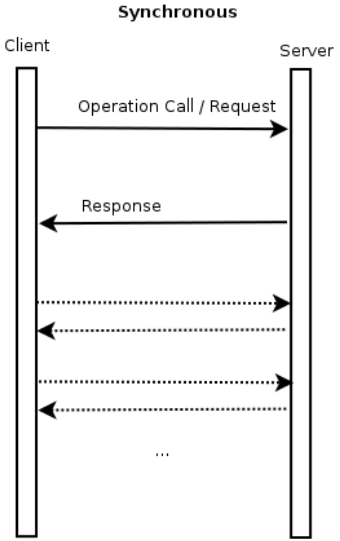
```
DemoCouchbaseConnect connectDemo;
connectDemo.test();
```

Understanding Non-Blocking I/O

in libcouchbase

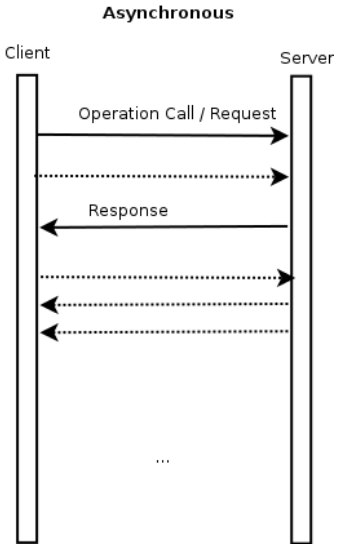
Blocking

- Libcouchbase is designed to use non-blocking I/O
  - Scheduled operations
- But lcb_wait() blocks by default
  - Waits for pending requests
  - Used for synchronous operation execution
- Callback functions are used
  - e.g. storage_callback



Non-Blocking

- External event loop integration
  - Provides mechanism to execute a callback function when a specific event occurs
  - e.g. libevent
- Asynchronous operation execution
- No need for lcb_wait()



Working with Documents

Exercise 8 - 11

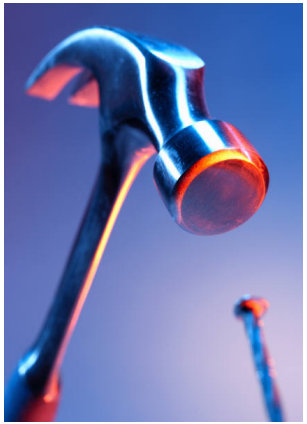
Get a Document

Make sure that the travel-sample data is installed!
Implement the following methods in CBDataSource:

- CouchbaseDocument Get (QString key);

Test your implementation by executing:

```
DemoCouchbaseGet getDemo;
getDemo.test();
```



Perform a Multi-Get

Make sure that the travel-sample data is installed!

Implement the following methods in CBDataSource:

- CouchbaseDocumentMap MultiGet(QStringList keys);

Test your implementation by executing:

```
DemoCouchbaseMultiGet multiGetDemo;
multiGetDemo.test();
```

Create/Update a Document

Implement the following methods in CBDataSource:

- bool Upsert(QString key, QString document)

Test your implementation by executing:

```
DemoCouchbaseUpsert upsertDemo;
upsertDemo.test();
```

Delete a Document

Implement the following methods in CBDataSource:

- bool Delete(QString key);

Test your implementation by executing:

```
DemoCouchbaseDelete deleteDemo;
deleteDemo.test();
```

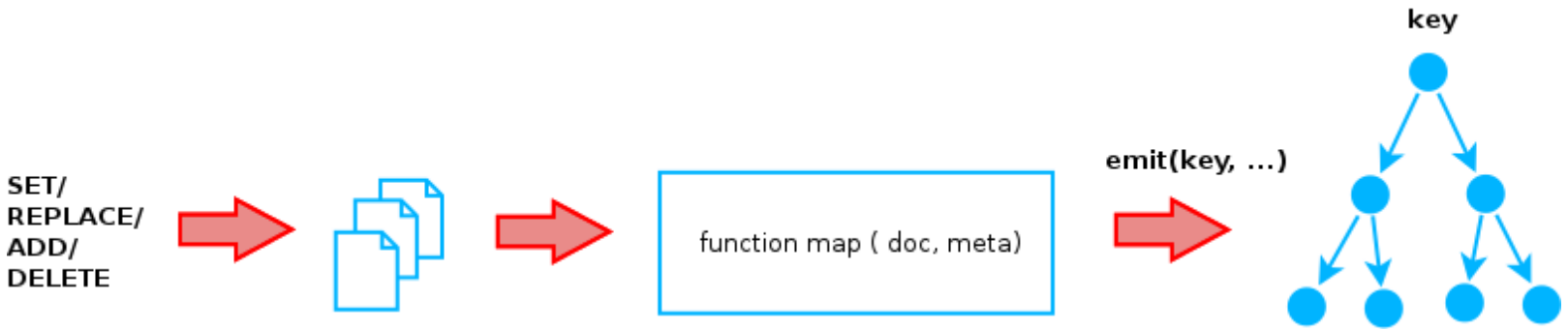
Querying via Views

Exercise 12

Views

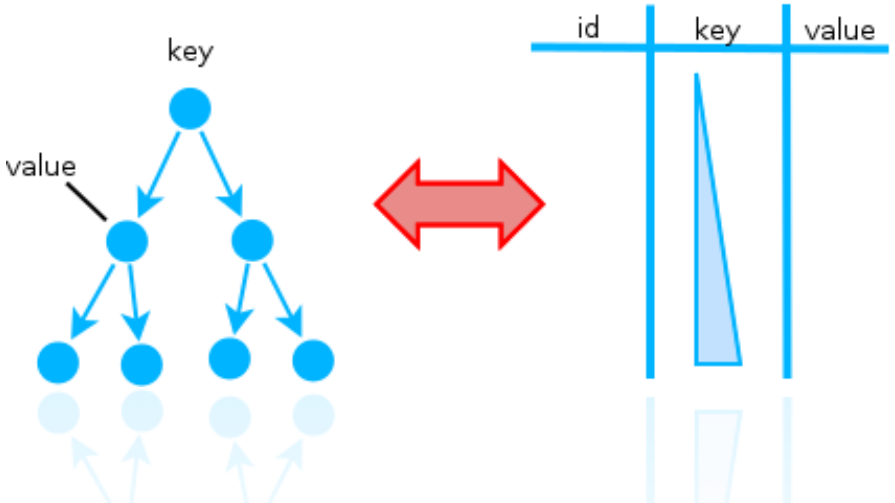
- Organized in Design Documents
- Incremental Map-Reduce
- Spread indexing load across nodes

Map	Reduce
Process, filter, map and emit a row	Aggregate mapped data Built in: _count, _sum, _stats



Views

- Multiple roles
  - A Primary Index to access all document id-s
  - A Secondary Index as an alternative access path
  - A View provides you an alternative view on your data

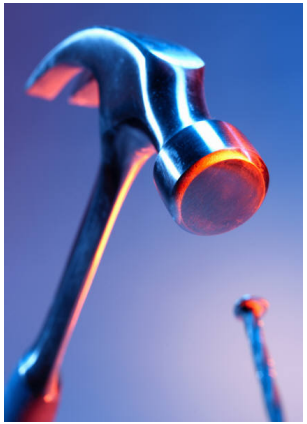


Query a View

Create the View 'airports/by_name' !

Implement the following methods in CBDataSource:

- CBQueryResult QueryView(QString designDocName,
  QString viewName, int limit=0, int skip=0);



Test your implementation by executing:

```
DemoCouchbaseView viewDemo;
viewDemo.test();
```

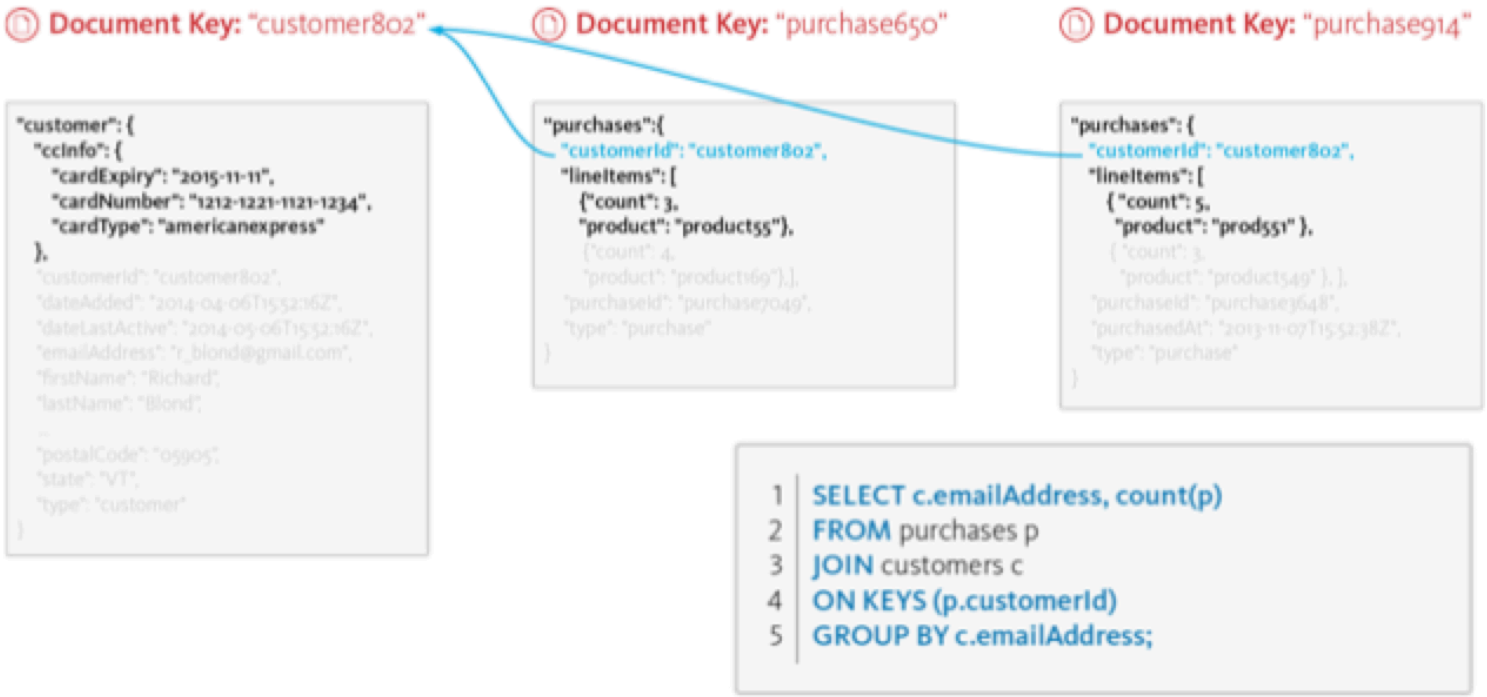
Querying via N1QL

Exercise 13

N1QL Introduction

- Next generation, NoSQL query language
- SQL-like
  - WHERE
  - LIKE
  - GROUP
  - JOINS
- Powerful Extensions for JSON and hierarchical data structures
  - NEST
  - UNNEST
- Multiple access paths
  - Views
  - Global Secondary Indexes
    - Memory Optimized Indexes

N1QL Introduction - Joins



N1QL Query Examples

```
SELECT airportname FROM `travel-sample` WHERE
faa ='LAX'
```

```
SELECT airportname FROM `travel-sample` WHERE
faa ='LHR'
```

```
SELECT faa as fromAirport,geo FROM `travel-sample`
WHERE airportname = 'Los Angeles Intl' UNION
SELECT faa as toAirport,geo FROM `travel-sample`
WHERE airportname = 'Heathrow'
```

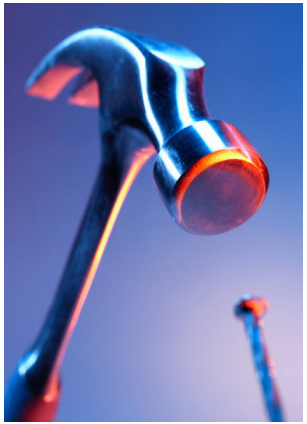
```
SELECT r.id, a.name, s.flight, s.utc, r.sourceairport,
r.destinationairport, r.equipment FROM `travel-sample` r
UNNEST r.schedule s JOIN `travel-sample` a ON KEYS
r.airlineid WHERE r.sourceairport='LHR' AND
r.destinationairport='LAX' AND s.day=6 ORDER BY a.name
```

```
SELECT airportname FROM `travel-sample` WHERE
airportname LIKE 'Los An%'
```

Query via N1QL

Make sure that at least a Primary Index is created!
Also Double check that the Secondary Index on 'faa' is there!
Implement the following methods in CBDataSource:

- CBN1qlResult QueryN1ql(QString query);



Test your implementation by executing:

```
DemoCouchbaseN1ql n1qlDemo;
n1qlDemo.test();
```

Error Handling and Logging

Handling errors

- Operations return lcb_error_t status code
- Check for
  - err == LCB_SUCCESS
- Error Codes
  - <libcouchbase/error.h>
- Examples
  - LCB_KEY_EEXISTS: Key already exists
  - LCB_KEY_ENOENT: Key does not already exist if replacing it
  - LCB_ETIMEDOUT: Transient error which indicates that something took too long
  - LCB_ETMPFAIL: Transient error which indicates that the server was too busy
  - LCB_AUTH_ERROR: Authentication error
  - LCB_BUCKET_ENOENT: Bucket does not exist

Logging

- LCB_LOGLEVEL environment variable
  - 1 – basic
  - 5 – verbose
- Programmatically
  - LCB_CNTL_CONLOGGER_LEVEL setting
  - console_log_level option in the connection string
- Log entry format

```
1ms [I0] {14780} [DEBUG] (lcbio_mgr - L:383) <localhost:11210> (HE=0xe56760)
Creating new connection because none are available in the pool
```

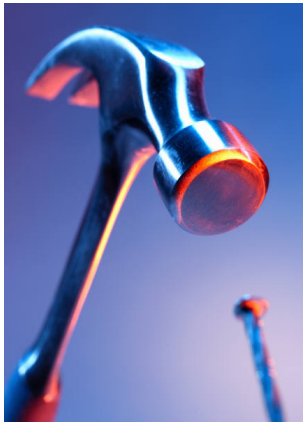
A Sample Application

Exercise 14
https://github.com/dmaier-couchbase/cb-workshop-cpp/tree/master/TravelAppSample

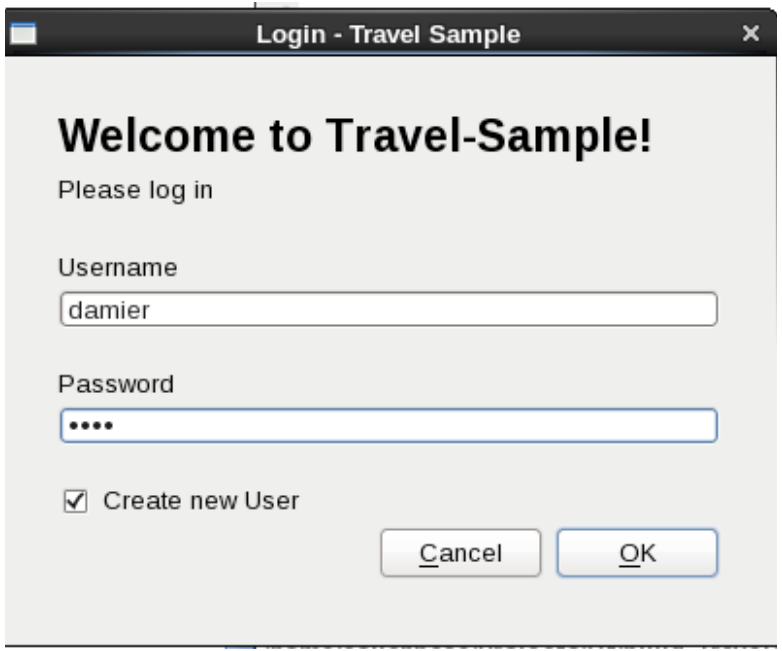
A Sample Application

Inspect the full source code of the Travel-Sample application!
Run the Qt application!

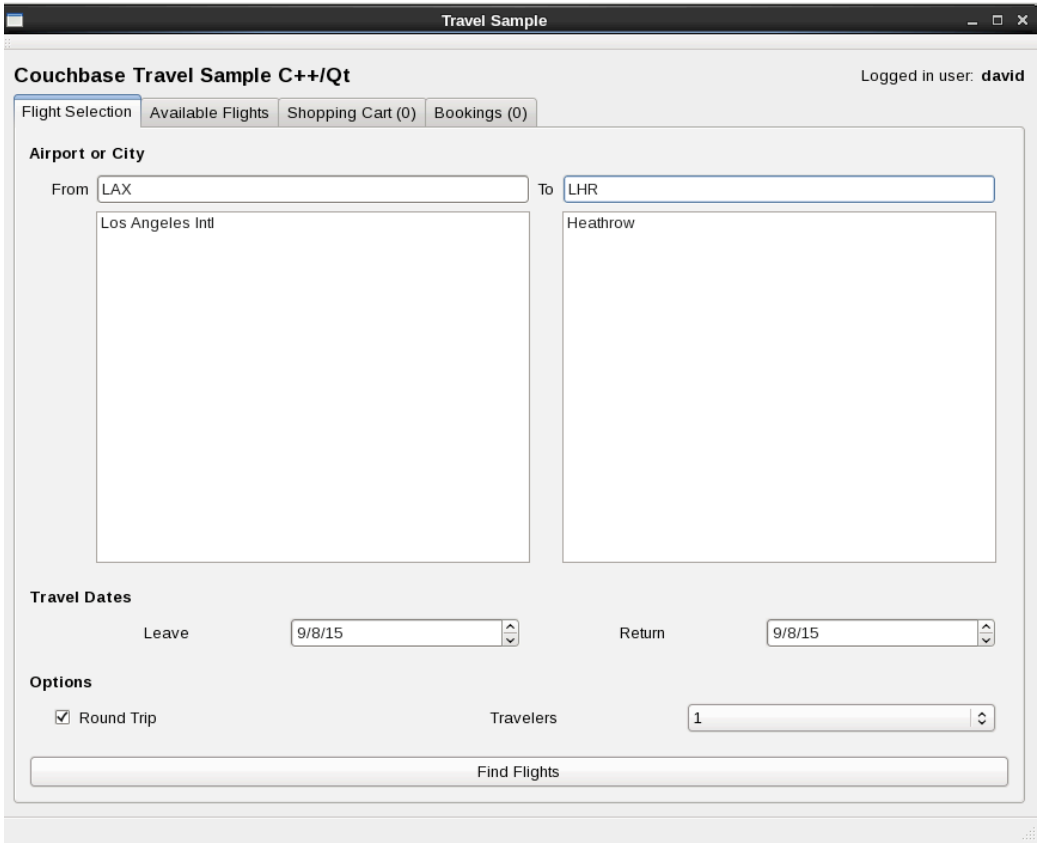
- Search for a flight from 'LAX' to 'LHR'



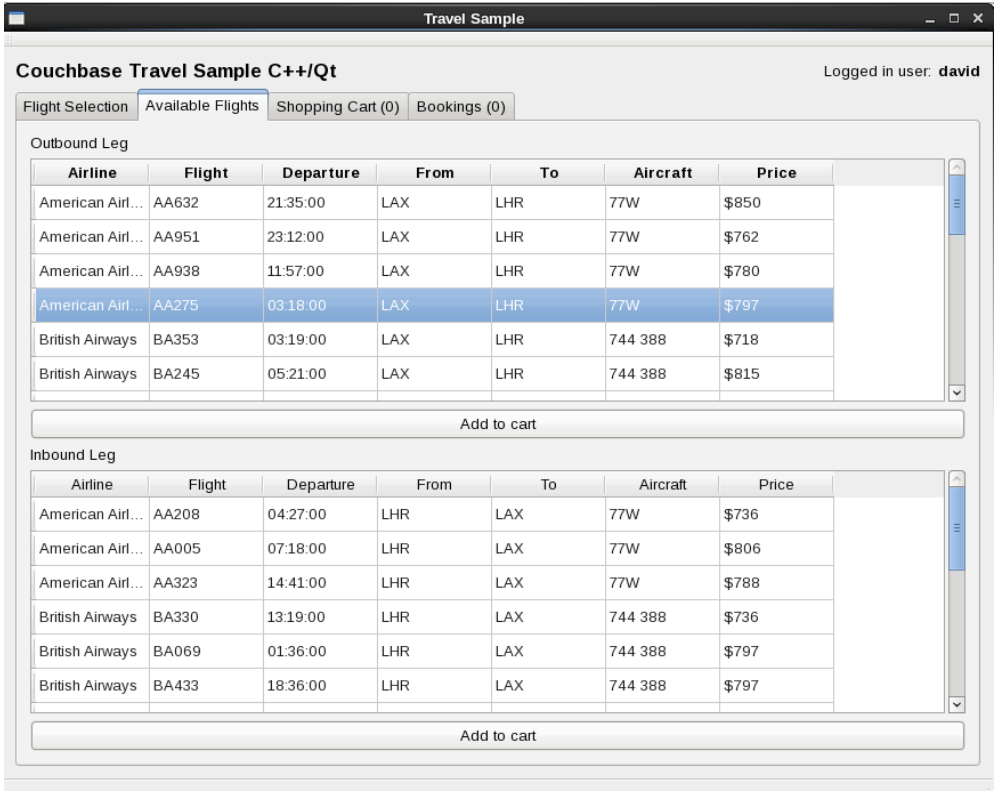
A Sample Application



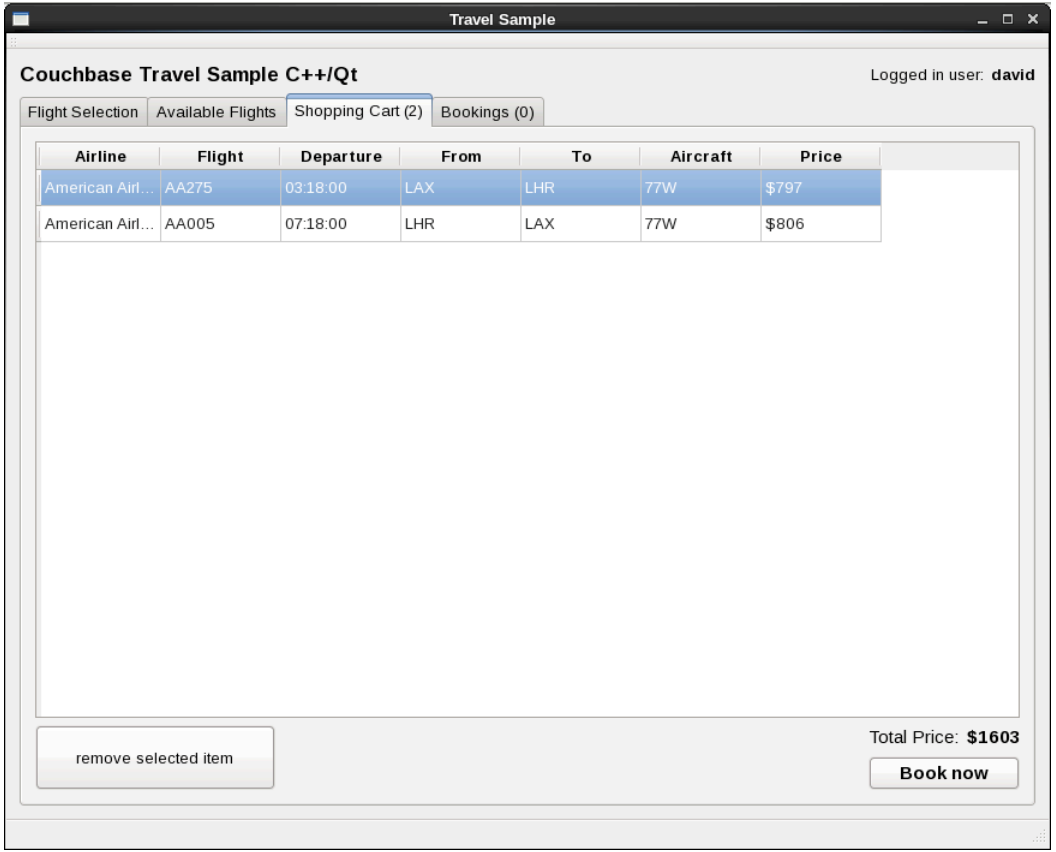
A Sample Application



A Sample Application



A Sample Application



Q&A

http://docs.couchbase.com/developer/c-2.4/c-intro.html