



# Architecture and Administration Basics

Workshop **Day 2 – Labs**  
**Java**

Ludovic Dufrenoy, Manager Professional Services EMEA  
<https://github.com/dufrenoyl/cb-workshop-2d/>

1

# Installation & Configuration SDK

# SDK Java - libcouchbase Installation



Perform the following steps in order to add a dependency to libcouchbase

- Clone the source repository in your home folder.

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

- Make sure Couchbase Server is running on your Machine.
- Go to the java/1 directory and edit the pom.xml
- Change Couchbase version to the latest. (2.7.1 on Nov 6<sup>th</sup> 2018)

```
<dependency>
    <groupId>com.couchbase.client</groupId>
    <artifactId>java-client</artifactId>
    <version>2.7.1</version>
</dependency>
```

<https://developer.couchbase.com/documentation/server/current/sdk/java/start-using-sdk.html>

# Load the Project into NetBeans

The image  
part with  
width=10  
height=10  
was not  
found in  
any mo-

## Perform the following steps:

- Start a Server X on your **Oracle Virtual Box Terminal** or any **Client X**.

[startXFCE4](#)

[\(Linux\) VNC Viewer](#)

[\(MacOS\) vnc://<IP>:5901](#)

- Start NetBeans IDE from the Desktop.
- Open the Existing projects “1” (from scratch) ; “2” (basics) ; “3” (solutions) in NetBeans IDE
- Create a bucket in Couchbase UI named “workshop”.
  - 100MB
  - Flush enabled
  - No Replica.
- Build & Run your application => should be successful.

# Documentation & Examples

## Open the documentation for libcouchbase!

- <https://docs.couchbase.com/java-sdk/2.7/start-using-sdk.html>
- <http://docs.couchbase.com/sdk-api/couchbase-java-client-2.7.1/>
- Open the “3” project => This is the solution.
- Open the “1” or “2” project => This is where you start.
- Check the Sample App backend  
<https://docs.couchbase.com/java-sdk/2.7/sample-app-backend.html>

# 2

## Managing Connections

# Create an applicative User via RBAC

The image  
part with  
p 10 r1z2  
was not  
found in  
any mo.

## Perform the following operations:

- Go the Security Tab in Couchbase
- Create a new User.
- Username = “workshop”
- Password = “couchbase”
- Roles
  - Data Reader on ‘workshop’
  - Data Writer on ‘workshop’

### Edit User workshop

X

**Authentication Domain**

Couchbase    External

**Username**

workshop

**Full Name (optional)**

**Roles**

Admin

Cluster Admin

Read Only Admin

▶ Bucket Roles

▶ Data Roles

▶ FTS Roles

▶ Query Roles

[Cancel](#) [Save](#)

# Cluster and Bucket

The image  
part with  
width=10  
height=10  
was not  
found in  
any res.

- Cluster
  - **Bootstrap the Connection**
  - **Just needs one or multiple IP-s / host names**
- Bucket
  - **Perform data operations**
  - **Thread-Safe**
  - **Reuse it (Singleton)**

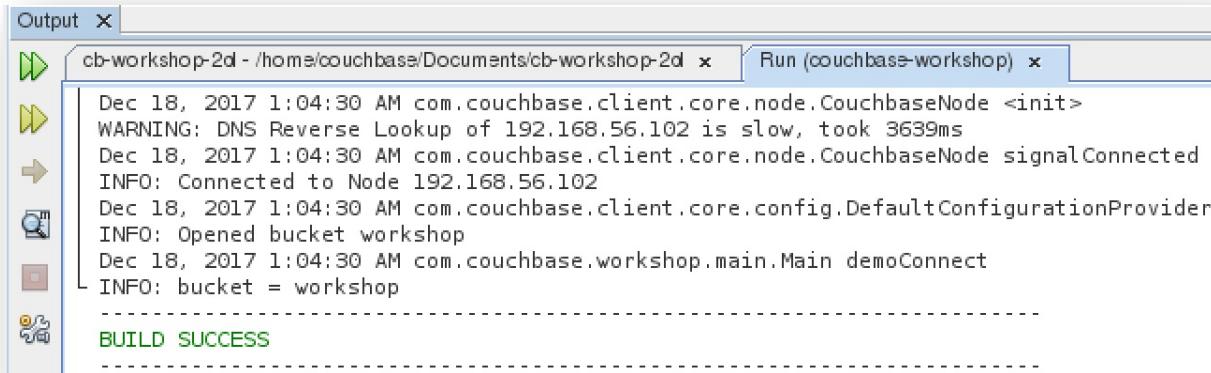
# Connecting to Couchbase with RBAC

The image  
part with  
width=10  
height=10  
was not  
found in  
any res.

Update cb.properties file with:

- Couchbase cluster IP
- Bucket name
- Username and password

Bootstrap your connection and run your application:



```
Output ×
cb-workshop-2d - /home/couchbase/Documents/cb-workshop-2d × Run (couchbase-workshop) ×
Dec 18, 2017 1:04:30 AM com.couchbase.client.core.node.CouchbaseNode <init>
WARNING: DNS Reverse Lookup of 192.168.56.102 is slow, took 3639ms
Dec 18, 2017 1:04:30 AM com.couchbase.client.core.node.CouchbaseNode signalConnected
INFO: Connected to Node 192.168.56.102
Dec 18, 2017 1:04:30 AM com.couchbase.client.core.config.DefaultConfigurationProvider$1
INFO: Opened bucket workshop
Dec 18, 2017 1:04:30 AM com.couchbase.workshop.main.Main demoConnect
INFO: bucket = workshop
-----
BUILD SUCCESS
```

3

# Working with Documents

# Document

The image  
part with  
width=10  
height=10  
was not  
found in  
any res.

- Properties
  - **Content**
  - **Metadata (id, CAS value, expiry)**
- Types
  - **JsonDocument**
  - **BinaryDocument**
  - **SerializableDocument**
  - **RawJsonDocument (to use own JSON Serializer)**

# Operations

- Get
- GetFromReplica
- GetAndLock
- Touch
- Upsert
- Insert
- Replace
- Append
- Prepend
- Unlock
- ...

The image  
part with  
width=10  
height=10  
was not  
found in  
any res.

# Management

- ClusterManager
  - **info**
  - **hasBucket**
  - **getBucket**
  - **insert/remove/updateBucket**
- BucketManager
  - **info**
  - **flush**
  - **get/insert/upsert/remove/publishDesignDocument**

# The Patterns

The image  
part with  
width=10  
height=10  
was not  
found in  
any res.

- In the Observer pattern, Observers are used to observe a subject (Observable). The Observers are realizing (signaled by the Observable) if the state of the Observable is changing by being able to react on such an event. It's easy to see that you can use it also for asynchronous data access.
- The Iterator pattern allows to iterate through a collection of objects without the need to know the internal structure of the collection of objects. To combine the Observer pattern with the Iterator pattern means that you can receive a data stream (instead the whole collection of data) by then reacting on the arrival of new data items.
- The functional aspect is that you may pass functions (E.G. as anonymous classes or lambda expressions) as the arguments to an Observable.

# Observable

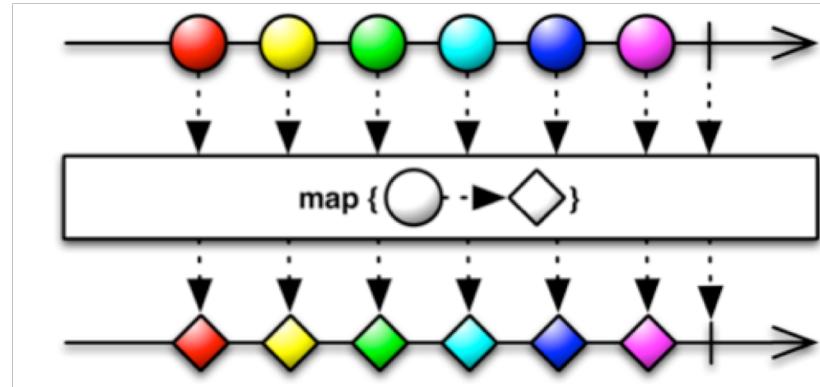
The image  
part with  
width=0  
and height=0  
was not  
loaded.

- Observable<T> and Observer<T>
- Observable = a Stream of Data
- Rx operators to manipulate the stream

	<i>Single</i>	<i>Multiple</i>
<i>Sync ("pull")</i>	<i>T</i>	<i>Iterable&lt;T&gt;</i>
<i>Async ("push")</i>	<i>Future&lt;T&gt;</i>	<i>Observable&lt;T&gt;</i> 

# Transformation

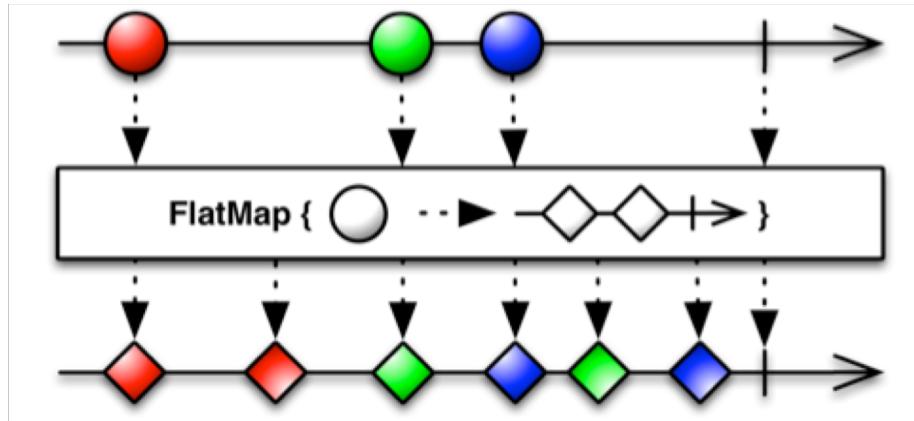
- Observable<T> source
- Observable<R> transformed = source.map(function)
- map(function) transforms each T into R



# Transformation

The image part with width=10 and height=10 was not found in any file.

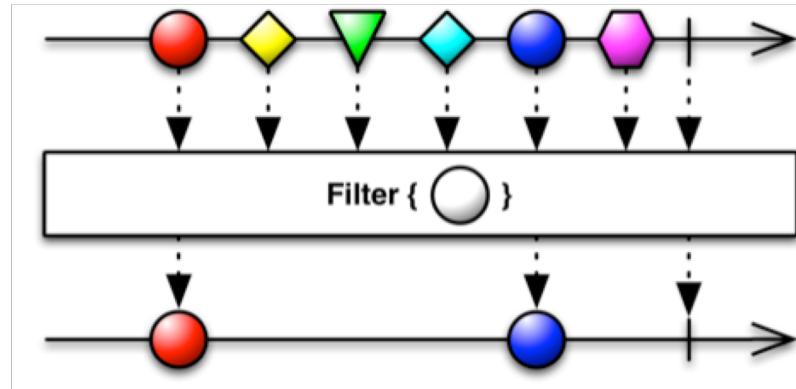
- flatMap is like map but emits again an Observable which is then flattened into one Observable (output stream)
- When the transformation must be also async.



# Filter

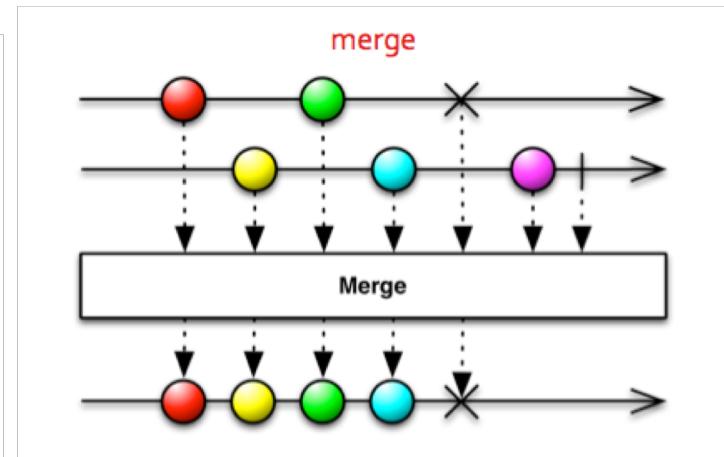
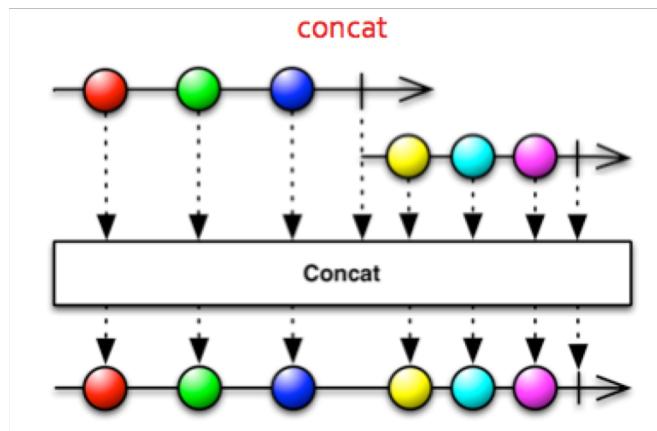
The image  
part with  
width=0  
and height=0  
was not  
loaded.

- Observable<T> filter(predicate)
- E.G.
  - **first()**
  - **take(n)**
  - **skip(n)**
  - ...



# Combine

- E.G.
  - **merge(observableOfT)**
  - **concat(observableOfT)**
  - ...



# Subscribe

The image  
part with  
width=10  
height=22  
was not  
found in  
any res.

- By passing an Observer or the following Actions
  - **onError**
  - **onNext**
  - **onComplete**

# CRUD Operations

Create a company document which has the following properties:

- Id
- Type ("company")
- Name
- Address



Create 7 user documents those have the following properties:

- |  |  |
|--|--|
| <ul style="list-style-type: none"><li>■ Uid</li><li>■ Type ("user")</li><li>■ First name</li><li>■ Last name</li></ul> | <ul style="list-style-type: none"><li>■ Email address</li><li>■ Birthday</li></ul> |
|--|--|

# CRUD Operations

The image  
part with  
width=10  
height=10  
was not  
found in  
any res.

Assign every user to a company!

For every user create an email lookup document! (Optional)



# 4

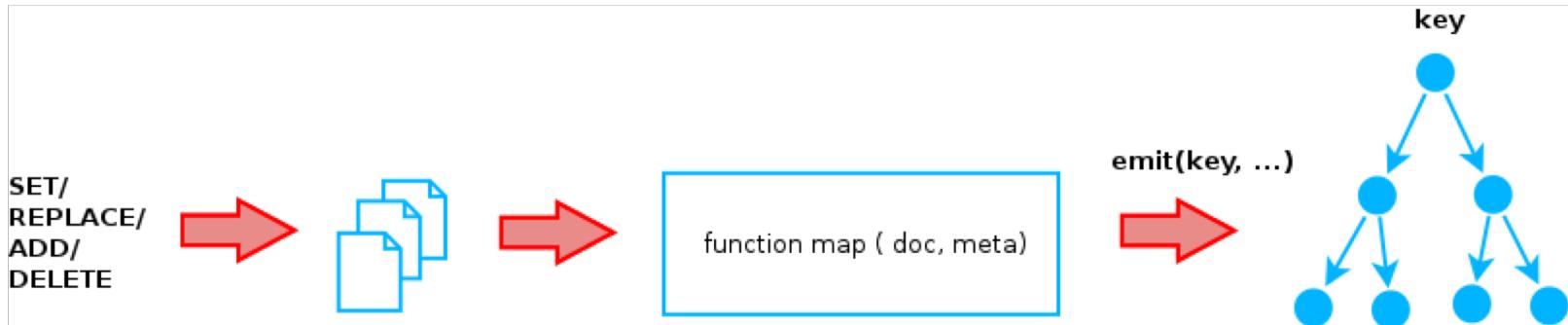
# Indexes using VIEWS

# Views

The image part with id=10 not found in any file.

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

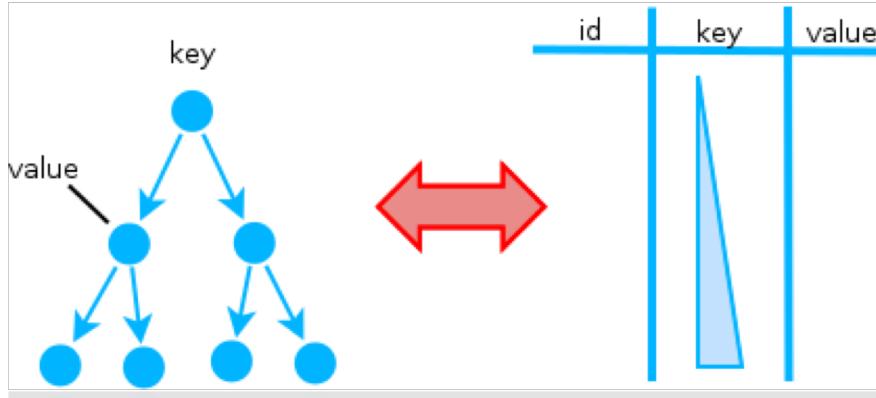
Map	Reduce
Process, filter, map and emit a row	Aggregate mapped data Built in: <code>_count, _sum, _stats</code>



# Views

The image  
part with  
width=0  
p10\_rect2  
was not  
found in  
any mo

- 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



# Querying via Views

Create a new Design Document with the name 'persons'!

Create a View with the name 'by\_birthday'

Implement a View which:

- Filters users
- Emits the birthday as a date array and so as a compound key
- Emits the first name and the last name as the value



Query the View by using the web browser:

- Use the built-in reduce function \_count and group the result by the year of birth

# Querying via Views

**Query the View by using the web browser:**

- Use the built-in reduce function \_count and group the result by the year of birth

**Query the View by using the Java client!**



# 4

## Indexes using GSI (Plasma)

# Querying via N1QL

Make sure that Primary Index is created!

Also create a Secondary Index on the type attribute!

Create a Secondary Index on the last name!

- Create the indexes programmatically by using the Index API



# Querying via N1QL

**Query for users by their last name!**

- Use the builder API for building the N1QL statement

**Query users of a specific company and with a specific last name  
by using a JOIN!**

- Use a query string for this purpose

5

# Error Management & Logging

# Error Handling

- In the subscribe
  - **onError(function)**
- Return default value
  - **onErrorReturn(T defaultValue)**
- Defer to a backup stream on error
  - **onErrorResumeNext(Observable<T> backup)**
- Retry on error
  - **retry(predicate)**

6

# FTS

# Full Text Search

# Querying via FTS

Make sure the default FTS is created!

Give it the following name: "idx\_default"

- Search for an email in the "workshop" bucket.



# Thank you



Couchbase