

# RWDevCon / Lab

## Introduction

- What is Couchbase Mobile
- What are we going to do
  - Data storage
  - Synchronization
  - Security
- What are we going to build
  - Navigation flow
  - Wireframes
  - Model
  - Topology

## Lab

## Demo

1. Log In
2. CRUD
  - a. Create task list “Shopping”
  - b. Create tasks
    - i. Add task “Apples” + pic
    - ii. Add task “Bananas” + pic
    - iii. Add task “Eggz” + pic
  - c. Update task
    - i. Update “Eggz” to “Eggs”
    - ii. While viewing “Apples” pic on one device, update pic from another device
  - d. Delete task
    - i. Create new “test” task and then delete it
  - e. Query
    - i. Demo “Tasks Lists” starts-with query
    - ii. Demo “Tasks” contains query
3. Data Change Events
  - a. Show incomplete tasks count
    - i. While viewing “Task Lists” screen, check/uncheck items on “Tasks” screen from another device
  - b. Talk about Live Query
  - c. Talk about “Task List” tab-bar show/hide and demo during security coding

4. Security
  - a. Show “Task List > Users” screen and demo in security coding
  - b. Show offline login
  - c. Talk about key rotation and demo in security coding

## Build app using embedded database

5. Setup
  - a. Clone [repo](#)
  - b. Open project in Xcode
  - c. Install Couchbase Lite
    - i. [Download](#) Couchbase Lite for iOS
    - ii. Copy CouchbaseLite.framework to Frameworks directory
    - iii. Copy Extras/libsqlcipher.a to Frameworks directory
6. Create database (**AppDelegate.swift 68**)
7. Create UI components (**Main.storyboard**)
8. CRUD
  - a. Task Lists
    - i. Load (**TaskListsViewController.swift 189-191**)
    - ii. Reload on change (**TaskListsViewController.swift 190, 77, 214-215**)
    - iii. Query (**TaskListsViewController.swift 143-152**)
    - iv. Create (**TaskListsViewController.swift 231-249**)
    - v. Update (**TaskListsViewController.swift 253-261**)
    - vi. Delete (**TaskListsViewController.swift 265-271**)
    - vii. Incomplete Tasks count (**TaskListsViewController.swift 193-210, 96**)
  - b. Tasks
    - i. Load
    - ii. Reload on change
    - iii. Query
    - iv. Create
    - v. Update
    - vi. Delete
    - vii. Show/Hide Users tab (**TasksViewController.swift 329-354**)
  - c. Task List Users
    - i. Load
    - ii. Reload on change
    - iii. Query
    - iv. Create
    - v. Delete
  - d. Photo Viewer
    - i. Load (**TaskImageViewController.swift 89-94**)
    - ii. Reload on change (**TaskImageViewController.swift 29-44**)
    - iii. Create (**TaskImageViewController.swift 89-94**)

- iv. Update (**TaskImageViewController.swift 89-94**)
- v. Delete (**TaskImageViewController.swift 115-124**)

## Add replication

1. Setup
  - a. [Download](#) Sync Gateway for Mac OSX
  - b. Copy the sync\_gateway executable to your mobile-training-todo directory (the [configuration file](#) is also in this directory)
  - c. Start Sync Gateway
    - i. `cd ../mobile-training-todo`
    - ii. `./sync_gateway sync-gateway-config.json`
2. Couchbase Lite
  - a. Configure Url (**AppDelegate.swift 14**)
  - b. Push replicator (**AppDelegate.swift 205-210, 221**)
  - c. Pull replicator (**AppDelegate.swift 212-219, 222**)
  - d. Conflict Resolution (**AppDelegate.swift 264-419**)
    - i. Task Lists (accept default)
    - ii. Tasks (n-way merge)
    - iii. Task List Users (accept default)

## Add security

1. Couchbase Lite
  - a. Create UI components (**Main.storyboard**)
    - i. Login screen
2. Sync Gateway
  - a. Update [sync function](#)
  - b. Add Users and Moderator
    - i. Add 1 user using REST method (fujio)
      1. `curl -XPOST http://127.0.0.1:4985/todo/_user/ -d '{"name":"fujio","password":"fujio"}' -H 'Content-Type: application/json'`
    - ii. Add 1 moderator using REST method (wayne)
      1. `curl -XPOST -v http://admin:admin@127.0.0.1:4984/todo/ -d '{"_id": "moderator.wayne","type":"moderator","username": "wayne"}' -H 'Content-Type: application/json'`
3. Advanced security
  - a. Database encryption (**AppDelegate.swift 74-78**)
  - b. Offline login (**AppDelegate.swift 76**)
  - c. Key rotation (**AppDelegate.swift 82**)
    - i. Show by changing a user's password (wayne)

1. `curl -XPUT http://127.0.0.1:4985/todo/_user/wayne -d '{"name":"wayne","password":"wayne1"}' -H 'Content-Type: application/json'`