

NoSQL mit Couchbase - Übung

Autor: Thomas Rehm

Vorbereitungen

Um die nachfolgenden Übungen durchführen zu können, müssen die folgenden Programme heruntergeladen und installiert werden.

Download-Links oder Dateien vom Datenträger:

- Couchbase Server 3.0.1 64bit Community Edition (<http://www.couchbase.com/nosql-databases/downloads>)
 - Couchbase Server WIN (http://packages.couchbase.com/releases/3.0.1/couchbase-server-community_3.0.1-windows_amd64.exe)
 - Couchbase Server MAC (http://packages.couchbase.com/releases/3.0.1/couchbase-server-community_3.0.1-macos_x86_64.zip)
 - Couchbase Server Ubuntu12.04 (http://packages.couchbase.com/releases/3.0.1/couchbase-server-community_3.0.1-ubuntu12.04_amd64.deb)
- Couchbase SyncGateway 1.0.3 64bit Community Edition
 - Couchbase SyncGateway WIN (http://www.couchbase.com/dl/releases/couchbase-sync-gateway/1.0.3/couchbase-sync-gateway-community_1.0.3_x86_64.exe/download)
 - Couchbase SyncGateway MAC (http://www.couchbase.com/dl/releases/couchbase-sync-gateway/1.0.3/couchbase-sync-gateway-community_1.0.3_x86_64.tar.gz/download)
 - Couchbase SyncGateway Ubuntu (http://www.couchbase.com/dl/releases/couchbase-sync-gateway/1.0.3/couchbase-sync-gateway-community_1.0.3_x86_64.deb/download)
- AndroidStudio (<http://developer.android.com/sdk/index.html>)
- Clone CouchbaseSampleApp Repository (<https://github.com/thomasrehm/CouchbaseSample>)
- git (<http://git-scm.com/downloads>) (wenn nicht bereits installiert)
- Java Development Kit (<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>) (wenn nicht bereits installiert)

Teil 1

Installation Couchbase Server

Bitte installiert als erstes Couchbase Server auf eurem System und startet den Download und die Installation der restlichen Dateien. Während dessen werden wir einen ersten kleinen Einstieg in Couchbase Server durchführen.

- Installiert Couchbase Server auf eurem System.
- Startet die AdminUI unter der Adresse localhost:8091/index.html (<http://localhost:8091/index.html>) , falls diese sich nicht automatisch öffnet
- Führe das Setup mit den Standardparametern durch
 - *Step 1 of 5* einfach defaults lassen
 - *Step 2 of 5* Sample Buckets werden zunächst nicht benötigt
 - *Step 3 of 5* legt das "default" Bucket mit Standardparametern an, außer den zugewiesenen RAM-Speicher. Den RAM auf 100 bis 200 MB einstellen und nicht, wie vorgeschlagen, den gesamten verfügbaren RAM reservieren. Würden wir das so anlegen, könnten wir später keinen neuen Bucket anlegen, ohne das default-Bucket zu ändern, da kein RAM mehr verfügbar wäre.
 - *Step 4 of 5* Selbst entscheiden, ob ihr Update-Benachrichtungen bekommen und euch in der Community registrieren wollt.
 - *Step 5 of 5* Administrator anlegen, Passwort merken

Glückwunsch. Du hast Couchbase Server installiert!

Erste Schritte

Um einen Einstieg zu bekommen nutzen wir das UserInterface von Couchbase Server das unter der Adresse

localhost:8091/index.html (<http://localhost:8091/index.html>) erreichbar ist.

Falls git (<http://git-scm.com/downloads>) noch nicht auf deinem System installiert ist, installiere es jetzt, damit auf einem Windows-System eine Terminal-ähnliches Commandline-Interface vorhanden ist.

Um einen Einstieg zu bekommen, nutzen wir das UserInterface von Couchbase Server, das unter localhost:8091/index.html (localhost:8091/index.html) erreichbar ist. Zum Testen:

- Unter **Bucket** mittels **Create new Data Bucket** neuen Test Bucket anlegen mit Namen **people**
- Danach **Documents** beim angelegten Bucket auswählen, als nächstes mittels **Create Document** neues Document anlegen. Das Dokument sollte die Id **person_o** heißen und die Felder **first_name**, **last_name** sowie ein Embedded Doc mit dem Key **data** und den Feldern **language**, **country**, **age** enthalten

Wir könnten natürlich von Hand einige Daten einfügen, oder wir nutzen einfach den **cbdocloader** der sich in den mitgelieferten CLI Tools befindet. Starte ein neues Terminal oder Git Bash Window (am einfachsten direkt beim Ordner *Rechtsklick // Dienste // neues Terminal beim Ordner* oder *Rechtsklick // Git Bash here*) und importiere die Daten wie folgt:

Mac:

```
$ /Applications/Couchbase\ Server.app/Contents/Resources/couchbase-core/bin/cbdocloader -n 127.0.0.1:8091 -u Username -p Password -b Bucketname folder name
```

Windows:

```
$ /C/Program\ Files/Couchbase/Server/bin/cbdocloader.exe -n localhost:8091 -u Username -p Password -b Bucketname foldername
```

Dokumentation cbdocloader (http://docs.couchbase.com/admin/admin/CLI/cbdocloader_tool.html)

Neuen View anlegen:

- auf **Views** klicken
- **Create Development View** klicken und neues Design Dokument anlegen & ersten View benennen, zum Beispiel:
 - Design Document Name: `_design/dev_people`
 - View Name: `all_people`
- neue Map anlegen (siehe Folien), die alle Vornamen der Personen indexiert
- View aufrufen (siehe Folien)

Teil 2

Im zweiten Teil widmen wir uns Couchbase Lite und dem Couchbase SyncGateway. Dazu werden wir die vorbereitete Android App lokal zum laufen bringen und eine Verbindung zwischen App, SyncGateway und Server herstellen.

(In Anlehnung an Building your first Couchbase Lite Android app (<http://developer.couchbase.com/mobile/develop/training/build-first-android-app/index.html>))

Überblick:

- Installation & Einrichtung AndroidStudio (inkl. Android SDK und Emulator) (AndroidStudio prüft ob ein JavaDevelopmentKit verfügbar ist, falls nicht kommt ein Hinweis. Downloaden oder vom Stick)
- Vorbereitete App aus Bitbucket Repository laden und Variablen eintragen das Sync läuft
- Installation & Einrichtung Couchbase SyncGateway
- In Emulator oder Android-Smartphone direkt testen und Logs in der Konsole untersuchen

Installation

- Als erstes das heruntergeladene AndroidStudio installieren.
- AndroidStudio starten und Android SDK einrichten:
 - Klicke im Welcome-Screen von Android Studio auf **Configure >> SDK Manager**
 - Setzte haken bei den folgenden Packages
 - Tools/Android SDK Tools
 - Tools/Android SDK Platform-tools
 - Tools/Android SDK Build-tools

- Android API 21 (Android 5.0.1)
- Extras/Google Repository
- Extras/Google USB Driver
- Extras/Android Support Repository
- Extras/Intel x86 Emulator Accelerator (HAXM installer)
- Rest unchecken damit der Download nicht ewig dauert
- Klicke **Install Packages** >> Accept License >> Install
- Android SDK Manager geöffnet lassen bis alles installiert ist
- Wähle im Welcome-Screen **Open an existing Android Studio project** und wähle den Pfad zum bereits von Github (<https://github.com/thomasrehm/CouchbaseSample>) geklonten Projekt
- Öffne das Projekt, wenn es nicht ohnehin schon geöffnet wurde

Entscheide dich nun für eine der folgenden Optionen:

1. Nutze dein eigenes Android Smartphone via USB Debugging. Dazu folge der Anleitung auf [techtopia.com](http://www.techtopia.com) (http://www.techtopia.com/index.php/Testing_Android_Studio_Apps_on_a_Physical_Android_Device#An_Overview_of_the_Android_Debug_Bridge_28ADB29) . Falls du ein kein Android 5.0.1 besitzt, lade am besten zusätzlich die Android Version im **SDK Manager** herunter, die dein Smartphone unterstützt. Eventuell muss dann in der Datei **build.gradle (Module: app)** die **minSdkVersion** angepasst werden.
2. Nutze einen Android Emulator zum testen
 1. Emulator mit einem **x86 System Image** und Intels Hardware Accelerated Execution Manager (Weiter HAXM Infos auf envyandroid.com (<http://envyandroid.com/intel-haxm-hardware-acceleration-on-android-emulator/>))
 - Versuche Intels Hardware Accelerator zu installieren, zu finden unter (Win: User/AppData/Local/Android/sdk/extras/intel/Hardware_Accelerated_Execution_Manager/intelhaxm-android oder Mac: Users/User/Library/Android/sdk/extras/intel/Hardware_Accelerated_Execution_Manager/IntelHAXM_1.1.1_for...dmg)
 - Schlägt die Installation fehl, weiter mit **2.**, ansonsten weiter
 - Wähle **Tools >> Android >> AVD Manager** und richte einen Android Emulator zum testen der App ein
 - Wähle **Create virtual Device**
 - Wähle als Phone ein Nexus 4, 5 oder 6 und klicke **Next**
 - Wähle als nächstes das eben installierte System Image **Lollipop / API Level 21 / ABI x86 / Target 5.0.1** und klicke **Next**
 - Stelle sicher das **Use HOST GPU** gecheckt ist und klicke **Finish**
 2. Wenn dein System IntelHAXM nicht unterstützt, richte einen Emulator mit einem **arm System Image** ein:
 - Wähle **Tools >> Android >> AVD Manager** und richte einen Android Emulator zum testen der App ein
 - Wähle **Create virtual Device**
 - Wähle als Phone z.B. ein Nexus 4 und klicke **Next**
 - Wähle als nächstes das eben installierte System Image **Lollipop / API Level 21 / ABI armeabi-v7 / Target 5.0.1** und klicke **Next**
 - Stelle sicher das **Use HOST GPU** gecheckt ist und klicke **Finish**

Sync Gateway einrichten

- Neuen Bucket anlegen, Name z.B. **sync_gateway**
- Neue Config-Datei im JSON-Format für das SyncGateway anlegen (siehe Folien) oder Documentation (<http://developer.couchbase.com/mobile/develop/guides/sync-gateway/administering-sync-gateway/command-line-tool/index.html>) mit deinen Zugangsdaten zur lokalen Couchbase Server Instanz
- Am Mac einfach Binary im Ordner couchbase-sync-gateway/bin per Terminal mit Config-File als ausführen `$./sync_gateway [ConfigurationFile...]`
- Unter Windows **Couchbase SyncGateway** installieren und zum Ordner navigieren, Neues Git Bash beim Ordner und die exe mit der config.json ausführen

App vervollständigen

- Vervollständige die Variablen in der App (siehe TODOs)
- Wähle nun den **Run App**-Button (Play-Symbol) und starte den Build Vorgang (dauert etwas)
 - Ein Popup fragt nun welches Gerät genutzt werden soll, wähle entweder dein angeschlossenes Smartphone oder den

Aufgaben

- Lege einige Docs an, lass sie anzeigen, starte Sync und untersuche Sync-Log im Terminal
- Neue Sync-Funktion in der Config des SyncGateways (Beispielsweise so, dass bestimmte Documents nicht angenommen werden) und starte SyncGateway und Sync in der App neu (Dokumentation SyncFunction (<http://developer.couchbase.com/mobile/develop/guides/sync-gateway/sync-function-api-guide/index.html>))
- Erweitere die Funktion `SetNewDocumentContent` so, das jedem Document ein neues Key-Value-Paar angehängt wird, das den Namen der lokalen Datenbank enthält
- Erstelle am Server für das synchronisierte Bucket einen neuen View, der nur die Documents im Index aufnimmt, die zu einer bestimmten lokalen Datenbank gehören
- Frage den gerade erstellten View ab und sortiere die Ergebnisse in umgekehrter Reihenfolge (siehe Querying (<http://docs.couchbase.com/admin/admin/Views/views-querying.html>))