**Configuring and testing Cards and Accounts Service**

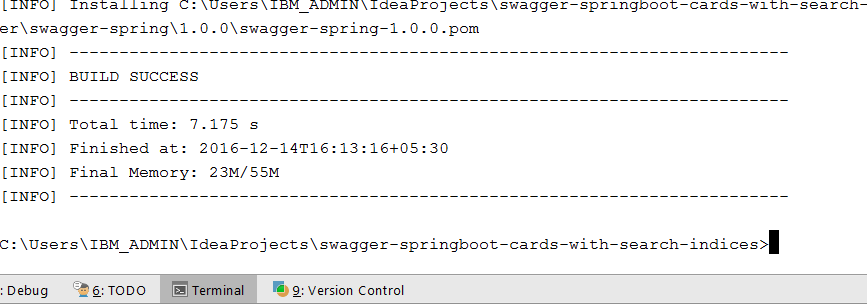
1. Clone the code in to intelliJ from <https://github.com/vajadhav/swagger-springboot-cards-with-search-indices.git>
2. Copy the **spring-cloud.properties** file to **c:/dev folder**
3. Make sure that the **cloudantdatabase** container is up and running in docker as below screen shot.



1. Install all the dependencies by running this command on terminal

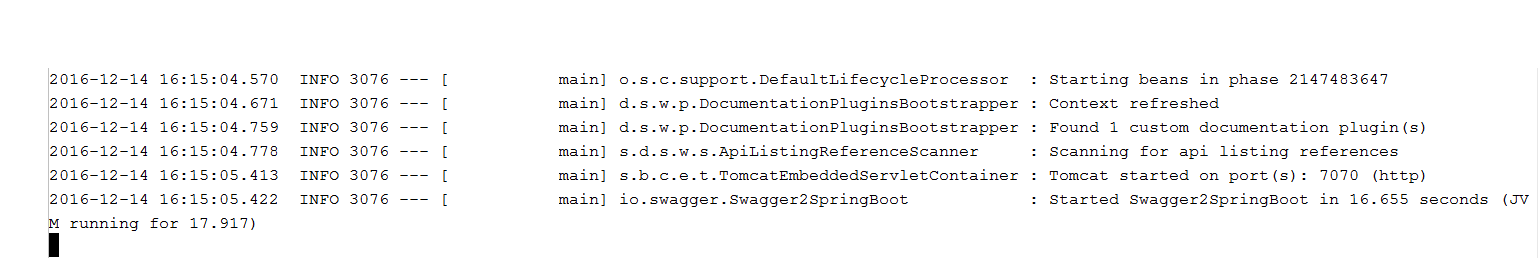
**mvn clean install -U**

This command will result the below screen-



1. After successful build run the below command to create the jar file

**java -jar target/swagger-spring-1.0.0.jar**

This will result the screen saying JVM is running as below.

1. Make sure that the **card\_db** OR **account\_db** database is created automatically in Cloudant database after running the above command, as shown in belowscreen shot.



1. Create the search index in card\_db and account\_db with name search\_card\_details and search\_account\_details respectively.
2. **Account’s search index details-**

**Save to Design Document**: AccountDetails

**Index name**: search\_account\_details

**Search index function:**

function(doc)

{

index("default", doc.\_id);

if(doc.userName ){

index("userName", doc.userName , {"facet": true, "store": true});

}

if(doc.customerId ){

index("customerId", doc.customerId , {"facet": true, "store": true});

}

if(doc.accountType ){

index("accountType", doc.accountType , {"facet": true, "store": true});

}

if (doc.accountStatus){

index("accountStatus", doc.accountStatus , {"store": true});

}

if(doc.address){

index("address", doc.address, {"facet": true, "store": true});

}

if(doc.accountNumber){

index("accountNumber", doc.accountNumber, {"facet": true, "store": true});

}

if(doc.mobileNumber){

index("mobileNumber", doc.mobileNumber, {"facet": true, "store": true});

}

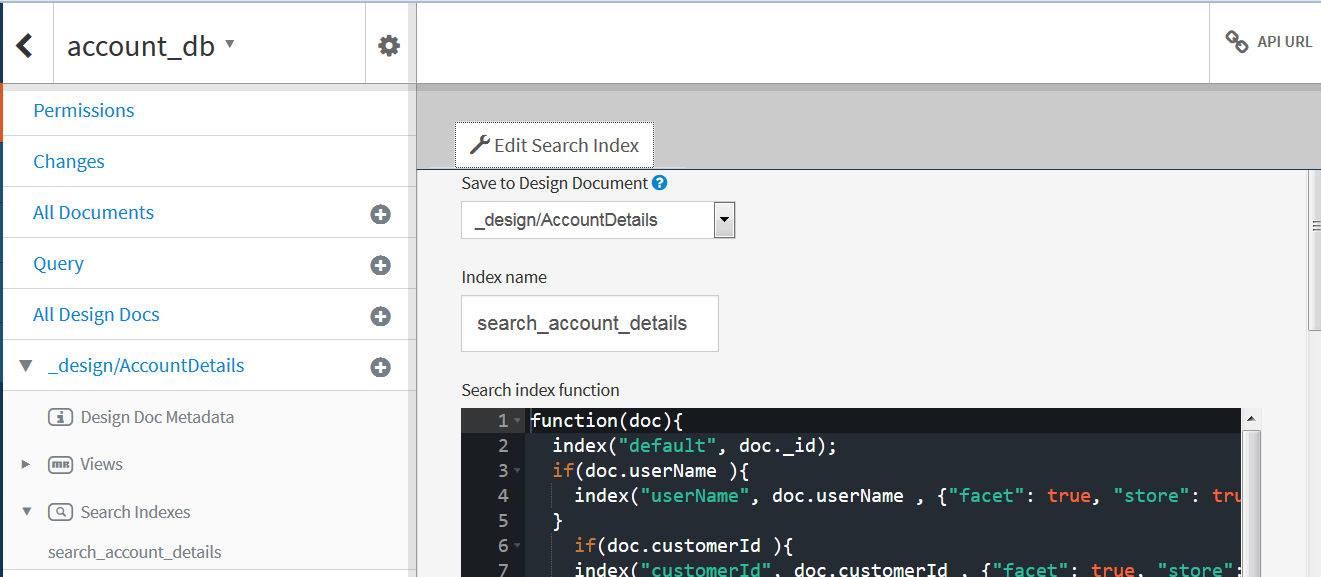
if (doc['class']){

index("class", doc['class'], {"facet": true, "store": true});

}

}

Account’s Search index should be look like below Screen shot-



1. **Card’s search index details-**

**Save to Design Document**: CardDetails

**Index name**: search\_card\_details

**Search index function:**

function(doc)

{

index("default", doc.\_id);

if(doc.cardNumber)

{

index("cardNumber", doc.cardNumber, {"facet": true, "store": true});

}

if(doc.customerName){

index("customerName", doc.customerName, {"facet": true, "store": true});

}

if(doc.cardType){

index("cardType", doc.cardType, {"facet": true, "store": true});

}

if (doc.custId){

index("custId", doc.custId, {"store": true});

}

if(doc.cardStatus){

index("cardStatus", doc.cardStatus, {"facet": true, "store": true});

}

if(doc.startDate){

index("startDate", doc.startDate, {"facet": true, "store": true});

}

if(doc.expiryDate){

index("expiryDate", doc.expiryDate, {"facet": true, "store": true});

}

if (doc.cardApplyMode){

index("cardApplyMode", doc.cardApplyMode, {"store": true});

}

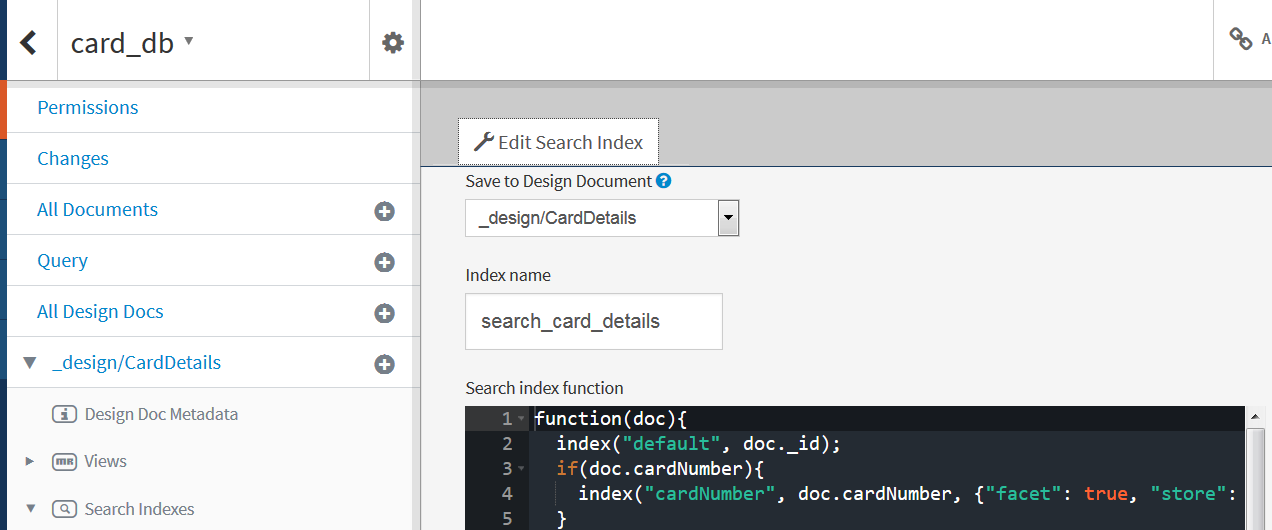
if (doc['class']){

index("class", doc['class'], {"facet": true, "store": true});

}

}

Account’s Search index should be look like below Screen shot-



1. Test the APIs on swagger-ui using below URL

**URL-** <http://localhost:7070/card-accounts>

1. Test APIs in REST Client using below URLs.

**Account APIs:**

1. Get all accounts

**Method-** GET

**URL-** <http://localhost:7070/card-accounts/accounts>

1. Create account

**Method-** POST

**URL-** <http://localhost:7070/card-accounts/accounts>

**JSON Object-** {

"userName": "Garima Gupta",

"customerId": 890,

"accountType": "",

"accountStatus": "Active",

"accountBalance": 123.34,

"address": "Gurgaon",

"accountNumber": 1234,

"mobileNumber": 9899876285

}

1. GET account by accountNumber

**Method-** GET

**URL-** <http://localhost:7070/card-accounts/accounts/1234>

1. Update account by accountNumber

**Method-** PUT

**URL-** <http://localhost:7070/card-accounts/update-account/1234>

**JSON Object-** {

"userName": "Garima",

"customerId": 890,

"accountType": "Debit",

"accountStatus": "Active",

"accountBalance": 1236.34,

"address": "Gurgaon",

"accountNumber": 1234,

"mobileNumber": 9899876285

}

1. Delete account by accountNumber

**Method-** DELETE

**URL-** <http://localhost:7070/card-accounts/manage-account/1234>

**Card APIs:**

1. Get all cards

**Method-** GET

**URL-** <http://localhost:7070/card-accounts/card>

1. Create card

**Method-** POST

**URL-** <http://localhost:7070/card-accounts/card>

**JSON Object-** {

"cardNumber": 1112,

"cardType": "Credit",

"custId": 102,

"cardStatus": "Active",

"startDate": "12/12/2016",

"expiryDate": "12/12/2021",

"cardApplyMode": "mobile"

}

1. Get card by cardNumber

**Method-** GET

**URL-** <http://localhost:7070/card-accounts/fetch-card-details/1112>

1. Update card by cardNumber

**Method-** GET

**URL-** <http://localhost:7070/card-accounts/update-card/1111>

**JSON Object-** {

"cardNumber": 1112,

"cardType": "Debit",

"custId": 102,

"cardStatus": "Active",

"startDate": "12/12/2017",

"expiryDate": "12/12/2021",

"cardApplyMode": "web"

}

1. Delete card by cardNumber

**Method-** DELETE

**URL-** <http://localhost:7070/card-accounts/manage-card/1111>