CREATE TABLE student ( sid TEXT PRIMARY KEY, sfirstName TEXT, slastName TEXT , scourse TEXT );

INSERT INTO student (sid, sfirstName, slastName, scourse) VALUES (101 , ’ ankitss ’,’gupta’,’java’);

insert into student(sid,sfirstname,slastname,scourse)values('aa','a','b','c');

Casandra Table Creation and updation in Table

How to connect our Springboot Application from Cassandra;

Step-1--

docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

mariadb latest 1f9cfa8dc305 4 days ago 356MB

mariadb/server 10.3 890adfc23a42 5 days ago 345MB

cassandra latest 68144c842c79 5 weeks ago 324MB

hello-world latest fce289e99eb9 13 months ago 1.84kB

clue/json-server latest 2f3fead127fc 21 months ago 687MB

docker pull cassandra

Using default tag: latest

latest: Pulling from library/cassandra

619014d83c02: Pull complete

24c7ac5c73a5: Pull complete

b6875e232a43: Pull complete

e89165ecf457: Pull complete

078ed6a15b62: Pull complete

731cf201b941: Pull complete

7b87af3649c3: Pull complete

db3c5e9be55f: Pull complete

61a5f6b75d19: Pull complete

be9db1dfbf45: Pull complete

Step2-

docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

564a3a6df52b mariadb "docker-entrypoint.s…" 7 hours ago Up 7 hours 0.0.0.0:3306->3306/tcp test\_mysql

So here Cassandra not Up so we have to create container

Step-3

docker run --name test\_mycassandra -e MYSQL\_ROOT\_PASSWORD=test -e MYSQL\_DATABASE=practicedb -e MYSQL\_USER=test -e MYSQL\_PASSWORD=mypss -p 9042:9042 -d Cassandra

aaef191e515224df2ef13181427aec558eff4f9e6963a614780671dbbfb8c8bb

here test\_mycassandra is may container name and mysql root password is test

and mysql\_database is practicedb mysql user=test and password is mypass

step-3

docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

aaef191e5152 cassandra "docker-entrypoint.s…" 9 seconds ago Up 8 seconds 7000-7001/tcp, 7199/tcp, 9160/tcp, 0.0.0.0:9042->9042/tcp test\_mycassandra

564a3a6df52b mariadb "docker-entrypoint.s…" 7 hours ago Up 7 hours 0.0.0.0:3306->3306/tcp test\_mysql

Here I got my container got created that name is test\_mycassandra.

STEP-4

AFTER that we have to start our docker container like below commands.

docker start aaef191e5152

aaef191e5152

now container started after that we have to execute the Cassandra cmd

step-5

docker exec -it test\_mycassandra bash;

root@aaef191e5152:/# cqlsh

after that we are able to see the

Connected to **Test Cluster** at 127.0.0.1:9042.

[cqlsh 5.0.1 | Cassandra 3.11.5 | CQL spec 3.4.4 | Native protocol v4]

Use HELP for help.

Step-6

After that we have to create our keyspace----

cqlsh>

CREATE KEYSPACE mykeyspace WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication\_factor' : 1 };

cqlsh> use mykeyspace;

Now we are entering the mykeyspace.

Step-7- create table and insert data and check it like below.

cqlsh:mykeyspace> CREATE TABLE people ( id INT PRIMARY KEY, fullname TEXT, age INT );

cqlsh:mykeyspace> INSERT INTO people (id, fullname, age) VALUES (1, 'Ankit Gupta', 30);

cqlsh:mykeyspace> select \* from people;

**id** | **age** | **fullname**

----+-----+-------------

**1** | **30** | **Ankit Gupta**

Step-8. Adding the dependency in springboot application.

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-cassandra-reactive</artifactId>

</dependency>

Step-9- Adding the keyspace name and other properties to the application.properties.

server.port = 8082

spring.data.cassandra.keyspace-name=mykeyspace

spring.data.cassandra.contact-points=localhost

spring.data.cassandra.port=9042

Now Springboot Crud Api-

Student.java

@AllArgsConstructor

@Getter @Setter

@Table

**public** **class** Student {

@PrimaryKey

**private** @NonNull String sid;

**private** @NonNull String sfirstName;

**private** @NonNull String slastName;

**private** @NonNull String scourse;

StudentRepository.java

**public** **interface** StudentRepository **extends** CrudRepository<Student, String> {

}

StudentController.java

@RestController

**public** **class** StudentController {

@Autowired

**private** StudentRepository studentRepository;

@GetMapping(value = "/healthcheck", produces = "application/json; charset=utf-8")

**public** String getHealthCheck()

{

**return** "{ \"isWorking\" : true }";

}

@GetMapping("/student")

**public** List<Student> getAllStudents()

{

Iterable<Student> result = studentRepository.findAll();

List<Student> employeesList = **new** ArrayList<Student>();

result.forEach(employeesList::add);

**return** employeesList;

}

@GetMapping("/student/{id}")

**public** Optional<Student> getStudent(@PathVariable String id)

{

Optional<Student> stud = studentRepository.findById(id);

**return** stud;

}

@PutMapping("/student/{id}")

**public** Optional<Student> updateEmployee(@RequestBody Student newStudent, @PathVariable String id)

{

Optional<Student> optionalStudent = studentRepository.findById(id);

**if** (optionalStudent.isPresent()) {

Student stud = optionalStudent.get();

stud.setSfirstName(newStudent.getSfirstName());

stud.setSlastName(newStudent.getSlastName());

stud.setScourse(newStudent.getScourse());

studentRepository.save(stud);

}

**return** optionalStudent;

}

@DeleteMapping(value = "/student/{id}", produces = "application/json; charset=utf-8")

**public** String deleteEmployee(@PathVariable String id) {

Boolean result = studentRepository.existsById(id);

studentRepository.deleteById(id);

**return** "{ \"success\" : "+ (result ? "true" : "false") +" }";

}

@PostMapping("/studentAdd")

**public** Student addEmployee(@RequestBody Student newStudent)

{

String id = String.*valueOf*(**new** Random().nextInt());

Student st = **new** Student(id, newStudent.getSfirstName(), newStudent.getSlastName(),newStudent.getScourse());

studentRepository.save(st);

**return** st;

}

}

After That we have open Postman and check

In postman---

<http://localhost:8082/studentAdd>

{

"sfirstName": "Rajkumar",

"slastName": "Sharma",

"scourse": "Java"

}

Status 200 Ok

Also we can verify from the data table created and updated in Cassandra database.

cqlsh:mykeyspace> select \* from student;

**sid** | **scourse** | **sfirstname** | **slastname**

-----------+----------+------------+-----------

**102** | **angular** | **sumit** | **sharna**

**234236351** | **Java** | **Rajkumar** | **Sharma**

**467872752** | **Angular6** | **Ram** | **Sudheer**

**101** | **java** | **Ankit** | **gupta**

Now Succusefully working restapi with Cassandra database.

Note- Also we have ton create the table in database then we are able to connect our application properly.

Thanks you

CREATE TABLE student ( sid TEXT PRIMARY KEY, sfirstName TEXT, slastName TEXT , scourse TEXT );

insert into student(sid,sfirstname,slastname,scourse)values('101','Ankit','gupta','java');