

# Getting Started

NucleoDB Spring Repository Library makes it easier to use NucleoDB.

## Installation

### Dependencies

- Kafka Cluster
  - `/docker/kafka/docker-compose.yml`

### Import library

```
repositories {  
    mavenCentral()  
    maven { url "https://nexus.synload.com/repository/maven-repo-releases/" }  
}  
dependencies {  
    implementation 'com.nucleodb:spring:3.3.49'  
}
```

### Initializing DB

```
@SpringBootApplication  
@EnableNDBRepositories(  
    dbType = NucleoDB.DBType.NO_LOCAL, // does not create a disk of current up-to-date version of DB  
    // Feature: Read To Time, will read only changes equal to or before the date set.  
    //readToTime = "2023-12-17T00:42:32.906539Z",  
    scanPackages = {  
        "com.package.string" // scan for @Table classes  
    },  
    basePackages = "com.package.string.repos"  
)  
class Application{  
    public static void main(String[] args) {
```

```
SpringApplication.run(Application.class);  
}  
}
```

#### DataEntry model files

```
import java.io.Serializable;  
  
@Table(tableName = "author", dataEntryClass = AuthorDE.class)  
public class Author implements Serializable {  
    @Index  
    String name;  
    public Author(String name) {  
        this.name = name;  
    }  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
}  
  
public class AuthorDE extends DataEntry<Author>{  
  
}
```

#### Repository for DataEntry

```
@Repository  
public interface AuthorDataRepository extends NDBDataRepository<AuthorDE, String>{  
    Set<AuthorDE> findByNameAndKey(String name, String key);  
    Author findByName(String name);  
    void deleteByName(String name);  
}
```

#### Connection model files

```
@Conn("CONNECTION_BETWEEN_DE")  
public class ConnectionBetweenDataEntryClasses extends Connection<ConnectingToDataEntryDE,  
    ConnectingFromDataEntryDE>{
```

```
public ConnectionBetweenDataEntryClasses() {  
}  
  
public ConnectionBetweenDataEntryClasses(ConnectingFromDataEntryDE from, ConnectingToDataEntryDE to)  
{  
    super(from, to);  
}  
  
public ConnectionBetweenDataEntryClasses(ConnectingFromDataEntryDE from, ConnectingToDataEntryDE to,  
Map<String, String> metadata) {  
    super(from, to, metadata);  
}  
}
```

Repository for Connections

```
@Repository  
public interface ConnectionRepository extends NDBConnRepository<ConnectionBetweenDataEntryClasses,  
String, ConnectingFromDataEntryDE, ConnectingFromDataEntryDE>{  
  
}
```

---

Revision #3

Created 23 December 2023 19:07:32 by Nathaniel

Updated 29 January 2024 09:34:27 by Nathaniel