Build a Spring Boot based Blog site application with MongoDB

Step 1: Setting the Development Environment

Step 2: Including the Code Base

Step 3: Writing in the CRUD operations for BLOG application.

Step 4: Set up the spring application using Mongo Template

Step 5: Create a Database and Collection in MongoDB required using the Backend Application

Step 6: Import the Data in Mongo

Step 7: Perform the aggregation operations by criteria’s provided.

1. **Setting the Development Environment**

We can choose either IntelliJ , Eclipse , Spring tool Suite for the Development Process

Go to link: <https://start.spring.io/>

Select the Project accordingly for either Gradle or Maven.

Select the Language Java or Kotlin

Select the Spring boot version and also we can modify the name and artifact given and also select the java version

At the right side we can select the dependencies required for this application

After selecting the dependencies tab search for

* + - Lombok
    - Spring data JPA
    - Spring data MongoDB

After selecting the dependencies click on generate and this will download a JAR file

This JAR file must be imported in IntelliJ.

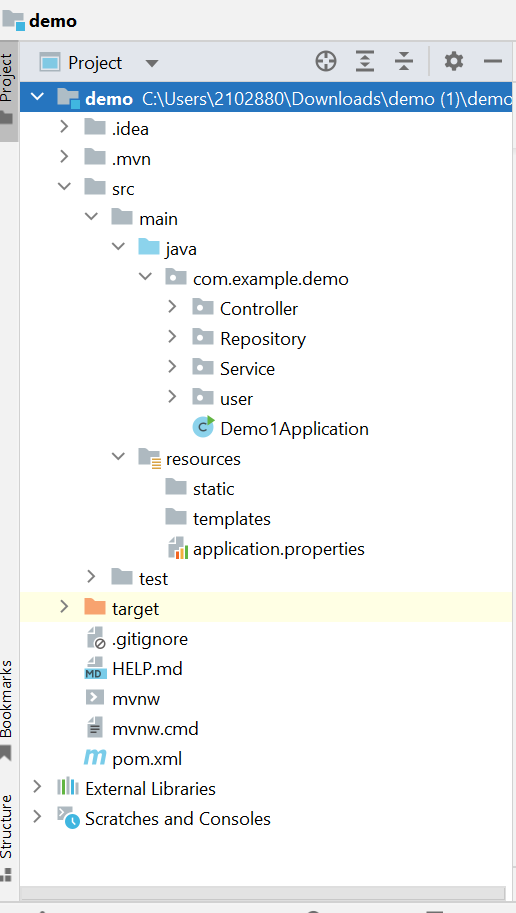
1. Including the Code Base

After Importing the JAR file in IntelliJ we will get the following structure

Where we have to Append what ever the Packages are required for our application such as :

* Controller
* Service
* Model
* Repository(if required)

Create java classes in each package accordingly so that each class should have the data accordingly



The controller class consist of the Admin class and also the User controller class where each class consists of their own data respectively.

Admin Controller should consists of the CRUD operations such as

Admin can CREATE, POST, UPDATE and also DELETE all the Blog Posts which are added by the users.

* getAllUsers
* postBlog
* postComment
* deleteBlog
* deleteUser
* RestrictUser

USER Controller

* Post the Blogs, stories , Like , Comment, Tag the users
* Create the account either public or private
* Delete the Blogs posted by the User
* Using Get Methd we can get the followers of the users
* Users can get to see their likes and comments of their blog

Service Layer

Include all the code required for MongoDB and the logic part in the service class itself

It can also consists of the admin and user service classes

Admin Service

* Can save the Blogs created by all the users and save them in the database.
* Execute all the logic part for what ever we are having in the controller such as the get all the users
* Can Delete any users blog which is inappropriate.

User Service

* <getAllUsers>and save
* <postStory>
* <postComment>
* <postBlog>
* <deleteByUser>
* <deleteStory>
* <deleteComment>

Setting Up the Mongo Template In the Service class

We just need to Include Mongo Template attribute in our service class in order to connect to the mongodb

Creating the Model Class

These model classes should be consisting of the entity class so that can create the collection in the database

@Document(collection = “userdetails”

User Class {

Name<string>

Id (int)

Number of posts<int>

Profile <img>

Bio <string>

DOB <date>

Number of Followers <int>

}

Admin class {

Int Id

Int numberOfusers

String Name

}

Setup the Application properties

server.port=8081  
spring.data.mongodb.host=localhost  
spring.data.mongodb.port=27017  
spring.data.mongodb.database=Demo

We can provide the database name and also the port which we are going to use in the mongo in the properties

Now We need to create a new connection in the Mongo and use the same port number in the mongo

A screenshot of a computer

Description automatically generated

After ruuning the Application We can be able to see the Database which is created in the MongoDB

Where we can also see the Collection created in the database

By using the Mongo we can create the aggregation based on the criteria where we can use the $match, $project, $sort

Based on $match we can get the users where they are having the same name , number of followers, number of likes etc.,

Based on $project we can remove the criterias which are not required something like the profile, DOB

Based on the $sort we can sort the users based on the likes and comments and the number of users

So these can be done by using the Aggregation Pipeline .

Done By :

Bayyarapu Hemasree.