

# Integrate MongoDB within SpringBoot

## Install MongoDB

Install MongoDB using brew install (MacOS):

```
brew install mongodb-community@4.4
```

Install MongoDB in other OS see: <https://docs.mongodb.com/manual/installation/>

Check your MongoDB version:

```
$ mongod --version
```

You may also use visualization tools to make MongoDB easier, here we recommend ROBO 3T, see <https://robomongo.org/> for help.

## Configure MongoDB in SpringBoot

Add MongoDB support in pom.xml:

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-data-mongodb</artifactId>  
</dependency>
```

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-thymeleaf</artifactId>  
</dependency>
```

MongoDB properties:

```
#mongodb configuration
#spring.data.mongodb.database=coachingmate
#spring.data.mongodb.host=8.129.65.2
#spring.data.mongodb.password=unimelb
#spring.data.mongodb.port=27017
#spring.data.mongodb.username=coachingmate
```

or:

```
spring.data.mongodb.uri = mongodb://coachingmate:unimelb@8.129.65.2:27017/coachingmate
```

Create UserEntity:

```
@Document(collection = "users")
public class UserEntity implements Serializable {
    private static final long serialVersionUID = -3258839839160856613L;
    @Field("id")
    private Long id;
    @Field("userName")
    private String userName;
    @Field("password")
    private String passWord;

    public Long getId() { return id; }

    public void setId(Long id) { this.id = id; }

    public String getUserName() { return userName; }

    public void setUserName(String userName) { this.userName = userName; }

    public String getPassWord() { return passWord; }

    public void setPassWord(String passWord) { this.passWord = passWord; }

    @Override
    public String toString() {
        return "UserEntity{" +
            "id=" + id +
            ", userName='" + userName + '\'' +
            ", passWord='" + passWord + '\'' +
            '}';
    }
}
```

Implement the Entity:

```

@Component
public class MongoTest implements UserDao {
    @Autowired
    private MongoTemplate mongoTemplate;

    /**
     * @Description test remote mongodb using collection name
     * @Date 23:24 22/9/20
     * @param user
     * @return {@link }
     */
    public void saveUser(UserEntity user) { mongoTemplate.save(user, "coachingmate"); }

    /**
     * @Description use to test the garmin connect push feature
     * @Date 23:24 22/9/20
     * @param activity
     * @return {@link }
     */
    public void saveActivityFile(String activity) { mongoTemplate.save(activity, "activity"); }

    public UserEntity findUserByUserName(String userName) {
        Query query=new Query(Criteria.where("userName").is(userName));
        UserEntity user = mongoTemplate.findOne(query , UserEntity.class);
        return user;
    }

    public int updateUser(UserEntity user) {
        Query query=new Query(Criteria.where("id").is(user.getId()));
        Update update= new Update().set("userName", user.getUserName()).set("password", user.getPassword());
        //更新查询返回结果集的第一条
        UpdateResult result =mongoTemplate.updateFirst(query,update,UserEntity.class);
        //更新查询返回结果集的所有
        // mongoTemplate.updateMulti(query,update,UserEntity.class);
        if(result!=null)
            return (int) result.getMatchedCount();
        else
            return 0;
    }

    @Override
    public void deleteUserById(Long id) {
        Query query=new Query(Criteria.where("id").is(id));
        mongoTemplate.remove(query,UserEntity.class);
    }
}

```

## Run test cases

Start MongoDB client on localhost or proxy first.

Test CRUD of MongoDB:

```

public class UserDaoTest extends AppTest {

    @Autowired
    private UserDao userDao;

    @Test
    public void testSaveUser() throws Exception {
        UserEntity user=new UserEntity();
        user.setId(21);
        user.setUserName("tao");
        user.setPassword("xiaoming1");
        userDao.saveUser(user);
    }

    @Test
    public void findUserByUserName(){
        UserEntity user= userDao.findUserByUserName("tao");
        System.out.println("user is "+user);
    }

    @Test
    public void updateUser(){
        UserEntity user=new UserEntity();
        user.setId(21);
        user.setUserName("xiaohong");
        user.setPassword("xiaohong1");
        userDao.updateUser(user);
    }

    @Test
    public void deleteUserById() { userDao.deleteUserById(11); }
}

```

Test results:

(5) ObjectId("5f633e9aab45ec0e75567551")	{ 5 fields }	Object
_id	ObjectId("5f633e9aab45ec0e75567551")	ObjectId
id	2	Int64
userName	tao	String
password	xiaoming1	String
_class	coachingmateanalytics.coachingmate.entity.UserEntity	String