一、集成简介

spring-data-mongodb提供了MongoTemplate与MongoRepository两种方式访问mongodb，MongoRepository操作简单，MongoTemplate操作灵活，我们在项目中可以灵活适用这两种方式操作mongodb，MongoRepository的缺点是不够灵活，MongoTemplate正好可以弥补不足。

1、在service_hosp引入依赖

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

```

2、添加MongoDB配置

```
spring.data.mongodb.uri=mongodb://192.168.44.165:27017/test
```

二、基于MongoTemplate 开发CRUD

1、添加实体

添加com.atguigu.yygh.hosp.testmongo.User类

```
@Data
@Document("User")
public class User {
    @Id
    private String id;
    private String name;
    private Integer age;
    private String email;
```

```
    private String createDate;
}
```

2、常用方法

常用方法

mongoTemplate.findAll(User.class): 查询User文档的全部数据

mongoTemplate.findById(<id>, User.class): 查询User文档id为id的数据

mongoTemplate.find(query, User.class);: 根据query内的查询条件查询

mongoTemplate.upsert(query, update, User.class): 修改

mongoTemplate.remove(query, User.class): 删除

mongoTemplate.insert(User): 新增

Query对象

1、创建一个query对象（用来封装所有条件对象），再创建一个criteria对象（用来构建条件）

2、 精准条件：criteria.and(“key”).is(“条件”)

模糊条件：criteria.and(“key”).regex(“条件”)

3、封装条件：query.addCriteria(criteria)

4、大于（创建新的criteria）：Criteria gt = Criteria.where(“key”).gt（“条件”）

小于（创建新的criteria）：Criteria lt = Criteria.where(“key”).lt（“条件”）

5、Query.addCriteria(new Criteria().andOperator(gt,lt));

6、一个query中只能有一个andOperator()。其参数也可以是Criteria数组。

7、排序：query.with（new Sort(Sort.Direction.ASC, "age"). and(new Sort(Sort.Direction.DESC, "date")))

3、添加测试类

```
@RestController
@RequestMapping("/mongo1")
public class TestMongo1 {

    @Autowired
    private MongoTemplate mongoTemplate;

    //添加
    @GetMapping("create")
    public void createUser() {
```

```
        User user = new User();
        user.setAge(20);
        user.setName("test");
        user.setEmail("4932200@qq.com");
        User user1 = mongoTemplate.insert(user);
        System.out.println(user1);
    }

    //查询所有
    @GetMapping("findAll")
    public void findUser() {
        List<User> userList = mongoTemplate.findAll(User.class);
        System.out.println(userList);
    }

    //根据id查询
    @GetMapping("findId")
    public void getById() {
        User user =
                mongoTemplate.findById("5ffbfa2ac290f356edf9b5aa", User.class);
        System.out.println(user);
    }

    //条件查询
    @GetMapping("findUser")
    public void findUserList() {
        Query query = new Query(Criteria
                .where("name").is("test")
                .and("age").is(20));
        List<User> userList = mongoTemplate.find(query, User.class);
        System.out.println(userList);
    }

    //模糊查询
    @GetMapping("findLike")
    public void findUsersLikeName() {
        String name = "est";
        String regex = String.format("%s%s%s", "^.*", name, ".*$");
        Pattern pattern = Pattern.compile(regex, Pattern.CASE_INSENSITIVE);
        Query query = new Query(Criteria.where("name").regex(pattern));
        List<User> userList = mongoTemplate.find(query, User.class);
        System.out.println(userList);
    }
```

```
54
55      //分页查询
56      @GetMapping("findPage")
57      public void findUsersPage() {
58          String name = "est";
59          int pageNo = 1;
60          int pageSize = 10;
61
62          Query query = new Query();
63          String regex = String.format("%s%s%s", "^.*", name, ".*$");
64          Pattern pattern = Pattern.compile(regex, Pattern.CASE_INSENSITIVE);
65          query.addCriteria(Criteria.where("name").regex(pattern));
66          int totalCount = (int) mongoTemplate.count(query, User.class);
67          List<User> userList = mongoTemplate.find(query.skip((pageNo - 1) * pageSize)
68
69          Map<String, Object> pageMap = new HashMap<>();
70          pageMap.put("list", userList);
71          pageMap.put("totalCount",totalCount);
72          System.out.println(pageMap);
73      }
74
75      //修改
76      @GetMapping("update")
77      public void updateUser() {
78          User user = mongoTemplate.findById("5ffbfa2ac290f356edf9b5aa", User.class);
79          user.setName("test_1");
80          user.setAge(25);
81          user.setEmail("493220990@qq.com");
82          Query query = new Query(Criteria.where("_id").is(user.getId()));
83          Update update = new Update();
84          update.set("name", user.getName());
85          update.set("age", user.getAge());
86          update.set("email", user.getEmail());
87          UpdateResult result = mongoTemplate.upsert(query, update, User.class);
88          long count = result.getModifiedCount();
89          System.out.println(count);
90      }
91
92      //删除操作
93      @GetMapping("delete")
94      public void delete() {
95          Query query =
96                  new Query(Criteria.where("_id").is("5ffbfa2ac290f356edf9b5aa"));
```

```
97          DeleteResult result = mongoTemplate.remove(query, User.class);
98          long count = result.getDeletedCount();
99          System.out.println(count);
100     }
101
102 }
```