



## 《Spring Boot 基础教程》

## 第 14 节

## 使用 SQL 关系型数据库-spring-data-jpa

## 一、配置数据源：

```
<!-- 数据库 -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <scope>runtime</scope>
</dependency>
```

## 二、Jpa 配置

```
# JPA
spring.jpa.hibernate.ddl-auto= update
#显示 sql 语句
spring.jpa.show-sql=true
```

## 三、实体类

```
@Entity
public class RoncooUserLog {

    @Id
    @GeneratedValue
    private Integer id;

    @Column
    private Date createTime;

    @Column
    private String userName;

    @Column
    private String userIp;

    public Integer getId() {
        return id;
    }
}
```



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

public Date getCreateTime() {
    return createTime;
}

public void setCreateTime(Date createTime) {
    this.createTime = createTime;
}

public String getUsername() {
    return userName;
}

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

public String getUserIp() {
    return userIp;
}

public void setUserIp(String userIp) {
    this.userIp = userIp;
}

@Override
public String toString() {
    return "RoncooUserLog [id=" + id + ", createTime=" + createTime + ", userName="
+ userName + ", userIp=" + userIp + "]";
}

}
```

#### 四、定义接口（继承 JpaRepository）

```
public interface RoncooUserLogDao extends JpaRepository<RoncooUserLog, Integer>{
}
```

#### 五、测试

```
@Autowired
private RoncooUserLogDao roncooUserLogDao;
```



```
@Test
public void insert() {
    RoncooUserLog entity = new RoncooUserLog();
    entity.setUserName("无境");
    entity.setUserIp("192.168.0.1");
    entity.setCreateTime(new Date());
    roncooUserLogDao.save(entity);
}

@Test
public void delete() {
    roncooUserLogDao.delete(1);
}

@Test
public void update() {
    RoncooUserLog entity = new RoncooUserLog();
    entity.setId(2);
    entity.setUserName("无境 2");
    entity.setUserIp("192.168.0.1");
    entity.setCreateTime(new Date());
    roncooUserLogDao.save(entity);
}

@Test
public void select() {
    RoncooUserLog result = roncooUserLogDao.findOne(1);
    System.out.println(result);
}
```

## 六、扩展进行自定义的查询

### 1. 使用内置的关键词查询

<http://docs.spring.io/spring-data/jpa/docs/1.10.2.RELEASE/reference/html/>

### 2. 使用自定义语句查询

各种语法，具体查看

<http://docs.spring.io/spring-data/jpa/docs/1.10.2.RELEASE/reference/html/>

这个优先级比上面的高

```
@Query(value = "select u from RoncooUserLog u where u.userName=?1")
```

测试

```
@Test
```



```
public void select2() {
    List<RoncooUserLog> result =
roncooUserLogDao.findByUserName("无境");
    System.out.println(result);
}

@Test
public void select3() {
    List<RoncooUserLog> result =
roncooUserLogDao.findByUserNameAndUserIp("无境",
"192.168.0.1");
    System.out.println(result);
}

// 分页
@Test
public void queryForPage() {
    Pageable pageable = new PageRequest(0, 20, new Sort(new
Order(Direction.DESC, "id")));
    Page<RoncooUserLog> result =
roncooUserLogDao.findByUserName("无境", pageable);
    System.out.println(result.getContent());
}
```

更多课程信息，请关注 龙果学院 官方网站 <http://www.roncoo.com/>

或关注 龙果 微信公众号 RonCoo\_com

