

集成mybatis-plus与数据库连接池

我们将以oj-system服务为例，我们一起写一个测试类，集成mybatis-plus和数据库连接池

pom.yml

```
1 <!-- mybatis-plus -->
2 <dependency>
3     <groupId>com.baomidou</groupId>
4     <artifactId>mybatis-plus-spring-boot3-starter</artifactId>
5     <version>3.5.5</version>
6 </dependency>
7
8
9 <dependency>
10    <groupId>com.mysql</groupId>
11    <artifactId>mysql-connector-j</artifactId>
12 </dependency>
```

controller

```
1 package com.bite.system.test;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.RequestMapping;
6 import org.springframework.web.bind.annotation.RestController;
7
8 import java.util.List;
9
10 @RestController
11 @RequestMapping("/test")
12 public class TestController {
13
14     @Autowired
15     private ITestService testService;
16
17     @GetMapping("/list")
18     public List<?> list() {
19         return testService.list();
20     }
21 }
```

```
20     }  
21 }
```

Service

```
1 public interface ITestService {  
2     List<TestDomain> list();  
3 }
```

```
1 @Service  
2 public class TestServiceImpl implements ITestService {  
3  
4     @Autowired  
5     private TestMapper testMapper;  
6  
7     @Override  
8     public List<TestDomain> list() {  
9         List<TestDomain> testDomains = testMapper.selectList(new  
10             LambdaQueryWrapper<>());  
11         System.out.printf("%s");  
12         return testDomains;  
13     }  
14 }
```

Domain

```
1 @TableName("tb_test")  
2 public class TestDomain {  
3  
4     private Integer testId;  
5  
6     private String title;  
7  
8     public Integer getTestId() {  
9         return testId;  
10    }  
11  
12    public void setTestId(Integer testId) {  
13        this.testId = testId;  
14    }  
15 }
```

```

14     }
15
16     public String getTitle() {
17         return title;
18     }
19
20     public void setTitle(String title) {
21         this.title = title;
22     }
23 }

```

连接配置

```

1 spring:
2   datasource:
3     url: jdbc:mysql://localhost:3306/bitobj_dev?
        useUnicode=true&characterEncoding=utf8&useSSL=true&serverTimezone=GMT%2B8
4     username: ojtest
5     password: 123456
6     hikari:
7       minimum-idle: 5 # 最小空闲连接数
8       maximum-pool-size: 20 # 最大连接数
9       idle-timeout: 30000 # 空闲连接存活时间 (毫秒)
10      connection-timeout: 30000 # 连接超时时间 (毫秒)
11
12
13      ##connection-test-query: SELECT 1 如果驱动支持 JDBC4.0, 建议不设置, 这时默认会调
        用 Connection.isValid() 来检查, 该方式会更高效一些
14      ##keepalive-time 多久检查一次连接的活性 默认不启用

```

启动类修改:

`@MapperScan` 是 Spring Boot 中 MyBatis 集成的一个注解, 用于指定 MyBatis Mapper 接口所在的包路径。当 Spring Boot 应用程序启动时, 它会扫描这个包路径下的所有接口, 并将它们注册为 MyBatis 的 Mapper。

```

1 import org.mybatis.spring.annotation.MapperScan;
2 import org.springframework.boot.SpringApplication;
3 import org.springframework.boot.autoconfigure.SpringBootApplication;
4
5 @SpringBootApplication
6 @MapperScan("com.bite.**.mapper")

```

```
7 public class OjSystemApplication {  
8  
9     public static void main(String[] args) {  
10         SpringApplication.run(OjSystemApplication.class, args);  
11     }  
12 }
```

比特就业课