



SSM J2EE企业级项目开发

SSM-MyBatis框架

日照职业技术学院电子信息工程学院 • 盛雯雯

准备工作

Spring**整合**MyBatis

- Spring框架提供了IOC机制，可以管理所有组件的创建工作并进行依赖管理。因而整合的主要工作是把MyBatis框架中使用的核心组件配置到Spring容器中。交给Spring容器创建和管理。

准备工作

- 1.新建一个基于Maven的java项目
- 2.创建数据库与表
- 3.添加依赖包（配置pom文件）
- 4.新建POJO实体层：为了实现与数据库中的books表进行关系映射新建一个Book类

5.新建MyBatis SQL映射层

(1) 新建接口BookDAO

这个项目中我们采用接口与xml结束的形式完成关系与对象间的映射，在接口中定义一些数据访问的方法，在xml文件中定义实现数据访问需要的sql脚本。

一般我们都会定义增删改查。

5.新建MyBatis SQL映射层

(2) 创建Mapper映射文件

为MyBatis ORM创建的映射文件BookMapper.xml (命名尽量都遵循一个规则 , 便于扫描 , 这里约定以实体名+Mapper)

- 命名空间应该是对应接口的包名+接口名
- 注意id的命名规则。

6.完成Spring整合MyBatis配置

(1) db.properties

`jdbc.driver=com.mysql.cj.jdbc.Driver` , 注意要与数据库版本一致

用于存放数据库连接信息

6.完成Spring整合MyBatis配置

(2) applicationContext.xml

在源代码的根目录下新建 applicationContext.xml文件，用于整合MyBatis与Spring，该文件是整个项目的控制中心，非常关键。

- *配置C3P0数据源*
- *配置会话工厂bean sqlSessionFactoryBean*
- *配置自动扫描对象关系映射*
- *声明式事务管理*
- *容器自动扫描IOC组件*
- *aspectj支持自动代理实现AOP功能*

(1) **配置** *sqlSessionFactoryBean*

- 配置完数据源，就可以在此基础上集合SQL映射文件信息以及MyBatis配置文件中的其他信息，创建SQLSessionFactory实例。
- 在MyBatis中，SqlSessionFactory的实例需要使用SqlSessionFactoryBuilder创建，而在集成环境中，可以使用MyBatis-spring整合包中的SqlSessionFactoryBean来代替。
- mapperLocations属性扫描式加载SQL映射文件。

(2) **注入映射器** - MapperScannerConfigurer

- `basePackage` : 指定扫描基准包
- `sqlSessionFactoryBeanName` : 指定sqlSessionFactory

(3) 声明事务

- 业务层(Service)的职能不单单是调用DAO这么简单,事务处理是任何企业级应用中都不可能回避的一个问题.
- Spring提供了事务管理机制,它基于AOP实现,无须编写任何事务管理代码,所有工作都在配置文件中完成.
- 业务与代码完全分离,配置即可用.
- @Transaction注解事务

七、创建服务层

- 创建BookService服务类，完成图书管理业务，有些项目中也叫业务层，这里我们叫服务层.
- 服务层不只是一个dao的接力棒，认为他可有可无，其实是因为我们现在的的示例中没有涉及到更多的复杂业务，所以显得比较空，实现开发可能有更多的业务逻辑要在这里处理。另外给bookdao成员变量注解为自动装配，service类注解为IOC组件。

八、JUnit测试服务类

- `ApplicationContext ctx=new
ClassPathXmlApplicationContext("applicationContext.xml");
bookservice=ctx.getBean(BookService.class);`
- `List<Book> books=bookservice.getAllBooks();`
-