

参考文档

简介

入门

SqlSessionFactoryBean

事务

使用 SqlSession

注入数据源

使用 Spring Boot

使用 MyBatis API


使用 Spring Batch

示例代码

返回文档

运营信息

项目图表



入门

本文将会以底层的步骤告诉你如何安装和配置 MyBatis-Spring，并构建一个简单的具备事务管理功能的数据访问应用程序。

安装

要使用 MyBatis-Spring 模块，只需要在类路径下包含 `mybatis-spring-3.0.4.jar` 文件和相关依赖即可。

如果使用 Maven 作为构建工具，只需要在 pom.xml 中加入以下代码即可：

```
<dependency>
<groupId>org.mybatis</groupId>
<artifactId>mybatis-spring</artifactId>
<version>3.0.4</version>
</dependency>
```

快速上手

要和 Spring 一起使用 MyBatis，需要在 Spring 应用上下文中定义至少两样东西：一个 `SqlSessionFactory` 和至少一个数据映射器类。

在 MyBatis-Spring 中，可使用 `SqlSessionFactoryBean` 来创建 `SqlSessionFactory`。要配置这个工厂 bean，只需要把下面代码放在 Spring 的 XML 配置文件中：

```
<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
<property name="dataSource" ref="dataSource" />
</bean>
```

```
@Configuration
public class HybatisConfig {
    @Bean
    public SqlSessionFactory sqlSessionFactory() throws Exception {
        SqlSessionFactoryBean factoryBean = new SqlSessionFactoryBean();
        factoryBean.setDataSource(dataSource());
        return factoryBean.getObject();
    }
}
```

注意：`SqlSessionFactory` 需要 一个 `DataSource`（数据源）。这可以是任意的 `DataSource`，只需要和配置其它 Spring 数据库连接一样配置它就可以了。

假设你定义了一个如下的 mapper 接口：

```
public interface UserMapper {
    @select("SELECT * FROM users WHERE id = #{userId}")
    User getUser(@Param("userId") String userId);
}
```

那么可以通过 `MapperFactoryBean` 将接口注入到 Spring 中

```
<bean id="userMapper" class="org.mybatis.spring.mapper.MapperFactoryBean">
<property name="mapperInterface" value="org.mybatis.spring.sample.mapper.UserMapper" />
<property name="sqlSessionFactory" ref="sqlSessionFactory" />
</bean>
```

需要注意的是：所指定的映射器类必须是一个接口，而不是具体的实现类。在这个示例中，通过注解来指定 SQL 语句，但是也可以使用 MyBatis 映射器的 XML 配置文件。

配置好之后，你就可以像 Spring 中普通的 bean 注入方法那样，将映射器注入到你的业务或服务对象中。`MapperFactoryBean` 将会负责 `SqlSession` 的创建和管理。如果使用了 Spring 的事务功能，那么当事务完成时，session 将会被提交或回滚。最终任何异常都会被转码成 Spring 的 `DataAccessException` 异常。

使用 Java 代码来配置的方式如下：

```
@Configuration
public class HybatisConfig {
    @Bean
    public UserMapper userMapper() throws Exception {
        SqlSessionTemplate sqlSessionTemplate = new SqlSessionTemplate(sqlSessionFactory());
        return sqlSessionTemplate.getMapper(UserMapper.class);
    }
}
```

要调用 MyBatis 的数据方法，只需一行代码：

```
public class FooServiceImpl implements FooService {

    private final UserMapper userMapper;

    public FooServiceImpl(UserMapper userMapper) {
        this.userMapper = userMapper;
    }

    public User doSomeBusinessStuff(String userId) {
        return this.userMapper.getUser(userId);
    }
}
```