

快速入门	
简介	
入门	
SqlSessionFactoryBean	
事务	>
使用 SqlSession	>
注入映射器	>
使用 Spring Boot	
使用 MyBatis API	
使用 Spring Batch	
示例代码	
返回主页	
项目信息	>
项目版本	>



使用 MyBatis API

使用 MyBatis-Spring, 你可以继续直接使用 MyBatis 的 API. 只需简单地使用 `SqlSessionFactoryBean` 在 Spring 中创建一个 `SqlSessionFactory`, 然后按你的方式在代码中使用工厂即可。

```
public class UserDaoImpl implements UserDao {
    // SqlSessionFactory 一般由 SqlSessionFactorySupport 进行设置
    private final SqlSessionFactory sqlSessionFactory;

    public UserDaoImpl(SqlSessionFactory sqlSessionFactory) {
        this.sqlSessionFactory = sqlSessionFactory;
    }

    public User getUser(String userId) {
        // 注意对标准 MyBatis API 的使用 - 手工打开和关闭 session
        try (SqlSession session = sqlSessionFactory.openSession()) {
            return session.selectOne("org.mybatis.spring.sample.mapper.UserMapper.getUser", userId);
        }
    }
}
```

小心使用此选项。错误地使用会产生运行时错误，更糟糕地，会产生数据一致性的问题。滥用使用 API 时，注意以下弊端：

- 它不会参与 Spring 的事务管理之中。
- 如果 `SqlSession` 使用与 Spring 事务管理使用的相同 `DataSource`，并且有进行中的事务，代码将会抛出异常。
- MyBatis 的 `DefaultSqlSession` 是线程不安全的。如果在 bean 中注入了它，将会发生错误。
- 使用 `DefaultSqlSession` 创建的映射器也不是线程安全的。如果你将它们注入到 bean 中，将会发生错误。
- 你必须确保总是在 `finally` 块中关闭 `SqlSession`。