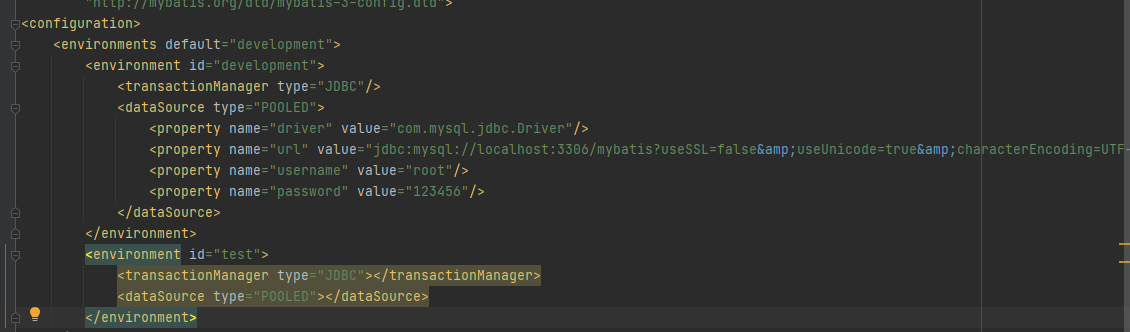
配置文件解析

1.environment



environment改个id就能有多个environment,默认事务管理器是JDBC,还有一个managed，不常用

、默认dataSource是POOLED, WEB响应更快

要学会的就是**：尽管可以配置多个环境，但每个 SqlSessionFactory 实例只能选择一种环境**

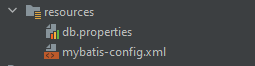
学会使用配置多套运行环境！

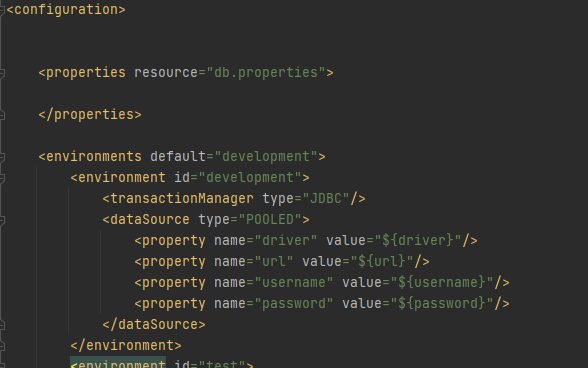
MyBatis默认的事务管理器就是JDBC ，连接池：POOLED

2.properties

这些属性可以在外部进行配置，并可以进行动态替换。你既可以在典型的 Java 属性文件中配置这些属性，也可以在 properties 元素的子元素中设置。【db.poperties】

换句话来说，现在可以用db.properties,然后environment里再去引用



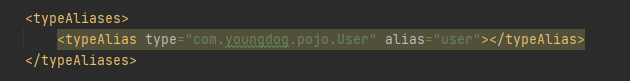


3.typeAliases

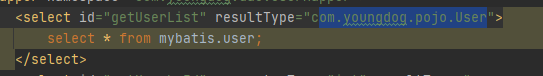
主要目的就是为类（尤其是pojo类）设置一个别名，这个别名仅用于.xml配置（例如Mapper的xml）

3.1

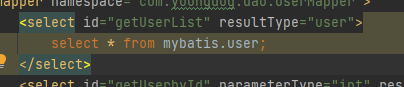
typeAlias



原来

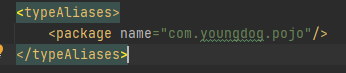


这样xml实装文件我们可以简写pojo类user



3.2

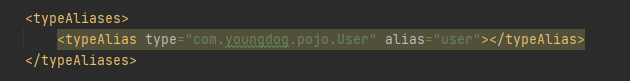
package



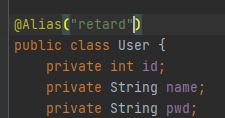
指定一个package，然后这个package目录下所有class将会有一个别名，这个别名默认为小写文件名字，例如



这时alias默认为user

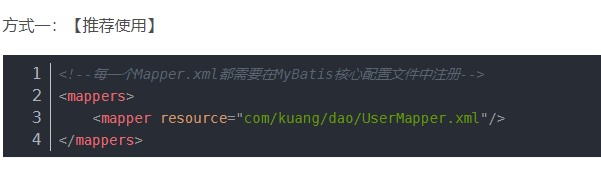


如果想改变默认，要使用Alias



4.映射器Mappers

注册绑定我们所写的mapper文件



方法二：



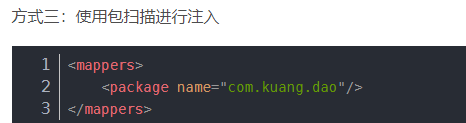
class的缺陷

1.接口和配置文件必须同名

比如接口叫UserMapper,那么配置文件就必须交UserMapper.xml

2.接口与配置文件必须在一个包下

方法三：



这个包下Mapper都被注册

还是老问题，接口和配置文件必须同名

5.SqlSession的作用域与生命周期

第一层：sqlSessionFactoryBuilder：有一个，只用一次，局部

局部变量，目的就是创造一个sqlSessionFactory，一旦创造，我们就不关心这个builder了



就这一行代码要用

第二层：sqlSessionFactory：有一个，从头到尾存在，应用作用域

顾名思义，他是创造sqlSession的工厂，说白了我们可以把它想象成一个连接数据库的池，我们在上面那行代码让他连接数据库就成

SqlSessionFactory一旦被创建就应该在应用的运行期间一直存在，没有任何理由丢弃它或重新创建一个实例。

因此SqlSessionFactory的最佳作用域是应用作用域（ApplocationContext）。

最简单的就是使用单例模式或静态单例模式。//也就是说有且只有一个Factory，每个人都能得到他，从而从这个池里提取SqlSession或放回去销毁

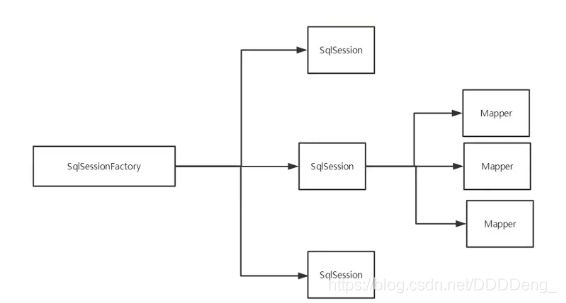
第三层：sqlSession：可以多个，非线程安全所以最好method结束以后销毁，节省资源

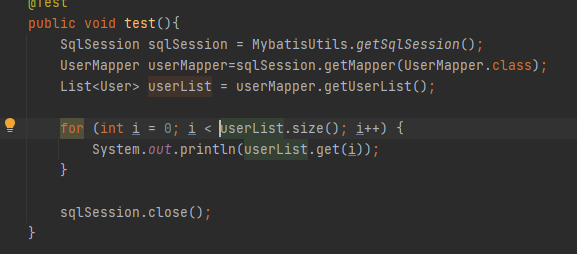
连接到连接池的一个请求

SqlSession 的实例不是线程安全的，因此是不能被共享的，所以它的最佳的作用域是请求或方法作用域。

用完之后需要赶紧关闭，否则资源被占用！

一个SqlSession可以注册多个Mapper,





我们用sqlSession与Mapper接口来创造对象，