

😊3、几个关键类

公众号java-mindmap

mybatis-config.xml

XML 配置文件（configuration XML）中包含了对 MyBatis 系统的核心设置，
包含获取数据库连接实例的数据源（DataSource）和决定事务作用域和控制方式的事务管理器（TransactionManager）

SqlSessionFactoryBuilder

SqlSessionFactoryBuilder通过类名就可以看出这个类的主要作用就是创建一个SqlSessionFactory，通过输入mybatis配置文件的字节流或者字符流，生成XMLConfigBuilder，XMLConfigBuilder创建一个Configuration，Configuration这个类中包含了mybatis的配置的一切信息，mybatis进行的所有操作都需要根据Configuration中的信息来进行。

作用域（Scope）和生命周期

可以重用 SqlSessionFactoryBuilder 来创建多个 SqlSessionFactory 实例，但是最好还是不要让其一直存在以保证所有的 XML 解析资源开放给更重要的事情
这个类可以被实例化、使用和丢弃，一旦创建了 SqlSessionFactory，就不再需要它了。因此 SqlSessionFactoryBuilder 实例的最佳作用域是方法作用域（也就是局部方法变量）

```
org.apache.ibatis.session
└─ SqlSessionFactoryBuilder
    ├── build(Reader) : SqlSessionFactory
    ├── build(Reader, String) : SqlSessionFactory
    ├── build(Reader, Properties) : SqlSessionFactory
    ├── build(Reader, String, Properties) : SqlSessionFactory
    ├── build(InputStream) : SqlSessionFactory
    ├── build(InputStream, String) : SqlSessionFactory
    ├── build(InputStream, Properties) : SqlSessionFactory
    ├── build(InputStream, String, Properties) : SqlSessionFactory
    └─ build(Configuration) : SqlSessionFactory
```

SqlSessionFactory接口

概念

作用域（Scope）和生命周期

SqlSessionFactory 一旦创建就应该在应用的运行期间一直存在，没有任何理由对它进行清除或重建
最佳作用域是应用作用域。有很多方法可以做到，最简单的就是使用单例模式或者静态单例模式。

使用xml构建

```
String resource = "org/mybatis/example/mybatis-config.xml";
InputStream inputStream = Resources.getResourceAsStream(resource);
SqlSessionFactory sqlSessionFactory = new SqlSessionFactoryBuilder().build(inputStream);
```

如何创建

java代码构建

```
DataSource dataSource = BlogDataSourceFactory.getBlogDataSource();
TransactionFactory transactionFactory = new JdbcTransactionFactory();
Environment environment = new Environment("development", transactionFactory, dataSource);
Configuration configuration = new Configuration(environment);
configuration.addMapper(BlogMapper.class);
SqlSessionFactory sqlSessionFactory = new SqlSessionFactoryBuilder().build(configuration);
```

SqlSession接口

概念

SqlSession是MyBatis的一个重要接口，定义了数据库的增删改查以及事务管理的常用方法。
SqlSession还提供了查找Mapper接口的有关方法。

作用域（Scope）和生命周期

每个线程都应该有它自己的 SqlSession 实例
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
每次收到的 HTTP 请求，就可以打开一个 SqlSession，返回一个响应，就关闭它。

如何创建

```
SqlSession session = sqlSessionFactory.openSession();
try {
    // do work
} finally {
    session.close();
}
```

Mapper接口

概念

承载了实际的业务逻辑，其生命周期比较短，由SqlSession创建,用于将Java对象和实际的SQL语句对应起来。
Mapper接口是指程序员自行定义的一个数据操纵接口，类似于通常所说的DAO接口。跟DAO不同的地方在于Mapper接口只需要程序员定义，不需要程序员去实现，MyBatis会自动为Mapper接口创建动态代理对象。Mapper接口的方法通常与Mapper配置文件中的select、insert、update、delete等XML结点存在一一对应关系。

实现方式

- (1)使用XML配置文件的方式。
- (2)使用注解方式。
- (3)直接使用MyBatis提供的API。

```
com.foo...XXMapper.java

public interface XXMapper{
    public List<?> querySome(...);

    public int updateSome(...);

    public int insertSome(...);

    public int deleteSome(...);
    ...
}
```

```
com/foo/.../XXMapper.xml

<mapper namespace="com.foo...XXMapper">
  <select id="querySome" parameterType="..." resultMap="...">
    ...
  </select>
  <update id="updateSome" parameterType="...">
    ...
  </update>
  <insert id="insertSome" parameterType="...">
    ...
  </insert>
  <delete id="deleteSome" parameterType="...">
    ...
  </delete>
</mapper>
```

Mapper接口和Mapper.xml配置文件之间的对应关系
Designed by LouLuan
http://blog.csdn.net/juanlouis