

# 一、基础知识

## 1.mybatis简介

ibatis是mybatis的前生

## 2.mybatis配置

### A.全局xml配置文件

MyBatis全局XML配置文件包含影响MyBatis行为的设置和属性。

#### (1) 属性(properties)

配置属性时作为占位符使用

xml中可与使用<property>标签去配置属性

也可以用api的方式去注册属性，但是这个顺序是晚于用xml配置文件的

组装API接口： org.apache.ibatis.session.Configuration#variables

填充配置：

org.apache.ibatis.builder.xml.XMLConfigBuilder(XPathParser,String,Properties)

#### (2) 设置(settings)

用于修改mybatis的运行时行为

组装API接口： org.apache.ibatis.session.Configuration#setXXX(\*)

填充配置： org.apache.ibatis.builder.xml.

XMLConfigBuilder#settingsElement(Properties) 其实就是set方法

#### (3) 类型别名(typeAliases)

为java类型建立别名，一般用更短的名称替代

组装API接口：

org.apache.ibatis.session.Configuration#typeAliasRegistry

API定义： org.apache.ibatis.type.TypeAliasRegistry

填充配置：

org.apache.ibatis.builder.xml.XMLConfigBuilder#typeAliasesElement(XNode)

#### (4) 类型处理器 (typeHandlers)

用于将编译语句 (preparedStatement) 或结果集 (ResultSet) 中的 JDBC 类型转化为 Java 类型，如：

BooleanTypeHandler 将 JDBC 类型中的 Boolean 转化成 Java 类型中的 java.lang.Boolean 或者 boolean

注意：如果需要转换 Java8 新增的 Date 和 TimeApi，即 JSR-310，需要再引入 mybatis-typehandlers-jsr310

```
<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis-typehandlers-jsr310</artifactId>
  <version>1.0.2</version>
</dependency>
```

Date 是可变的，并且是有时区的

组装 API 接口：

org.apache.ibatis.session.Configuration#typeHandlerRegistry

API 定义：org.apache.ibatis.type.TypeHandlerRegistry

填充配置：

org.apache.ibatis.builder.xml.XMLConfigBuilder#typeHandlerElement  
(XNode)

#### (5) 对象工厂 (objectFactory)

用于创建结果对象实例，提供默认构造器或者执行构造参数初始化目标类型的对象。通常使用场景，不需要调整默认的实现。

#### (6) 插件 (plugins)

mybatis 提供插件的方式来拦截映射语句 (mapped statement) 的执行

在源码中是责任链的方式处理的，如 ParameterHandler、ResultSetHandler、StatementHandler

#### (7) 环境 (environments)

mybatis允许配置多个环境，在运行时，通过传递环境信息，切换关联sqlSessionFactory实例。因此，MyBatis中的环境 (environment) 类似于Maven和Spring的Profile。

#### (8) 数据库标识提供商 (databaseIdProvider)

mybatis是面向Sql的映射框架，所执行Sql语句的语法依赖于数据库提供商的实现，比如 MySQL, Oracle, Sql server等。在配置映射Sql语句时，可为其指定具体的数据库提供商的实现。因此，在全局XML配置文件中可以定义多个数据库标识供应器

#### (9) Sql映射文件(mappers)

几种方式：

resource = xml文件的相对路径

class = mapper接口的路径

url = 通过协议去找到的文件

name = 包名下的接口

### B.sql Mapper XML配置文件

sql mapper xml配置用于映射SQL模板语句与Java类型的配置。

### C.3.0之后Sql提供的Sql Mapper Annotation

sql Mapper Annotation是java Annotation的方式替代Sql Mapper Xml配置文件

## 3.mybatis核心API

### A.builder

org.apache.ibatis.session.SqlSessionFactoryBuilder

SqlSessionFactory 构建器，创建 SqlSessionFactory 实例。通过重载方法build，控制实例行为，其中方法参数如下：

MyBatis全局配置流 (java.io.InputStream、java.io.Reader)

Mybatis环境名称 (environment)

Mybatis属性 (java.util.Properties)

相关API:

配置构建器: org.apache.ibatis.builder.xml.XMLConfigBuilder

MyBatis配置: org.apache.ibatis.session.Configuration

MyBatis环境: org.apache.ibatis.mapping.Environment

为什么核心API中的方法参数中会有流

因为在全局配置文件中配置中配置mapper文件的时候, 有很多多种方式配置, 有用流的方式读入的

B.Factory

MyBatis 核心API

org.apache.ibatis.session.SqlSessionFactory

SqlSession 工厂类, 创建 SqlSession 实例, 通过重载方法  
openSession, 控制实例特性, 其中方法参数如下:

是否需要自动提交

JDBC 数据库连接 (java.sql.Connection)

Mybatis SQL语句执行器类型

(org.apache.ibatis.session.ExecutorType)

Mybatis 事务隔离级别

(org.apache.ibatis.session.TransactionIsolationLevel)

C.SqlSession

MyBatis 核心API

org.apache.ibatis.session.SqlSession

MyBatis SQL 会话对象, 类似于JDBC Connection

职责

封装java.sql.Connection

屏蔽java.sql.Statement (以及派生接口) 的细节

映射java.sql.ResultSet 到Java类型

事务控制

缓存

代理映射 (Mapper)

#### 4. mybatis在spring boot中的starter

凡是spring-boot中的starter，都会有 xxxAutoConfiguration 这个类去装配。而其中的属性是在properties文件中进行装配的。而要配置哪些属性，是在xxxAutoConfiguration中@EnableConfigurationProperties注解中的Properties类中的属性。

这里要去进行配置mybatis-starter（配置包括mybatis的配置和数据源的配置）

typeHandler接口可以实现在对一些字段进行处理 比如null之类的