|  |
| --- |
| mybatis配置文件有 2 类：  主配置文件mybatis.xml，定义在类路径下，即普通项目src目录，maven项目src/main/resources目录；sql映射配置文件mapper.xml 名称随意，定义在\*.dao包里； |
| 约束文件：  要严格按照框架定义的配置文件格式进行书写，否则框架定义的解析器无法正确解析配置文件，为了防止用户随意书写配置文件，引入约束文件；  约束文件使用：  约束文件关联配置文件\*.xml，关联后，必须按照约束写配置文件，否则报错；  约束文件有 2 类：  dtd文件 \*.dtd；xmd文件 \*.xmd； |
|  |
| 数据库连接池：  为什么使用数据库连接池？  假设，取得数据库连接对象需要6个时间单位，操作数据库连接对象需要2个时间单位，关闭数据库连接对象需要2个时间单位，如果频繁创建，关闭数据库连接对象，浪费资源，因此，将数据库连接对象保存到本地内存中，使用时直接来本地内存取，使用完后，放回本地内存，不需要释放连接对象，本地内存即为数据库连接池； |

|  |
| --- |
| **获取类路径文件并读取文件内容到程序中** |
| 思路：  说道用JAVA获取文件，肯定会第一个想到用JAVA的File类，如要取得C:/test.txt文件，就会这样用，File file = new File("c:/test.txt"); 这样用有什么问题，相信大家都知道，路径硬编码，如果换到其他的运行环境下，这样写肯定是失败的，对于JAVA来说，代码应该做到一次编译，到处可用。  最优做法：（所有文件classpath路径下，即普通项目src目录下，maven项目resources目录下）  Url url = Demo.class.getResource(“/test.txt”);//获取test.txt文件url，必须加 /，并且该test.xml文件要定义在项目类路径classpath下，项目部署到不同环境时，会根据该相对路径补全，得到绝对路径；  String path = url.getFile();//获取资源全路径  File file = new File(path);//创建文件对象  InputStream in = new FileInputStream(file);//创建文件输入流  Reader reader = new InputStream(in);//字节输入流转为字符输入流  BufferedReader br = new BufferedReader(reader);//在reader对象外面包一层BufferedReader增强功能；  String temp = null;  while((temp = br.readLine()) != null){//读取文件到程序  System.out.println(temp);  } |

|  |
| --- |
| maven项目无法获取到 src/main/java下文件 |
| <build>  <finalName>mybatis</finalName>  <!--让项目启动的时候能够去读取到src/main/java下面的配置文件-->  <resources>  <resource>  <directory>src/main/java</directory>  <includes>  <include>\*\*/\*.\*</include>  </includes>  </resource>  </resources>  </build> |

|  |
| --- |
| 事务： |
| 概念：假设有3个操作，分别去改变数据库，如果有1个操作失败，则数据库回滚到初始状态，只有所有操作成功，才会提交，提交后才会真正改变数据库；  注意：只执行数据库操作语句，而不提交commit，数据库不会变化； |

|  |
| --- |
| sql映射文件 \*mapper.xml中命名空间namespace作用： |
| namespace相当于 \*mapper.xml的id号，用于区分 \*mapper.xml，实际中一个dao包里有很多 \*mapper.xml，需要区分 |

|  |
| --- |
| 线程安全： |
| 多线程并发访问单实例，该单实例存在可修改属性； |

|  |
| --- |
| 创建sqlsession对象： |
| sqlsession创建效率慢，因此采用单实例模式； |

|  |
| --- |
| 输入流输出流理解： |
| 输入输出流相当于一个管道，程序是输入流输出流的中转站，即输入流内容流入程序，程序通过输出流，将内容流到目的地  输入流：内容先装入输入管道，当输入流执行read()方法的时候，管道里面内容流到程序中，管道中不再有该内容，输入流关闭后，释放程序内存中资源；  输出流：输出流执行write()方法，向输出管道装入内容，输出流关闭后，内容流到目的地。 |
| 为什么最后要关闭流： |
| 1. 与流关联的文件会一直被占用，无法修改； 2. 例如，从文件读取10M内容到程序内存，虚拟机无法通过垃圾回收释放这些资源，只能你显式调用close方法来释放。 |

|  |
| --- |
| mybatis中dirty |
| dirty = false：数据库中数据与内存数据一致，不脏；  dirty = true：数据库中数据与内存数据不一致，脏； |

|  |
| --- |
| 打开xml配置文件的约束文件：F4 |
|  |
| 配置文件定义规则如下，在相应根元素下按顺序定义子元素 |

|  |
| --- |
| 省略DAO层接口实现类： |
| 1. \*mapper.xml的命名空间定义为接口全名称； 2. 接口调用方法名称与\*mapper.xml中id名称相同；   框架通过：接口名称.id名称找到相应sql语句； |
|  |
|  |