



河北师范大学软件学院
Software College of Hebei Normal University



MyBatis

第六讲 MyBatis其他配置



Java与移动智能设备开发



- 1 MyBatis配置优化**
- 2 MyBatis分页插件



■ MyBatis配置文件的元素结构如下：

➤ configuration (配置)

- ✓ properties (属性)
- ✓ settings (设置)
- ✓ typeAliases (类型别名)
- ✓ plugins (插件)
- ✓ environments (环境配置)
 - environment (环境变量)
 - transactionManager (事务管理器)
 - dataSource (数据源)
- ✓ mappers (映射器)



■ 配置别名 (typealiases)

- 类型别名是为 Java 类型命名一个短的名字。它只和XML配置有关, 只用来减少类完全限定名的多余部分。

```
<typeAliases>  
    <typeAlias type="net.onest.entity.User" alias="User"/>  
</typeAliases>
```



■ 配置别名 (typealiases)

- 也可以指定一个包名，MyBatis 会在包名下面搜索需要的 Java Bean

```
<typeAliases>  
    <package name="net.onest.entity"/>  
</typeAliases>
```



■ 数据源配置 (db.properties)

➤ 将数据库配置单独方法一个Java属性文件中

```
driver=com.mysql.jdbc.Driver  
url=jdbc:mysql://localhost:3306/mybatis?characterEncoding=utf-8  
username=root  
password=
```



■ 数据源配置 (db.properties)

- 使用properties元素在mybatis.xml配置文件中引用Java属性文件中的配置

```
<!-- 引用properties文件 -->
<properties resource="db.properties"></properties>

<environments default="development">
    <environment id="development">
        .....
    </environment>
</environments>
```



■ 数据源配置 (db.properties)

- 在dataSource中采用`${name}`的形式引用属性文件中定义的参数

```
<dataSource type="POOLED">
  <property name="driver" value="${driver}"/>
  <property name="url" value="${url}"/>
  <property name="username" value="${username}"/>
  <property name="password" value="${password}"/>
</dataSource>
```




■ MyBatis中\$和#的异同

- 可以获取对象中的属性值，`${userName}`和`#{userName}`相同
- `#`可以防止SQL注入，解析时会把所有使用`#`的地方变成？占位符，再设置参数的值
- `$`在解析时，会直接使用传入的参数作为字符串直接填充到SQL语句中，会导致SQL注入
- `#`会把传入的参数使用引号括起来，`$`则不会
- `$`一般用传入数据库相关参数，如数据库表名、字段名



- 1 MyBatis配置优化
- 2 MyBatis分页插件**

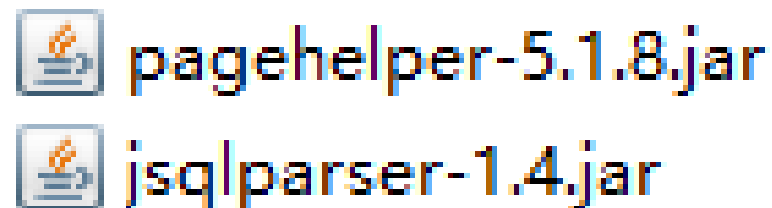


- MyBatis中要想实现分页功能，需要使用limit子句实现分页的SQL语句，比较麻烦
- PageHelper插件能方便解决MyBatis分页问题，其GitHub源码地址：
- <https://github.com/pagehelper/Mybatis-PageHelper>



■ PageHelper使用方法

- 引入PageHelper的jar包



- 在MyBatis的XML配置文件中配置拦截器插件
- 调用PageHelper类的方法实现分页功能



■ 在MyBatis的XML配置文件中配置拦截器插件

- MyBatis允许使用插件来拦截方法的调用，如update、query、commit、rollback等方法
- 实现插件时，只需定义类实现MyBatis的Interceptor接口，并指定想要拦截的方法名即可

```
<plugins>
  <!-- com.github.pagehelper为PageHelper类所在包名 -->
  <plugin interceptor="com.github.pagehelper.PageInterceptor">
  </plugin>
</plugins>
```



- 调用PageHelper类的方法实现分页功能

```
PageHelper.startPage(2,3);  
List<User> users = userMapper.findAllUsers();
```

- 调用PageHelper的startPage静态方法之后，跟在后面的第一个**查询方法**将进行分页查询



- 分页查询时，查询方法返回的List对象实际上是Page<E>类型，如果需要取得分页信息，需要将结果强制类型转换为Page<E>类型

```
Page<User> userPages = (Page<User>)users;  
System.out.println(userPages.getTotal()); //总条数  
System.out.println(userPages.getPageNum()); //当前页数  
System.out.println(userPages.getPageSize()); //每页条数  
System.out.println(userPages.getPages()); //总页数
```



■ MyBatis配置优化

- 别名配置、数据源配置

■ MyBatis分页插件

- 引入PageHelper的jar包
- 在MyBatis的XML配置文件中配置拦截器插件
- 调用PageHelper类的方法实现分页功能



THANK YOU
