

- 1、properties:引入外部资源文件，引入数据源
- 2、typeAliases: 可以定义别名
- 3、typeHandlers:把jdbc数据类型转换成java类型 Mybatis通过TypeHandlerRegistry自动转换类型
- 4、objectFactory:实例化返回值对象类
- 5、plugins:配置Mybatis插件： 分页插件

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration
PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-config.dtd">
<!-- mybatis全局配置文件：加载mybatis基础支持信息，
和spring进行整合被废弃 -->
<configuration>
    <!--其它配置，指定数据源 数据源文件见附件-->
    <properties resource="jdbc.properties"></properties>

    <!--对数据类型配置别名
    type: 要配置的数据类型
    alias: 配置后的别名 -->
    <typeAliases>
        <!--设置单个别名-->
        <typeAlias type="cn.rong.zx.Users" alias="users"></typeAlias>
        <!--批量设置别名 扫描包下所有类-->
        <package name="cn.rong.zx"/>
    </typeAliases>
    <environments default="development">
    <environment id="development">
    <!--jdbc事务管理，mybatis负责执行管理 -->
    <transactionManager type="JDBC"/>
    <!-- jdbc数据源，mybatis负责管理 -->
    <dataSource type="POOLED">
```

```

    <!--设置数据源-->
    <property name="driver" value="${jdbc.driver}"/>
    <property name="url" value="${jdbc.url}"/>
    <property name="username" value="${jdbc.username}"/>
    <property name="password" value="${jdbc.password}"/>
</dataSource>
</environment>
</environments>
<mappers>
    <mapper resource="cn.rong.liu/UserDao.xml"/>
</mappers>
</configuration>

```

6、mappers 配置开发模式

普通模式 与 普通接口代理开发模式：配置相同 都是读取配置资源 `<mapper resource="mapper/User.xml"/>`

接口代理开发要求：方式一

- 1、映射文件namespace名称必须是接口全类路径名
- 2、接口方法名称必须和映射文件sql语句Id一致
- 3、映射文件接受参数类型必须和接口传递参数类型一致
- 4、映射文件返回值类型必须和接口返回值类型一致

class接口代理开发模式： 扫描单个接口 `<mapper class="com.itheima.dao.UserDao"/>`

package接口代理开发模式： 扫描包下的所有接口 `<package name="com.itheima.dao"/>`

接口代理开发要求：方式二 和 方式三

- 1、映射文件namespace名称必须是接口全类路径名
- 2、接口方法名称必须和映射文件sql语句Id一致
- 3、映射文件接受参数类型必须和接口传递参数类型一致
- 4、映射文件返回值类型必须和接口返回值类型一致
- 5、映射文件必须和接口在同一个目录，且名称相同

