

JavaWeb 开发路线

- JSP
- JSP + SERVLET
- SPRING
- SPRING MVC
- SPRINGBOOT
- SPRING CLOUD

项目管理

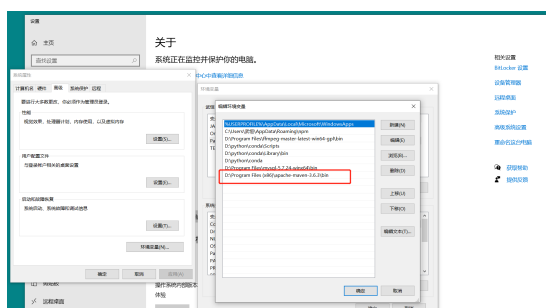
- 所有的高级语言, 都有项目管理工具
- Maven Java , npm Javascript , compose PHP, pip Pythen, golang go, pop ios

MAVEN 安装方法

- 下载地址 <https://archive.apache.org/dist/maven/maven-3/>
- 我们下载 3.6.3版本, 然后解压得到一个 apache-maven-3.6.3 文件夹
- 配置 maven bin 目录 到 PATH 环境变量内

MAVEN 使用方法

- pom.xml 是 maven 依赖的配置档, 需要放在项目的根目录
- Maven 坐标系 groupId artifactId version 组成
- mvn compile 编译当前项目 并产生临时目录 target
- mvn clean 清理编译产生的 临时文件
- mvn test 执行测试代码用例
- mvn package 把当前项目达成 指定的 软件包
- mvn install 把当前项目 发布到 本地的仓库内
- 第一种使用方法 命令行下使用 mvn, 需要配置环境变量
 - mvn 依赖于 JAVA 所以, 系统里必须配置了 JAVA_HOME 环境变量
 - mvn --version 验证是否安装成功



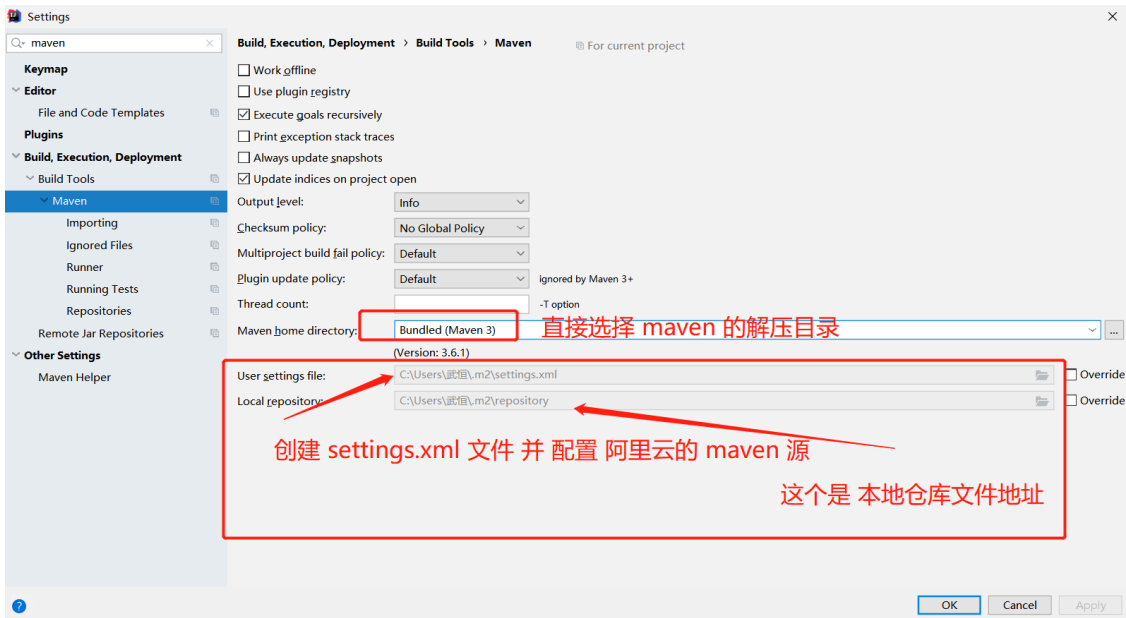
- mvn 仓库配置, 在当前用户目录下 建一个 .m2 文件夹,
- 在 .m2 文件夹内 建一个 settings.xml 文件
- 在 文件内 粘贴 如下 内容

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
5     http://maven.apache.org/xsd/settings-1.0.0.xsd">
6   <mirrors>
7     <mirror>
8       <id>alimaven</id>
9       <name>aliyun maven</name>
10      <url>http://maven.aliyun.com/nexus/content/groups/public/</url>
11      <mirrorOf>central</mirrorOf>
12    </mirror>
13  </mirrors>
14 </settings>

```

● 第二种使用方法 配置开发环境一起使用



Mybatis 的使用

- 第一步 在 maven 依赖中 添加 Mybatis 的依赖

```

1 <dependency>
2   <groupId>org.mybatis</groupId>
3   <artifactId>mybatis</artifactId>

```

```
4 <version>3.5.5</version>
5 </dependency>
```

- 第二步 在 resource 文件夹下面创建一个 xml 文件 名称为 mybatis-config.xml
- 第三步 在 mybatis-config.xml 添加数据库的连接信息

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!DOCTYPE configuration
3 PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
4 "http://mybatis.org/dtd/mybatis-3-config.dtd">
5 <configuration>
6 <environments default="default">
7 <environment id="default">
8 <transactionManager type="JDBC"></transactionManager>
9 <dataSource type="POOLED">
10 <property name="url" value="jdbc:mysql://127.0.0.1:3306/vip21?
11 useSSL=false"/>
12 <property name="username" value="root"/>
13 <property name="password" value="123456"/>
14 <property name="driver" value="com.mysql.cj.jdbc.Driver"/>
15 </dataSource>
16 </environment>
17 </environments>
18 <mappers>
19 <mapper resource="UserMapper.xml" />
20 </mappers>
21 </configuration>
```

- 第四步 在 resource 文件夹里 创建 UserMapper.xml 文件 并写入 SQL 语句

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
3 "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
4 <mapper namespace="com.lovecoding.test.dao.UserMapper">
5 <select id="selectUser" resultType="com.lovecoding.test.pojo.User" >
6 SELECT * FROM `tb_user`;
7 </select>
8 <select id="selectUserByone" resultType="com.lovecoding.test.pojo.User" >
9 SELECT * FROM `tb_user` Limit 1;
10 </select>
11 <select id="selectUserById" parameterType="int"
12 resultType="com.lovecoding.test.pojo.User">
```

```
11     SELECT * FROM `tb_user` WHERE id = #{uid} Limit 1;
12 </select>
13 <select id="selectUserByIdandName" resultType="com.lovecoding.test.pojo.User">
14     SELECT * FROM `tb_user` WHERE id = #{id} AND username = #{name} Limit 1;
15 </select>
16 </mapper>
```

- 第六步 创建 mybatis 实例 并调用sql

```
1 @Test
2 public void t0() throws IOException {
3     //我们定义了 mybatis 配置文件路径
4     String res = "mybatis-config.xml";
5     //我们用 Resources 去把配置文件转化为 数据流
6     InputStream resourceAsStream = Resources.getResourceAsStream(res);
7     //我们用 mybatis SqlSessionFactoryBuilder 工厂类 去创建 SqlSessionFactory
8     SqlSessionFactory build = new SqlSessionFactoryBuilder().build(resourceAsStream);
9     //我们用 SqlSessionFactory 创建 SqlSession
10    SqlSession sqlSession = build.openSession();
11    /**
12     * 直接调用mappersql方式
13     */
14    //查询一个列表
15    List<User> User =
16    sqlSession.selectList("com.lovecoding.test.dao.UserMapper.selectUser");
17 }
```