

# Idea开发工具

---

## Idea

### 一、卸载

### 二、官网：<https://www.jetbrains.com/idea/>

### 三、打不开官网，解决方案

### 四、简介

#### 1. JetBrains公司

#### 2. 优势（比较Eclipse）

### 五、下载安装

#### 1. 下载地址：<https://www.jetbrains.com/idea/download/#section=windows>

### 六、安装

### 七、修改IDEA配置

### 八、启动激活

### 九、创建Java工程

#### 第一步：创建工程

#### 第二步：选择自己的JDK

#### 第三步：选择模板

#### 第四步：选择工程保存位置

#### 第五步：创建包和Class

#### 第六步：运行

### 十、Module创建

### 十一、模块的删除

### 十二、项目配置

### 十三、Idea常用配置

#### 1. 配置修改

#### 2. 外观和行为

#### 3. Editor-General

### 十四、关于模块

#### 1. 常用的模块

## 十五、创建web工程

第一步：创建Java工程同时勾选web模块

第二步：指定项目名称

第三步：修改启动页面

第四步：配置Tomcat

## 十六、创建Maven工程

1. 创建Java工程

2. 创建Web工程

## 十七、使用框架（MyBatis）

第一步：导入jar

第二步：创建数据源与实体类

第三步：创建核心配置文件与映射文件

## 十八、整合数据库

第一步：添加数据库连接

第二步：设置用户名密码，测试连接

## 十九、Maven默认设置

## 廿十、Idea三大框架使用

1. MyBatis框架

2. Spring框架

3. SpringMVC框架

# Idea

## 一、卸载

控制面板卸载

选择电脑中自带软件管家卸载

卸载完成后检查本地安装目录，

配置C:\Users\Administrator

## 二、官网：<https://www.jetbrains.com/idea/>

## 三、打不开官网，解决方案

找到**hosts**文件，修改内容

文件位置：C:\Windows\System32\drivers\etc

```
16 #       102.54.94.97       rhino.acme.com       # source server
17 #       38.25.63.10       x.acme.com         # x client host
18
19 # localhost name resolution is handled within DNS itself.
20 #   127.0.0.1       localhost
21 #   ::1             localhost
22
23 #127.0.0.1 ieonline.microsoft.com
24 #0.0.0.0 account.jetbrains.com
25 #0.0.0.0 www.jetbrains.com
26
```

Hosts文件结尾添加

**#0.0.0.0 account.jetbrains.com**

**#0.0.0.0 www.jetbrains.com**

如果以上方式解决不了问题，使用VPN

## 四、简介

### 1.JetBrains公司

IDEA是JetBrains公司产品，其他产品

IDEs	
AppCode	PyCharm
CLion	Rider
Datalore ↗	RubyMine
DataGrip	WebStorm
GoLand	
IntelliJ IDEA	
PhpStorm	

**AndroidStudio**:用于开发Android（Google公司基于IDEA**社区版**进行迭代开发）

### 2.优势（比较Eclipse）

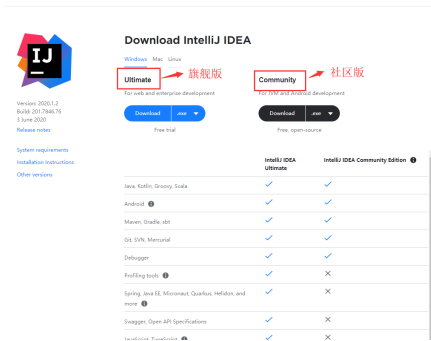
- 1.提示功能范围广：绑定数据库，写SQL语句时，都有提示
- 2.提示功能全面：JS，HTML...
- 3.代码模板很好用：psvm 生成main函数

**注意**：面试时如果遇到这个问题，并不是说的越多越好...

## 五、下载安装

1. 下 载 地 址 :

<https://www.jetbrains.com/idea/download/#section=windows>

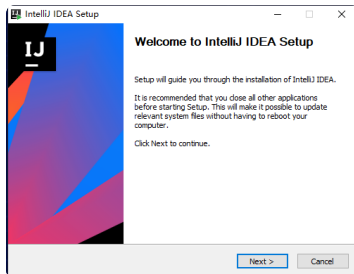


IDEA分两个版本：旗舰版，社区版  
旗舰版收费（30天免费使用）  
社区版（完全免费）

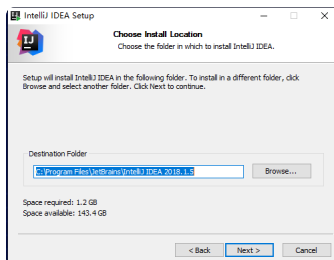
下载旗舰版

## 六、安装

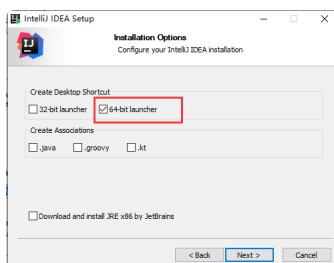
### 1.双击运行安装包



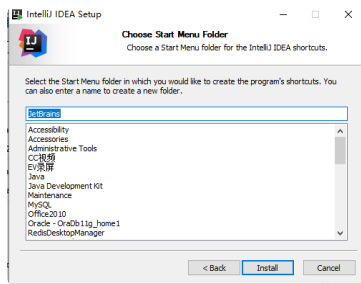
### 2.配置安装目录



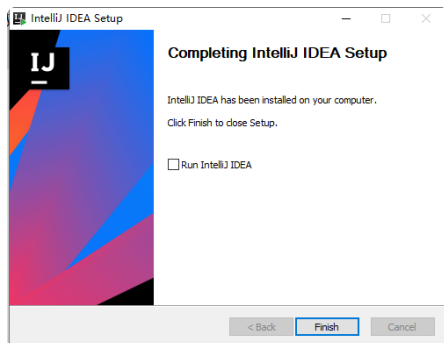
### 3.添加桌面启动图标



### 4.安装



## 5.安装完成



安装硬件要求：

内存：2G **8G以上 CPU i5**

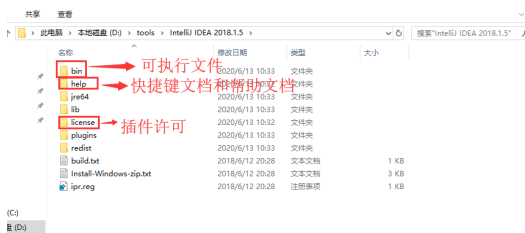
硬盘：1.5 **固态硬盘**

屏幕：1024\*768

实际开发中，IDEA执行时会由大量的缓存、索引文件

建议指望Idea解决卡、慢这类问题，基本上不可能

安装目录：



## 七、修改IDEA配置

以自身电脑配置做参考

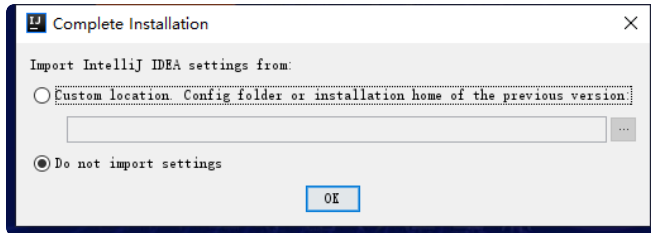
如果配置（64位，16内存）

修改配置

```
1 -Xms512m
2 -Xmx1500m
3 -XX:ReservedCodeCacheSize=500m
4 -XX:+UseConcMarkSweepGC
5 -XX:SoftRefLRUPolicyMSPerMB=50
6 -ea
7 -Dsun.io.useCanonCaches=false
8 -Djava.net.preferIPv4Stack=true
9 -XX:+HeapDumpOnOutOfMemoryError
10 -XX:-OmitStackTraceInFastThrow
11
```

## 八、启动激活

第一步：启动选择不导入设置



第二步：激活

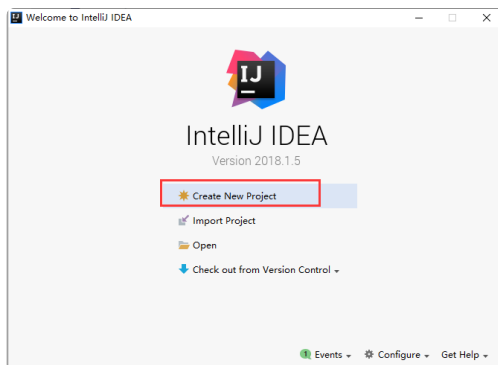
<https://www.cnblogs.com/jin-zhe/p/9267912.html>

建议有条件的同学使用正版

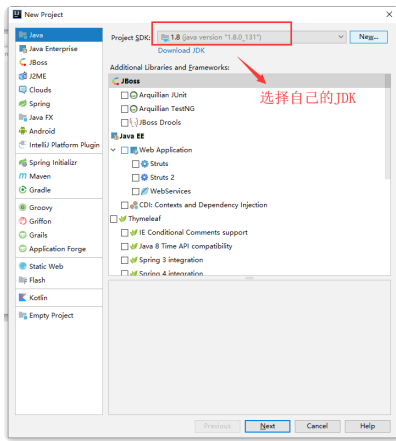
以上仅供学习使用，不可商用！！！！

## 九、创建Java工程

第一步：创建工程



第二步：选择自己的JDK

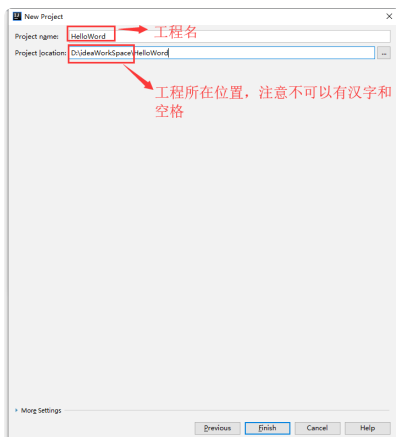


### 第三步：选择模板

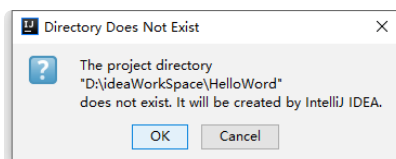


直接next

### 第四步：选择工程保存位置

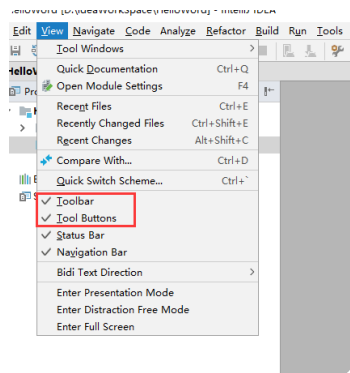


Finish后弹出新建工程提示：

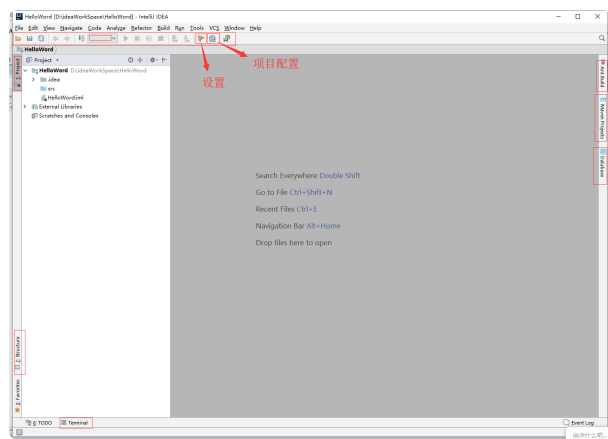


Ok就ok了

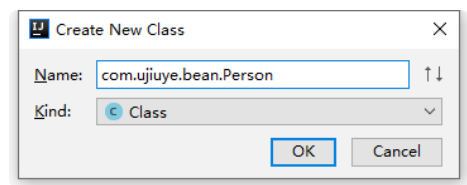
建议勾选视图：



窗口界面：



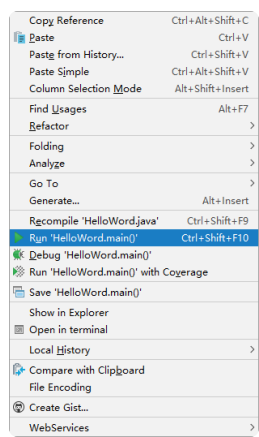
第五步：创建包和Class



包和类可以分开创建，也可以如上图一步到位

第六步：运行

可以右键运行：



也可以绿色按钮运行：



```

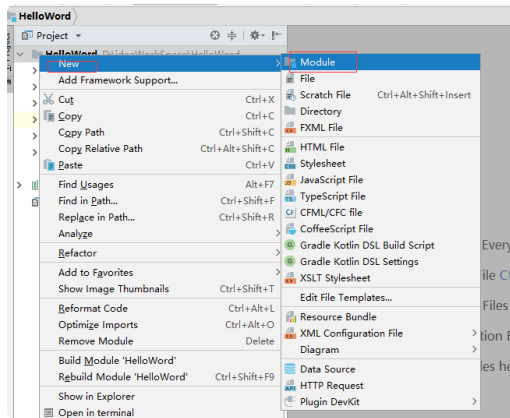
1 package com.ujiuye.test;
2
3 public class HelloWorld {
4     public static void main(String[] args) {
5         System.out.println("HelloWord");
6     }
7 }
8

```

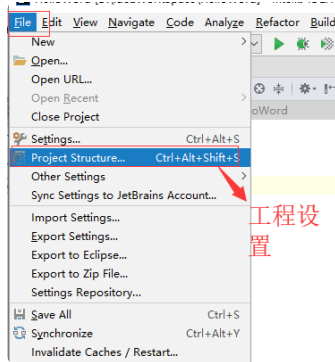
注意：写完代码之后不需要手动保存，代码在Idea中是自动保存的

## 十、Module创建

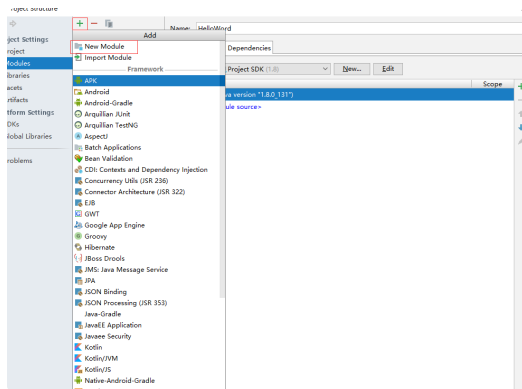
### 1.选中工程，右键创建



### 2.工程配置中添加Module

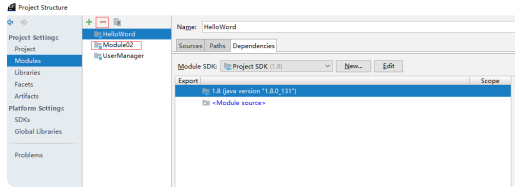


快捷键：选中工程，F4

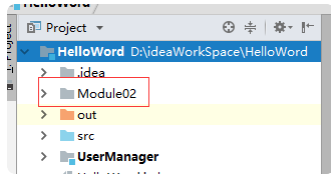


## 十一、模块的删除

默认情况下，Idea工具做了安全设置，避免误删Module，没有提供直接Delete方式  
删除操作需要先打开项目设置：

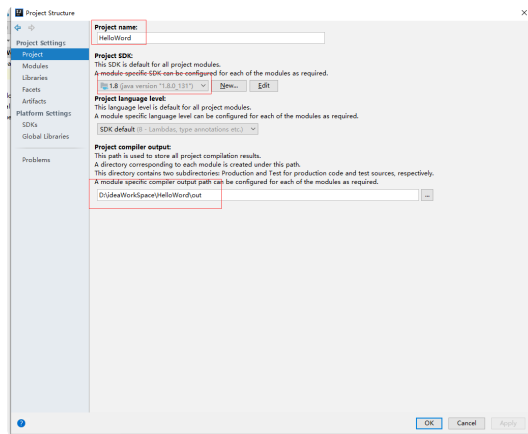


选中需要删除的模块，“\_”号删除



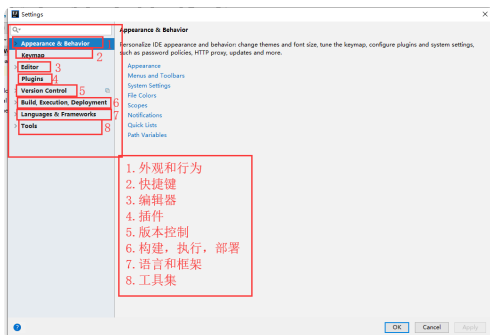
删除后，模块置灰，此时可以右键删除本地文件

## 十二、项目配置



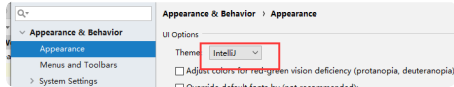
### 十三、Idea常用配置

## 1. 配置修改

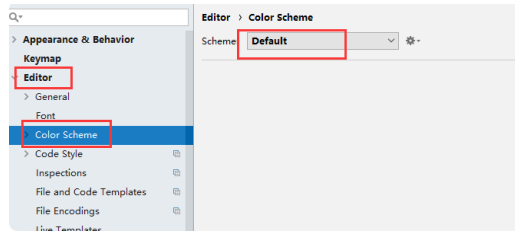


## 2. 外观和行为

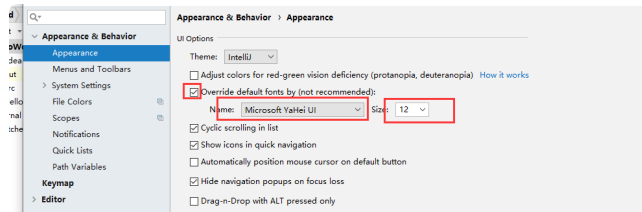
### 2.1 设置主题



编辑区也可修改主题：

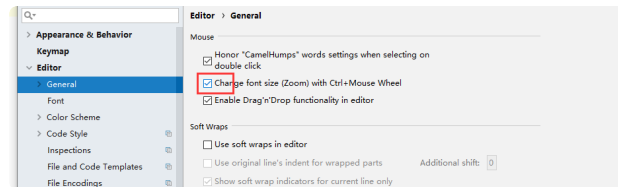


### 2.2 设置窗体以及菜单字体大小

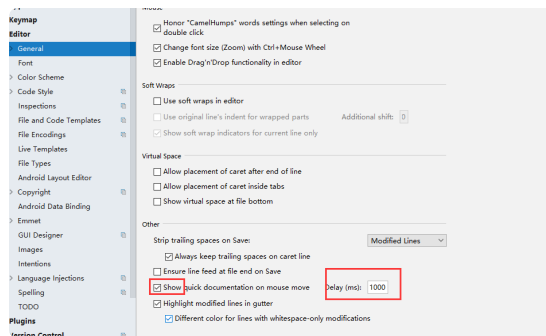


## 3. Editor-General

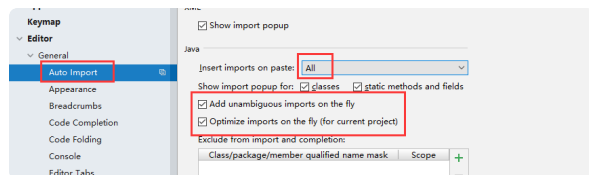
### 3.1 设置鼠标滚动修改字体大小



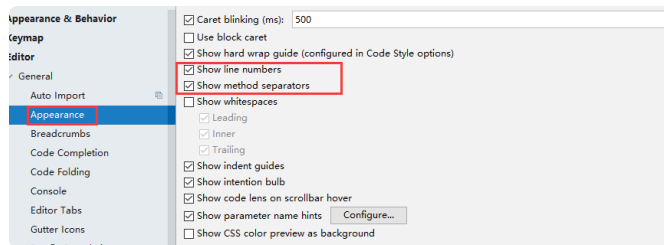
### 3.2 设置鼠标悬浮提示



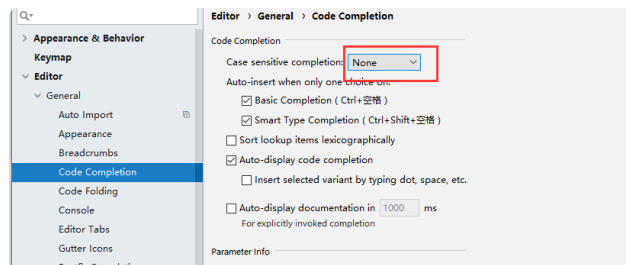
### 3.3 设置自动导包



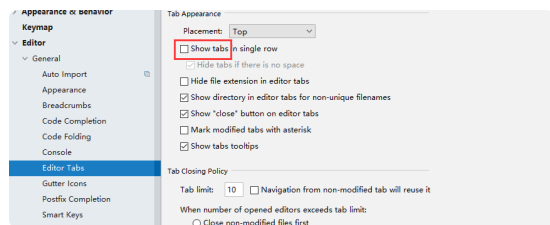
### 3.4 设置显示行号和方法间的分隔符



### 3.5设置忽略大小写提示

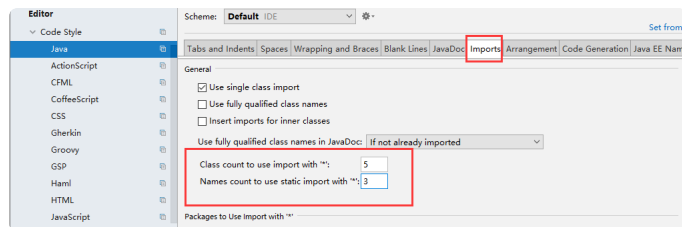


### 3.6设置取消单行显示tabs的操作

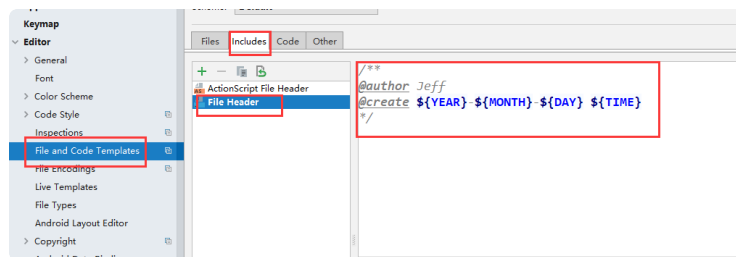


取消勾选，取消选项卡的单行显示

### 3.7设置超过指定import个数，显示\*



### 3.8修改类头的文档注释信息



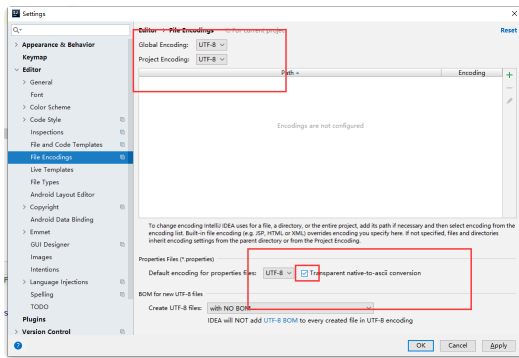
```
/**
```

```
@author LiHao
```

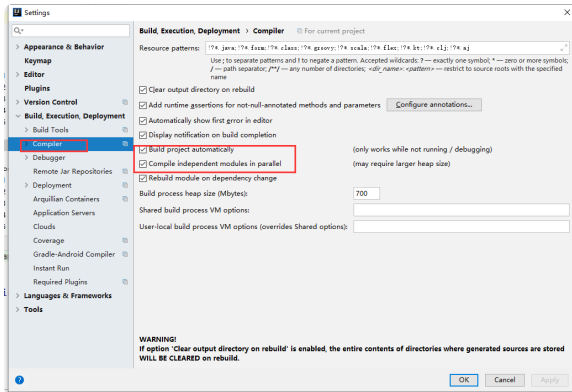
```
@create ${YEAR}-${MONTH}-${DAY} ${TIME}
```

```
*/
```

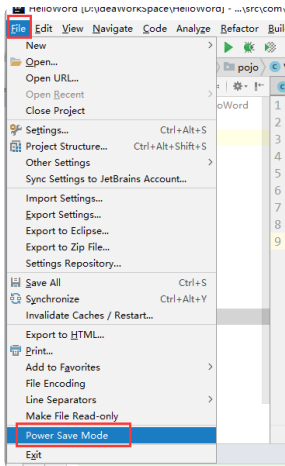
### 3.9设置编码



### 3.10设置自动编译

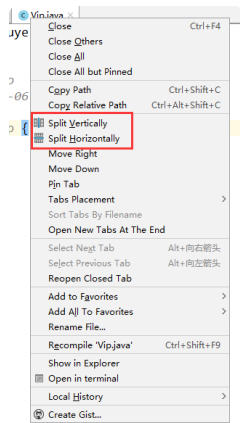


### 3.11省电模式



省电模式一旦开启，Idea不会检查代码，提示等功能  
又称为阅读模式

### 3.12代码水平或垂直显示



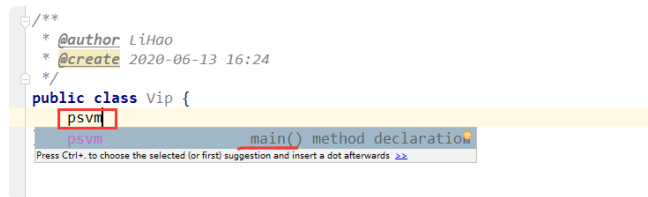
## 十四、关于模块

原理：配置一些常用的代码字母缩写，在输入简写时可以出现预定义的固定模式代码

Eclipse中：syso

### 1. 常用的模块

#### 1.1main函数：psvm



#### 1.2打印输出：sout

```
//sout
System.out.println();
//soutp: 打印方法参数
System.out.println("args = [" + args + "]");
//soutm: 打印方法名
System.out.println("Vip.main");
int i = 10;
int ii = 20;
//soutv: 打印变量
System.out.println("ii = " + ii);
//xxx.sout: 打印指定变量xxx
System.out.println(i);
```

#### 1.3fori:for循环

```
//fori: 生成for循环语句块
String [] arr = new String[]{"Tom","Jerry","LiHao"};
ArrayList arrayList = new ArrayList();
//iter: 增强FOR 循环
for (String s : arr) {
    System.out.println(s);
}
//iter: 遍历数据
for (int j = 0; j < arr.length; j++) {
    String s = arr[j];
}
//List.for: 遍历指定集合List
for (Object o : arrayList) {
}
//List.fori: 遍历指定集合List
for (int i1 = 0; i1 < arrayList.size(); i1++) {
}
//List.fori: 遍历指定集合List(倒叙)
for (int i1 = arrayList.size() - 1; i1 >= 0; i1--) {
}
}
```

## 1.4ifn: 条件判断

```
//inn: 非空判断
if (arrayList != null) {

}
//ifn: 空值判断
if (arrayList == null) {

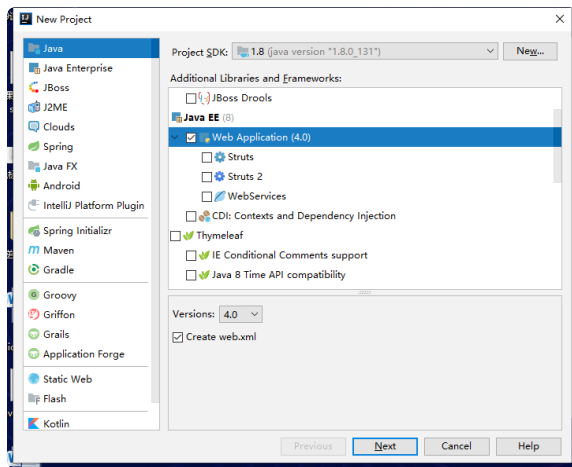
}
//xx.nn: 对指定集合做非空判断
if (arrayList != null) {

}
//xxx.null: 对指定集合做空值判断
if (arrayList == null) {

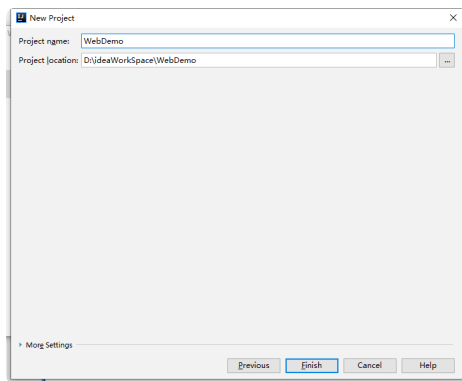
}
```

# 十五、创建web工程

## 第一步：创建Java工程同时勾选web模块



## 第二步：指定项目名称



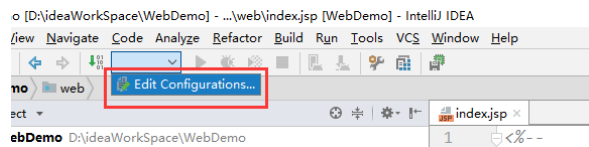
## 第三步：修改启动页面

```
1  <!--
2  Created by IntelliJ IDEA.
3  User: Administrator
4  Date: 2020/6/15
5  Time: 9:47
6  To change this template use File | Settings | File Templates.
7  -->
8  <%@ page contentType="text/html; charset=UTF-8" language="java" %>
9  <html>
10 <head>
11 <title>中公优就业官网</title>
12 </head>
13 <body>
14 <h1 style="color: burlywood"></h1>
15 </body>
16 </html>
17
```

Idea提示功能很强大

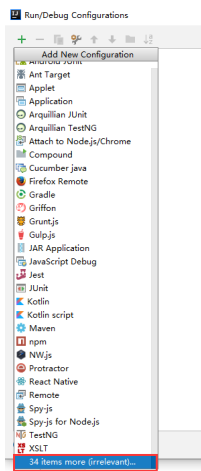
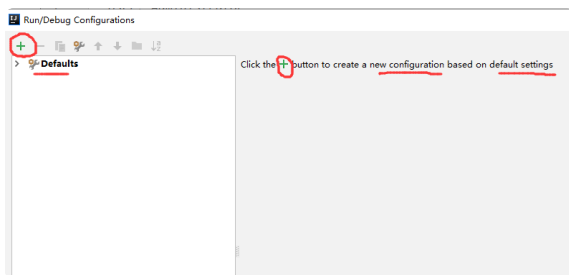
## 第四步：配置Tomcat

### 1. 配置位置：

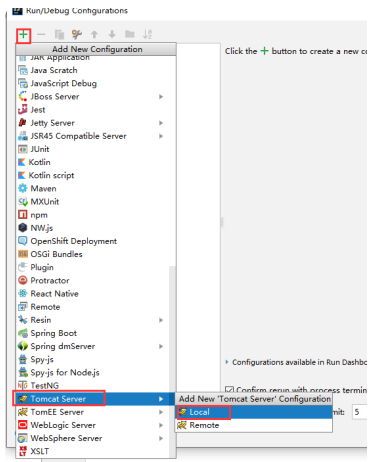


Run---Edit Configurations

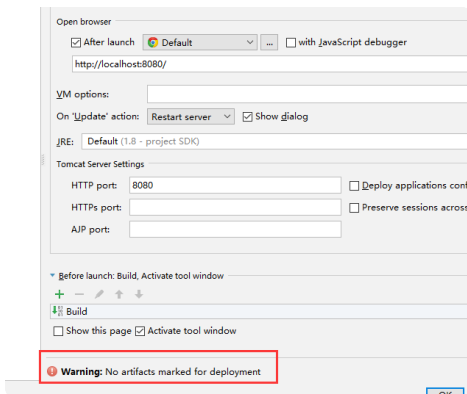
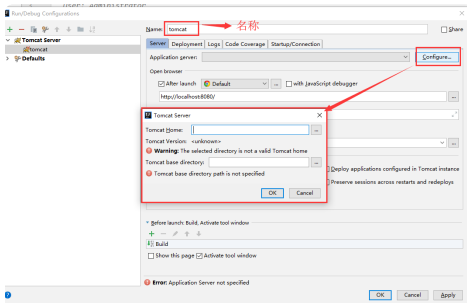
### 2. 点击+添加tomcat配置



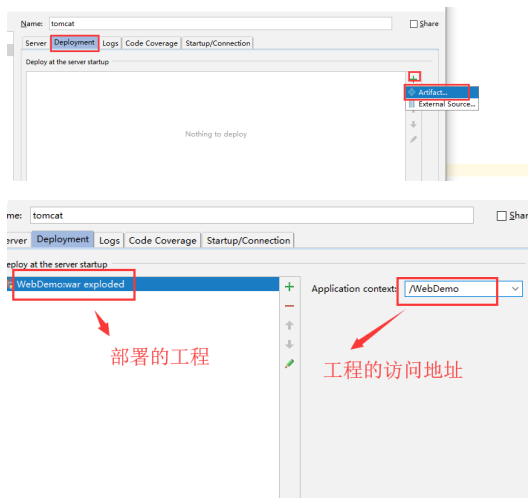


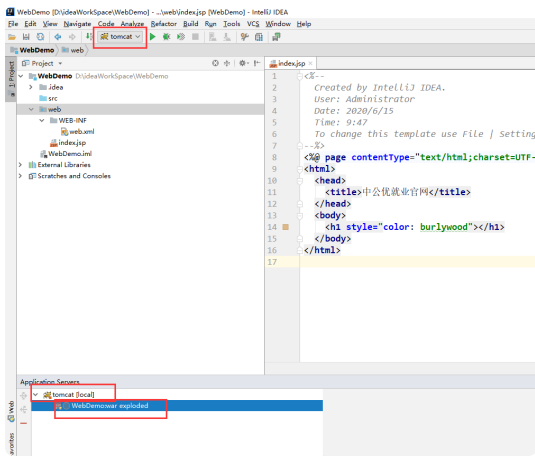


### 3. Tomcat基本配置

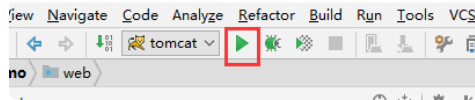


部署项目到当前的服务器上：



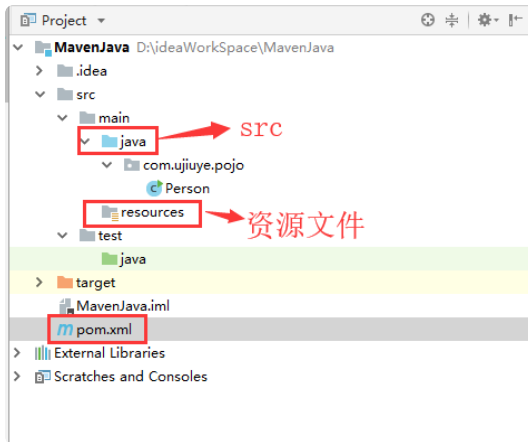


运行Web工程:



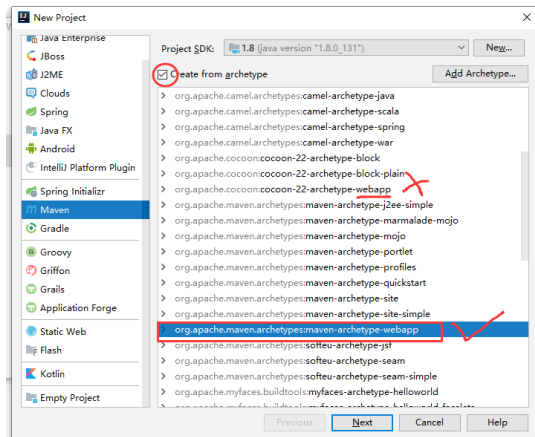
## 十六、创建Maven工程

### 1. 创建Java工程

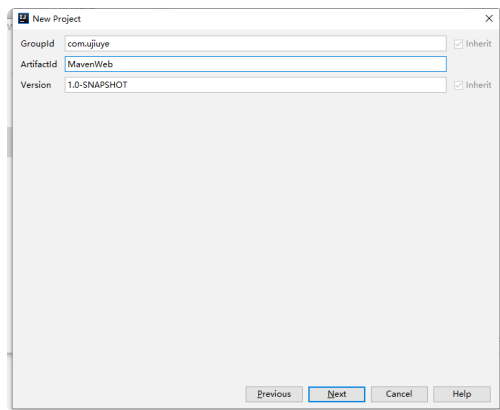


### 2. 创建Web工程

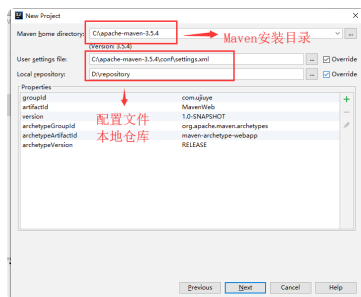
第一步：选择模板



## 第二步：起名

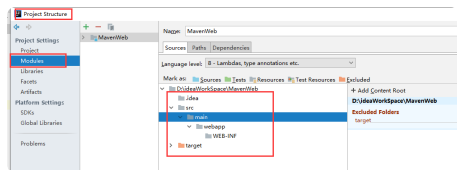


## 第三步：配置本地的Maven

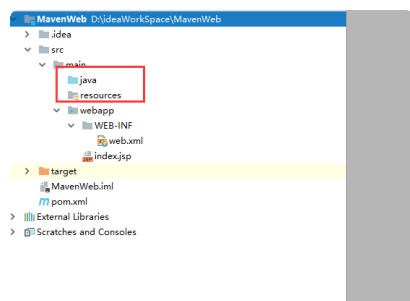
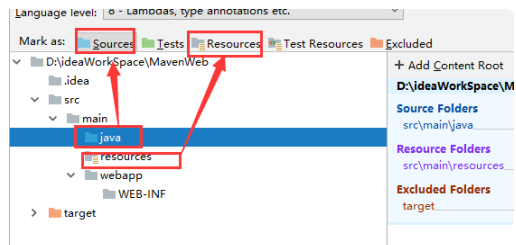


一路next创建完成，自动导包即可

## 第四步：修改项目结构，添加编写Java代码的模块



## 添加java和resources目录

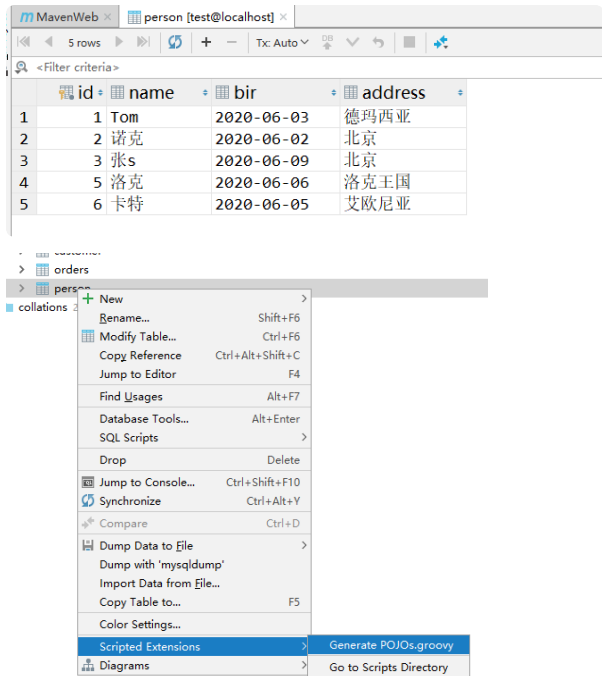


## 十七、使用框架（MyBatis）

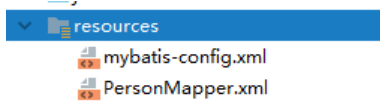
# 第一步：导入jar

```
<dependencies>
<!-- MyBatis包 -->
<dependency>
<groupId>org.mybatis</groupId>
<artifactId>mybatis</artifactId>
<version>3.5.1</version>
</dependency>
<!-- MySQL连接 -->
<dependency>
<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
<version>5.1.47</version>
</dependency>
<!-- junit测试 -->
<dependency>
<groupId>junit</groupId>
<artifactId>junit</artifactId>
<version>4.12</version>
<scope>test</scope>
</dependency>
<dependency>
<groupId>org.junit.jupiter</groupId>
<artifactId>junit-jupiter-api</artifactId>
<version>RELEASE</version>
<scope>compile</scope>
</dependency>
</dependencies>
```

# 第二步：创建数据源与实体类

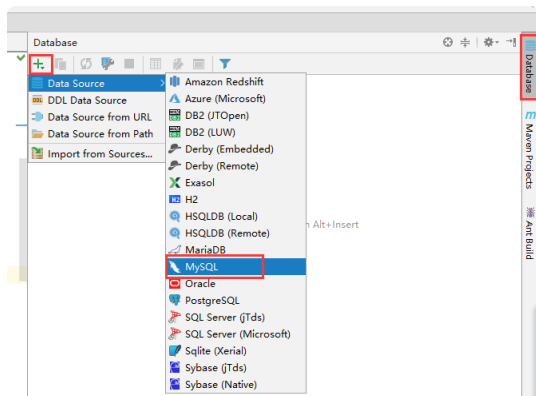


# 第三步：创建核心配置文件与映射文件

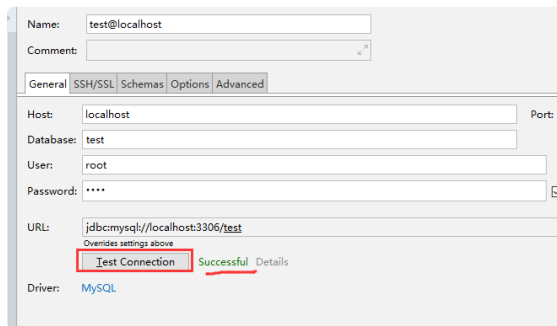
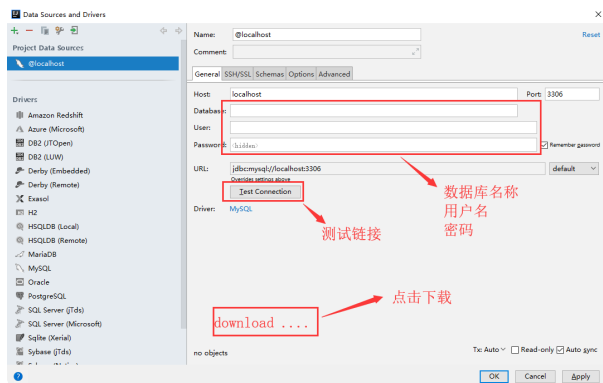


# 十八、整合数据库

## 第一步：添加数据库连接

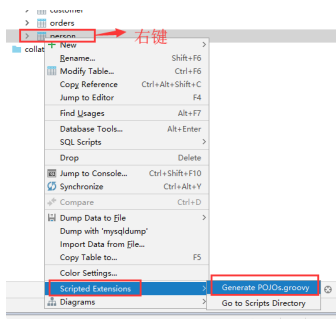


## 第二步：设置用户名密码，测试连接

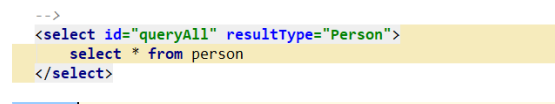


## 连接数据库常用功能（开发过程）

1. 使用Mybatis\Hibernate等ORM框架时，创建pojo对象

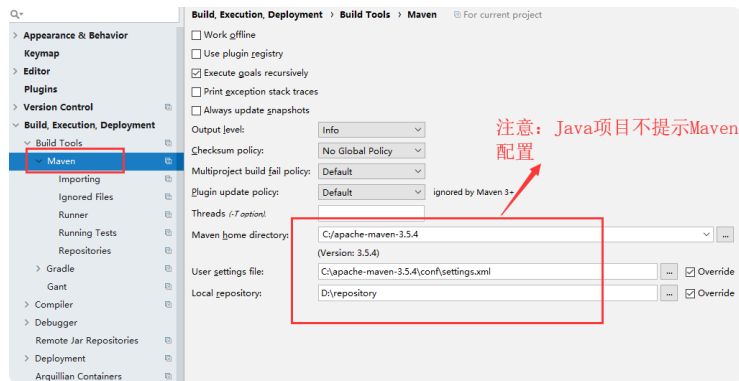


2. 在使用MyBatis时，映射文件中的SQL语句可以提示（提示范围广）



## 十九、Maven默认设置

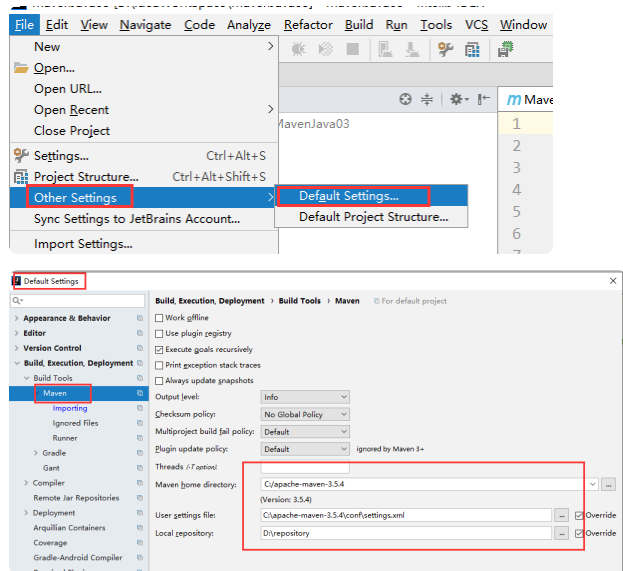
### 1. Java项目Maven配置的修改



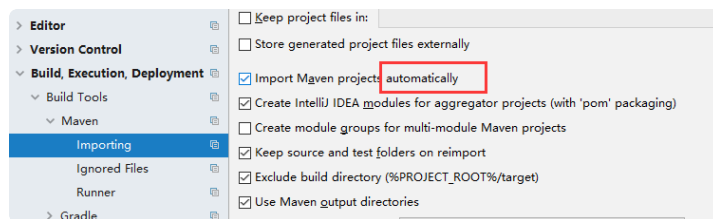
### 2. 修改默认设置

修改完上述内容后，再次创建项目还需要修改上述内容

修改默认设置



### 自动导包



## 廿十、Idea三大框架使用

### 1. MyBatis框架

接口代理模式开发

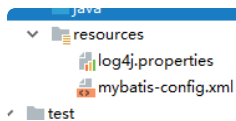
第一步：导入Jar包

```

<dependencies>
  <!-- MyBatis -->
  <dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis</artifactId>
    <version>3.5.1</version>
  </dependency>
  <!-- MySQL 连接 -->
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>5.1.47</version>
  </dependency>
  <!-- junit测试 -->
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.12</version>
    <scope>test</scope>
  </dependency>
  <!-- Log4j 日志文件 -->
  <dependency>
    <groupId>log4j</groupId>
    <artifactId>log4j</artifactId>
    <version>1.2.17</version>
  </dependency>
</dependencies>

```

## 第二步：引入核心配置文件及日志文件



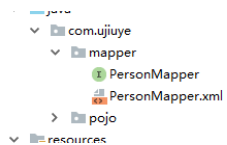
## 第三步：创建实体类

```

//1. 建议实现序列化接口
public class Person implements Serializable{
    //2. 基本数据类型属性建议使用包装类
    private Integer id;
    private String name;
    private Date bir;
    private String address;
    public Integer getId() {

```

## 第四步：定义接口与映射文件



## 第五步：测试

```

DEBUG [main] - PooledDataSource forcefully closed/removed all connections.
org.apache.ibatis.binding.BindingException: Invalid bound statement (not found): com.ujioye.mapper.PersonMapper.queryById
    at org.apache.ibatis.binding.MapperMethodSqlCommand.<init>("MapperMethod.java:235)
    at org.apache.ibatis.binding.MapperMethod.<init>("MapperMethod.java:233)
    at org.apache.ibatis.binding.MapperProxy.<init>("MapperProxy.java:60) <i internal call>
    at org.apache.ibatis.binding.MapperProxy.<init>("MapperProxy.java:60) <i internal call>
    at org.apache.ibatis.binding.MapperProxy.<init>("MapperProxy.java:60) <i internal call>
    at org.apache.ibatis.binding.MapperProxy.<init>("MapperProxy.java:60) <i internal call>
    at com.ujioye.test.MyTest.testQueryById(MyTest.java:36) <i internal call>
    at com.ujioye.test.MyTest.testQueryById(MyTest.java:36) <i internal call>

```

以上错误原因：

Idea编译xml文件时，只编译resources目录下的内容

映射文件在Idea编译工程时，并未被打包，所以报找不到方法

解决方案：在pom.xml中添加如下编译设置

```

<version>1.2.17</version>
</dependency>

</dependencies>

<build>
  <resources>
    <resource>
      <directory>src/main/java</directory>
      <includes>
        <include>/**/*.xml</include>
      </includes>
    </resource>
  </resources>
</build>
</project>

```

<build>

<resources>

```

<resource>
  <directory>src/main/java</directory>
  <includes>
    <include>**/*.xml</include>
  </includes>
</resource>
</resources>
</build>

```

## 2. Spring框架

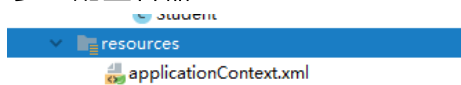
### 第一步：导包

```

<dependencies>
  <!-- Spring 相关依赖 -->
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>5.1.5.RELEASE</version>
  </dependency>
</dependencies>

```

### 第二步：配置容器



### 第三步：定义类型

```

/**
 * @author 买牌
 * @create 2020-06-15 15:20
 */
public class Student {
    private String name;

    public Student() {
    }

    public Student(String name) {
        this.name = name;
    }

    public void study() {
        System.out.println("正在学习Java");
    }
}

```

### 第四步：测试代码

```

/**
 * @author 买牌
 * @create 2020-06-15 15:23
 */
public class MyTest {
    @Test
    public void testSpring() {
        // 创建容器
        ApplicationContext applicationContext = new ClassPathXmlApplicationContext("applicationContext.xml");
        // 获取容器中的对象
        Student student = (Student) applicationContext.getBean("@student1");
        student.study();
    }
}

```

## 3. SpringMVC框架

### 第一步：创建Web工程，导包

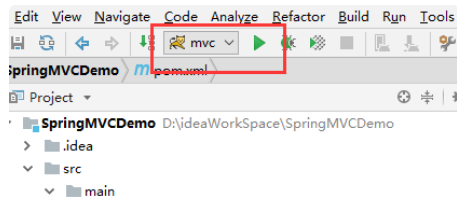


```

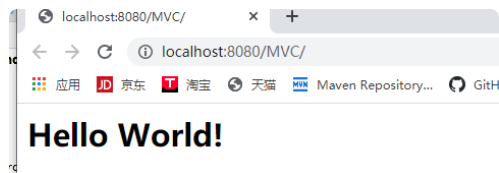
<!-- spring-webmvc -->
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-webmvc</artifactId>
<version>5.1.5.RELEASE</version>
</dependency>
<dependency>
<groupId>junit</groupId>
<artifactId>junit</artifactId>
<version>4.11</version>
<scope>test</scope>
</dependency>
</dependencies>

```

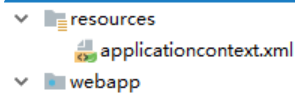
## 第二步：配置服务器



## 第三步：测试运行



## 第四步：添加SpringMVC核心配置文件

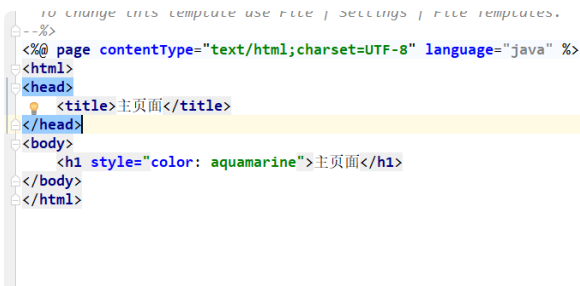


## 第五步：配置前端控制器（web.xml）



## 第六步：添加Controller

### 1.添加跳转页面



### 2.定义Controller类，处理请求

```
/**
 * @author 宋峰
 * @create 2020-06-15 16:12
 */
@Controller
public class HelloController {
    @RequestMapping("hello")
    public String jump(){
        return "main";
    }
}
```