# maven安装包conf的setting中新增配置：

<profile>

<id>jdk-1.8</id>

<activation>

<activeByDefault>true</activeByDefault>

<jdk>1.8</jdk>

</activation>

<properties>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>1.8</maven.compiler.target>

<maven.compiler.compilerVersion>1.8</maven.compiler.compilerVersion>

</properties>

</profile>

# 2.IDEA中maven选自己的仓库和配置路径

# 3.IDEA创建springboot项目

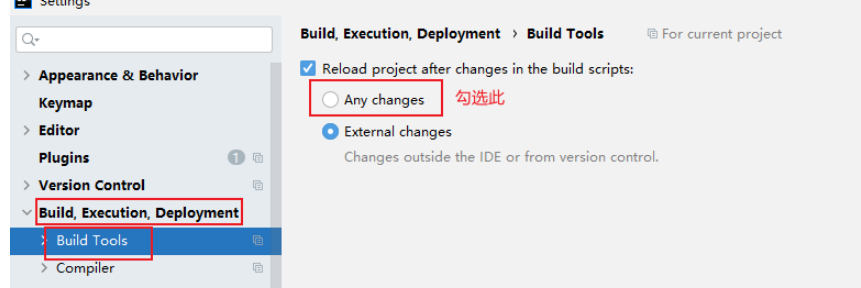
1.选new project

2.选 maven

3.maven设置自动配置后下载

老版本的idea--->importings-->auto import

新版本的idea-->Build Tools 勾选 Any changes



1. 去官网查看需要配置的依赖

<parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>2.3.12.RELEASE</version>  
</parent>  
<dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
</dependencies>

# 5、在com.pangus包下新创建一个启动java类com.pangus.MainApplication

添加注解：@SpringBootApplication

**public** **static** **void** **main**(String[] args) {

SpringApplication.run(MainApplication.class, args);

}

# 6.编写测试类HelloWorld

@Controller

public class HelloWorld {

@RequestMapping("/hello")

@ResponseBody

public String hello(){

return "Hello World!";

}

}

# 7.导入一个插件到BOM，简化部署(需要加版本号不然会报错)

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

<version>2.3.12.RELEASE</version>

</plugin>

</plugins>

</build>

# 8.idea右边maven工具package打包后，在window中执行java -jar xxxx即可运行