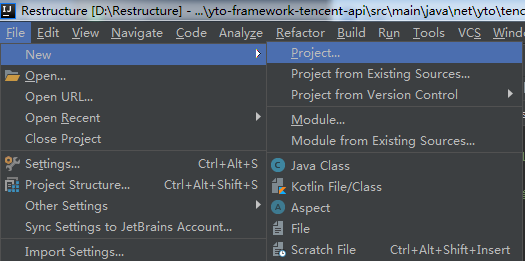
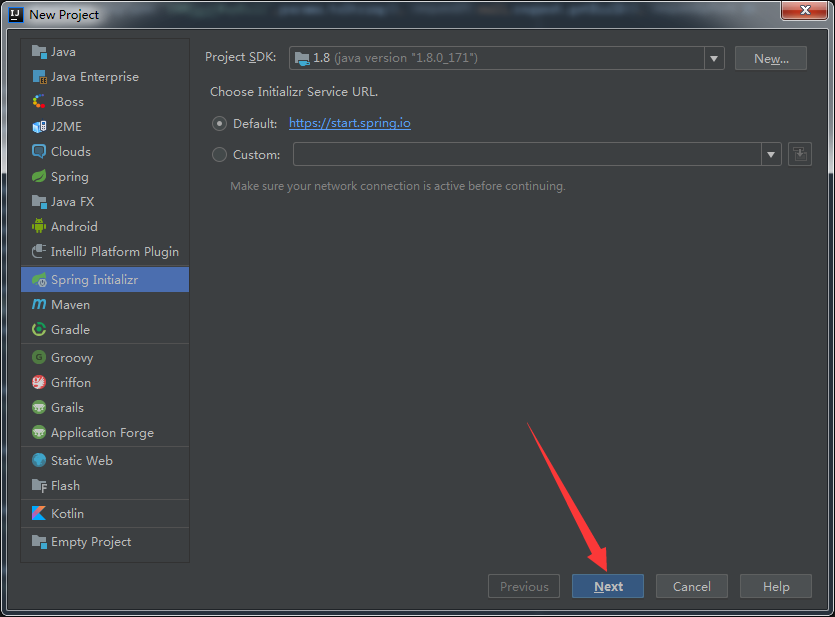
他

# 搭建一个简单的springboot项目

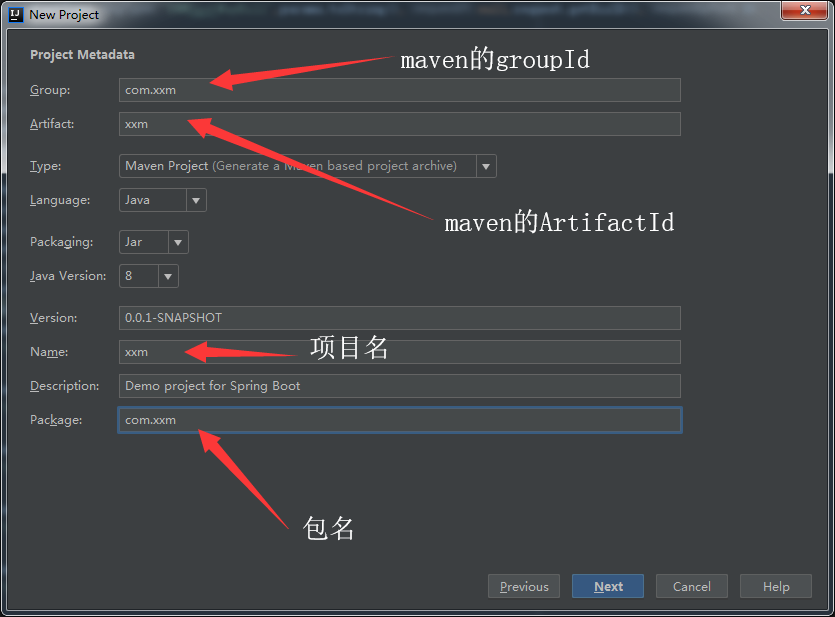
## 创建项目

懒得从头，直接在打开项目的情况下新建一个springboot，按图下一步就行了





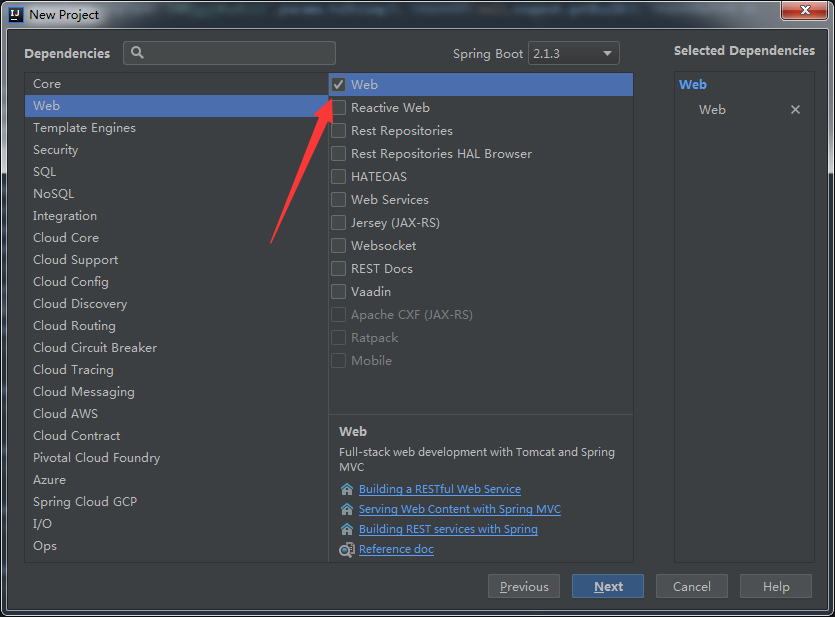
Maven的groupId和artifactId在pom文件中引用会用到、项目名和包名自由填写

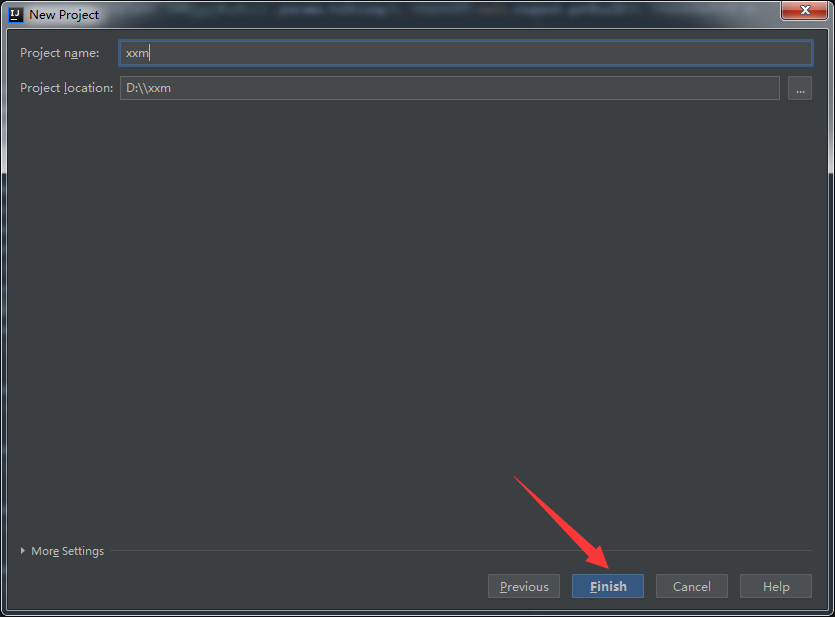


插一嘴，用这个pom.xml的配置解释一下这两个东西的用处，groupId和artitactId自己随便写，格式正确就行，格式不对idea会提示你的，一般artitactId和项目名对应，groupId是你的当前组织的包

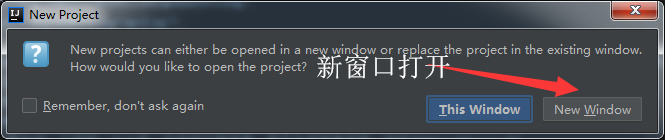


无脑下一步，项目名这就别动他了

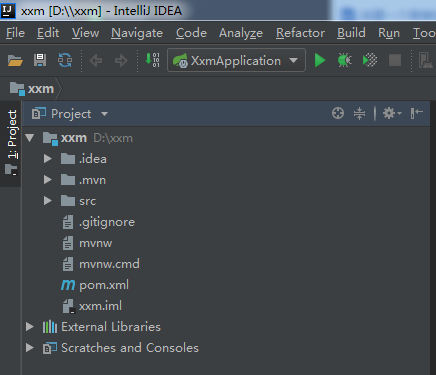


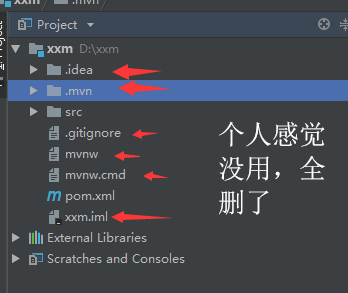


This window会将当前窗口关闭，选择打开一个新的窗口

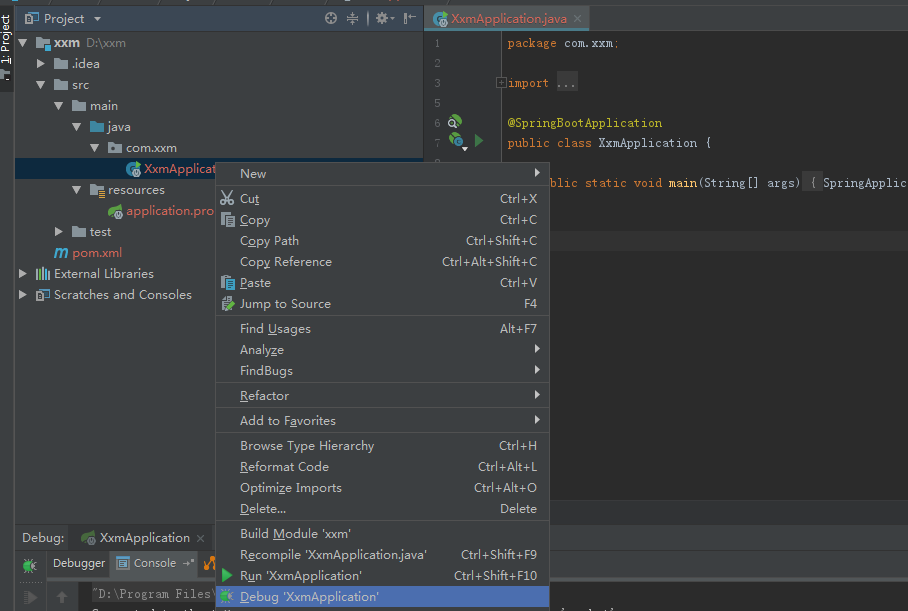


创建项目完成

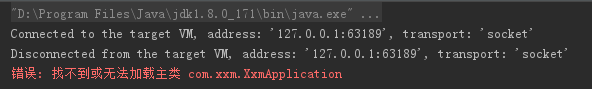




右键项目中的Application,运行

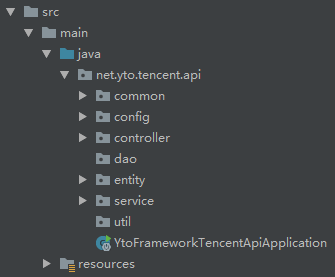


### 启动失败：加载不到主类

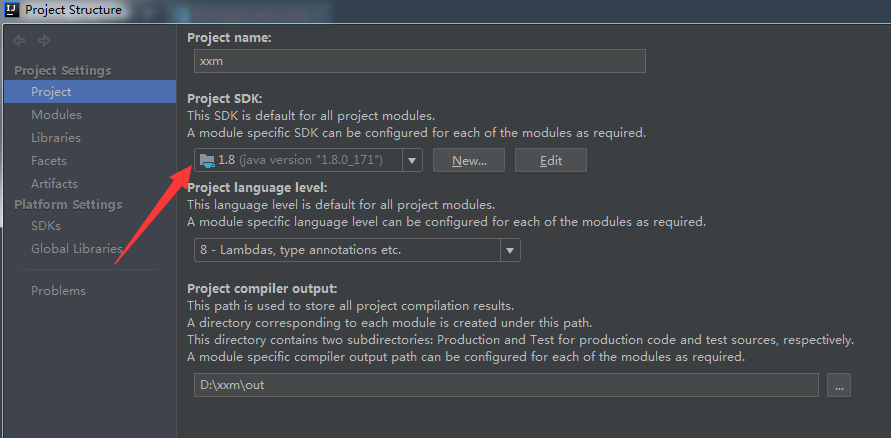
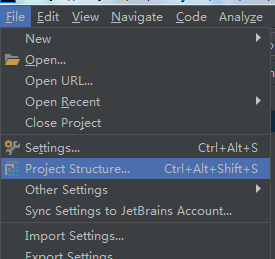


这里加载不到主类的原因有很多种，

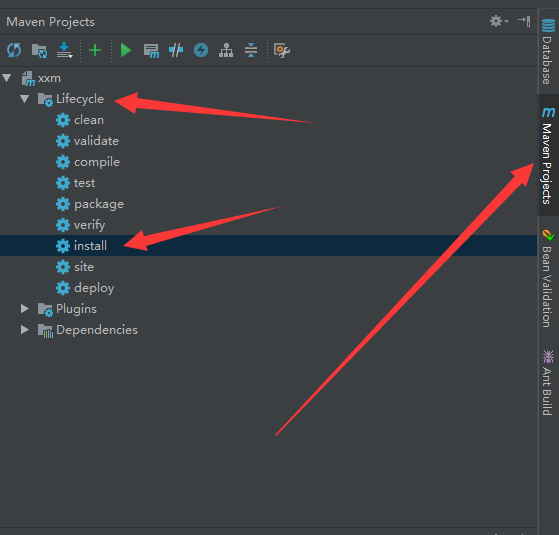
1、Application文件夹不在包的最外层

这种是正确位置，这种错是最常见的

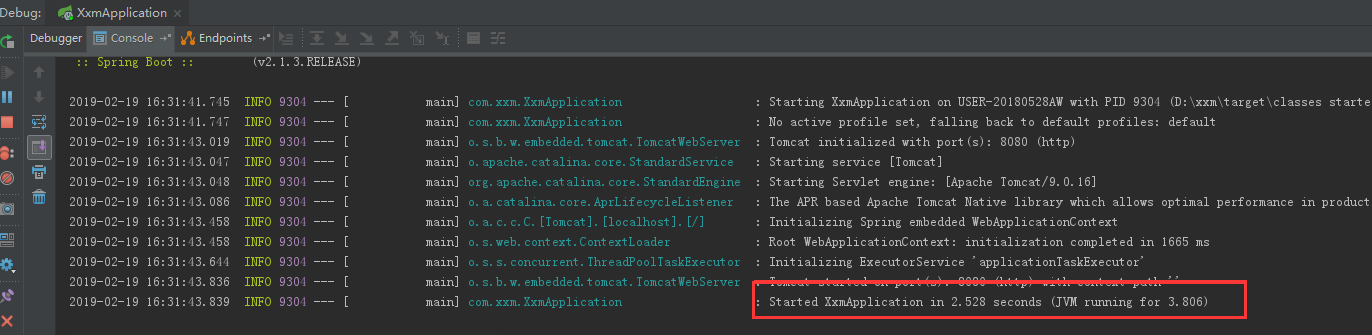
1. 新项目没有配置jdk，检查jdk配置，springboot最低需要jdk1.8的支持，必须是jdk不允许使用jre



1. 项目没有编译，在右侧的maven窗口中install一下即可，右边没有maven的点击菜单栏View->Tool  Windows->Maven projects



运行成功！



### 编写接口看是否成功

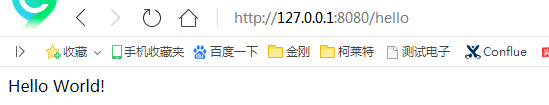
pom.xml添加依赖,代码如下

<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <!-- 项目的依赖关系及自身定义，parent是指父类的信息，引用父类的包，此处引用的是springboot默认依赖，groupId和artifactId在被引用时会用到，命名规范即可 -->  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>2.1.3.RELEASE</version>  
 <relativePath/>  
 </parent>  
 <groupId>com.xxm</groupId>  
 <artifactId>xxm</artifactId>  
 <version>0.0.1</version>  
 <name>xxm</name>  
 <!-- 相当于注释 -->  
 <description>Demo project for Spring Boot</description>  
  
 <!-- 设置项目的jdk版本，编码 -->  
 <properties>  
 <java.version>1.8</java.version>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>  
 </properties>  
  
 <!-- 子项目管理，添加子项目后要在此添加子项目名称，例：<module>xxm-interface</module> -->  
 <modules>  
 </modules>  
  
 <!-- 此处为添加依赖，parent一般只放置一些必要的jar包供子项目依赖使用，子项目根据自身所需引用相应的jar包 -->  
 <dependencies>  
 <!-- spring-boot-starter-web和spring-boot-starter-test为springboot核心包 -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 <!-- lombok 在pojo实体中免get、set方法的编写，在打包的时候会自动添加get、set方法，缺省很多繁琐的操作和代码 -->  
 <dependency>  
 <groupId>org.projectlombok</groupId>  
 <artifactId>lombok</artifactId>  
 <optional>true</optional>  
 </dependency>  
 </dependencies>  
 <!-- 此处一般是私服仓库的配置，如果没有私房仓库，不写 -->  
 <!--<repository>  
 <id>releases</id>  
 <url>http://10.1.193.100:8081/nexus/content/repositories/releases</url>  
 <snapshots>  
 <enabled>false</enabled>  
 </snapshots>  
 </repository>-->  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

创建com.xxm.controller包，创建HelloController，代码如下

package com.xxm.controller;  
  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
*/\*\*  
 \* ClassName: HelloController  
 \* Description: 测试项目是否搭建成功  
 \* Author: wangpengxiang  
 \* data： 2019/2/19 16:34  
 \* Version: 1.0  
 \*\*/*@RestController  
public class HelloController {  
  
 @RequestMapping("/hello")  
 public String Hello(){  
 return "Hello World!";  
 }  
}

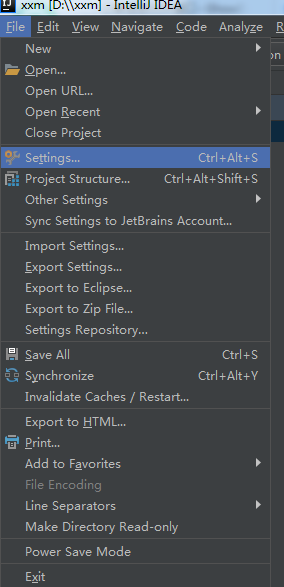
运行项目，浏览器访问127.0.0.1:8080/hello,成功！

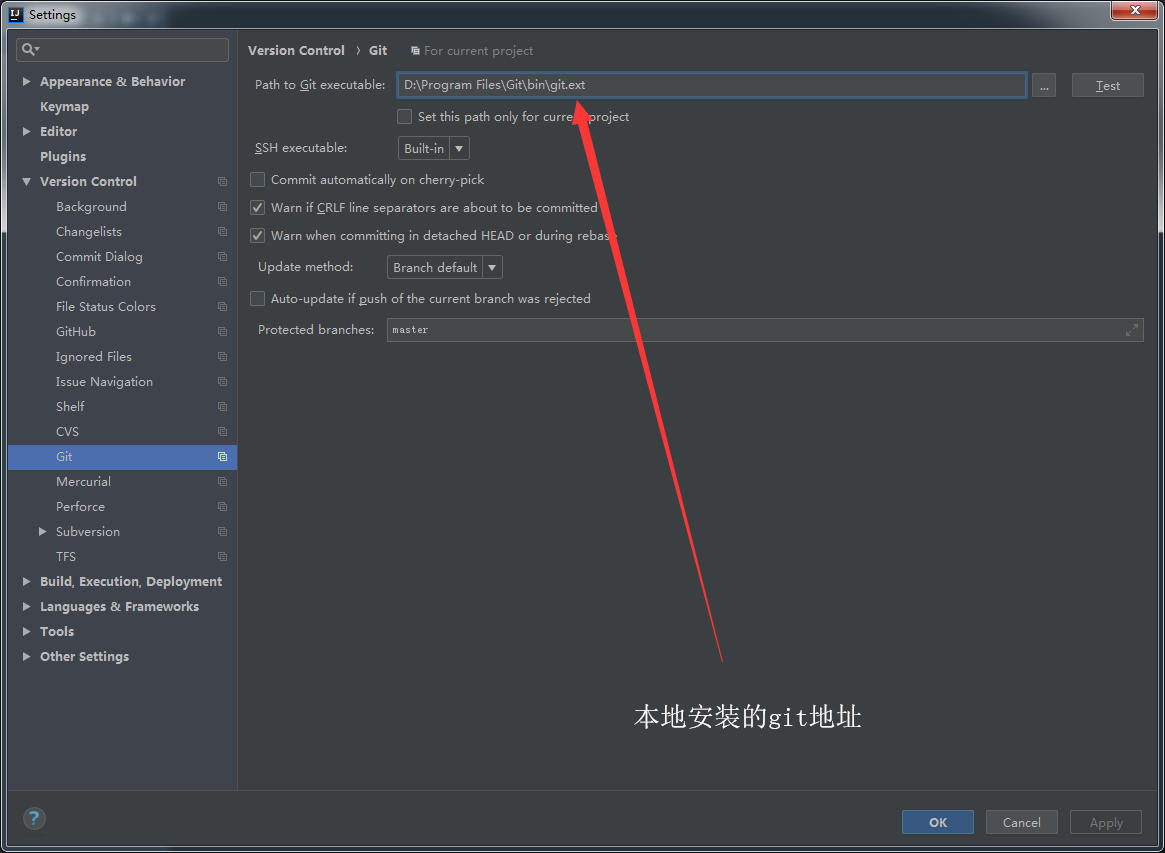


这里踩了一个坑，项目运行没问题，怎么访问都是404，后来重启idea解决了。。。

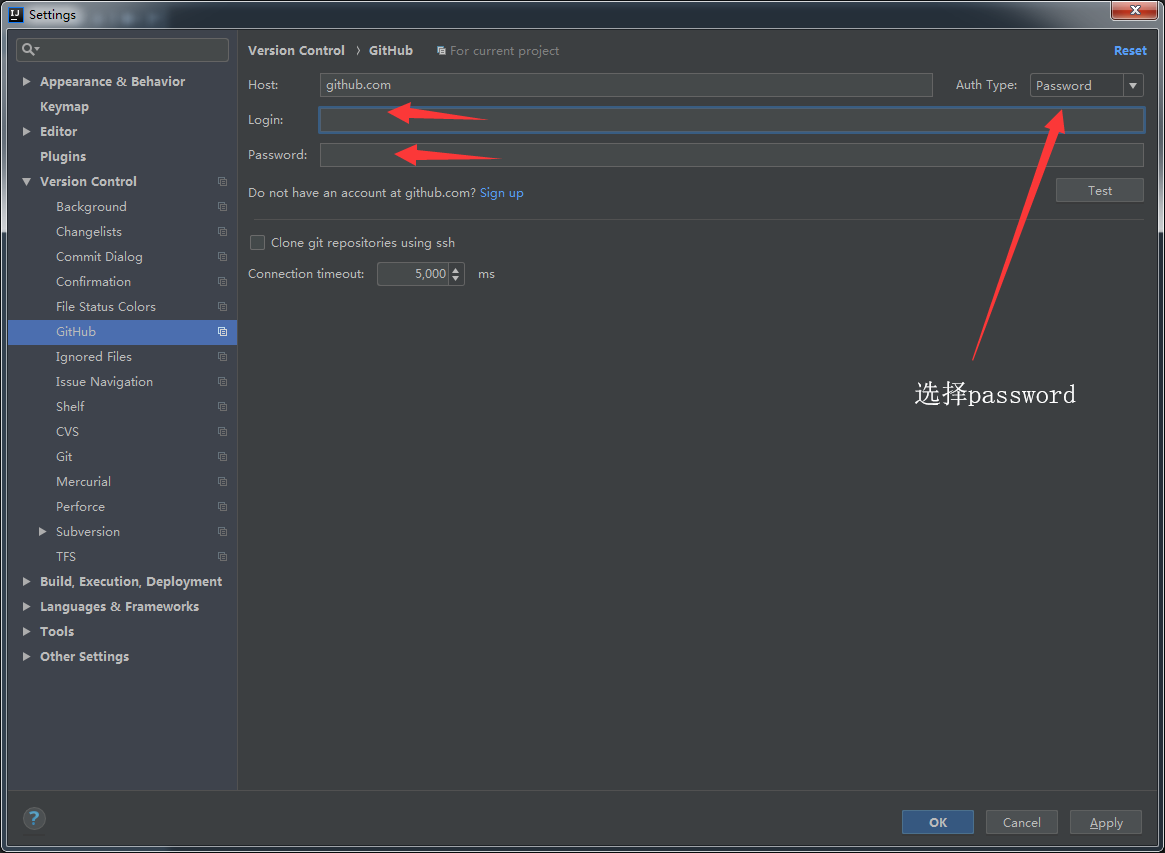
## 2、上传git

防止代码丢失，上传到git管理

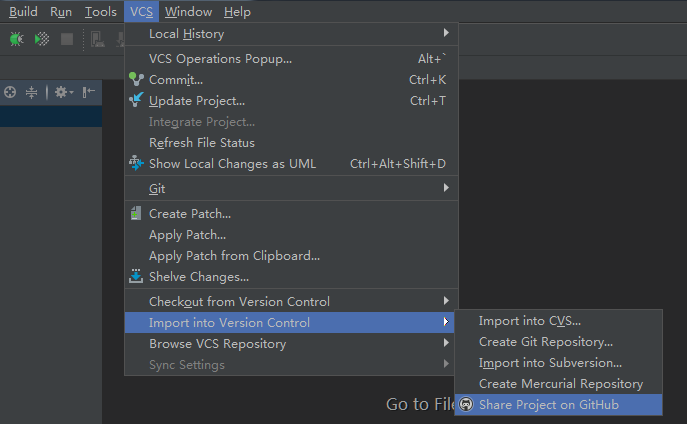




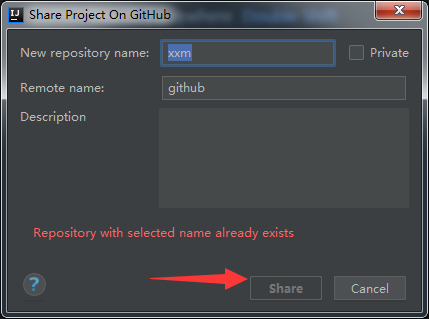
选择password输入帐号密码，最后记得点一下右边的test测试是否成功



将此工程上传到github网站，IDEA会自动帮我们在github网站上创建仓库，仓库名就是你的项目名

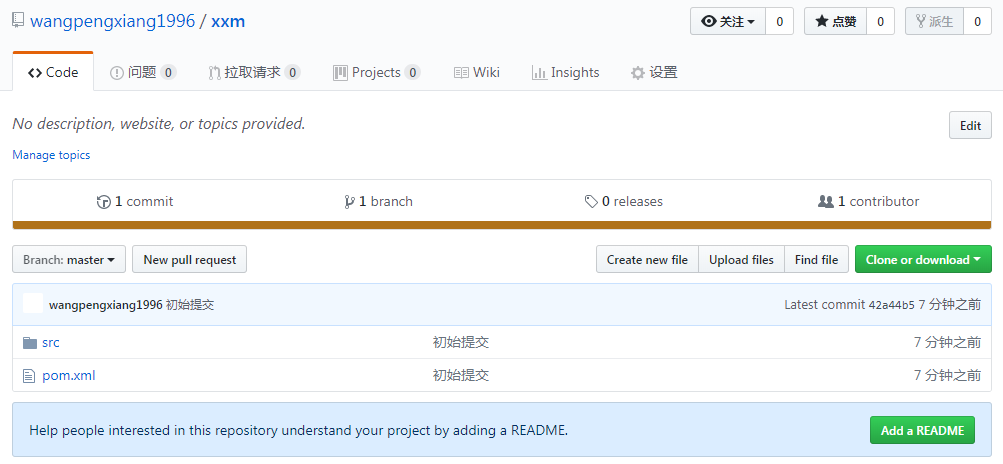


这里红字是因为我刚刚已经创建过一次，说我项目名存在了，这里点share就OK



### 补救操作

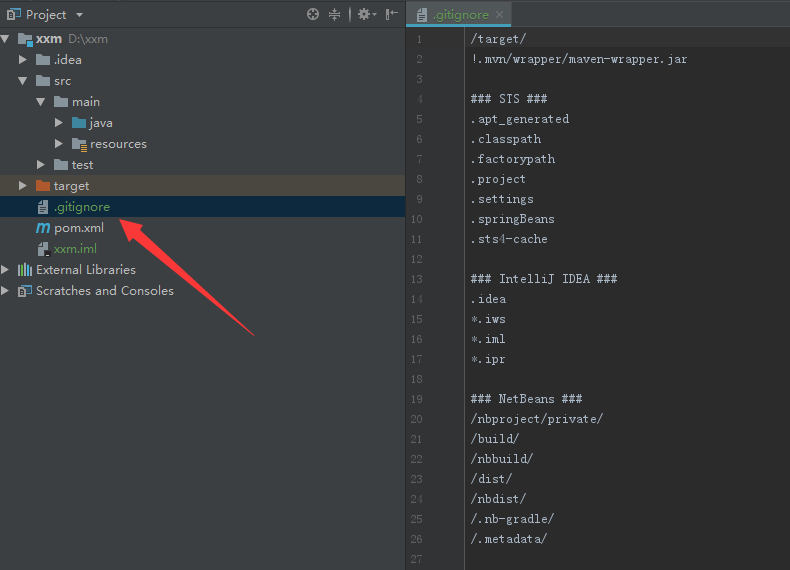
如果在git没有添加到远程仓库，打开项目所在的文件夹，右键git Bash here，输入git init，然后输入git add .；即可正常提交代码



打开git，已经可以看到代码了

### 忽略不想提交的部分

添加ignore，忽略不想提交的代码



代码如下

/target/  
!.mvn/wrapper/maven-wrapper.jar  
  
### STS ###  
.apt\_generated  
.classpath  
.factorypath  
.project  
.settings  
.springBeans  
.sts4-cache  
  
### IntelliJ IDEA ###  
.idea  
\*.iws  
\*.iml  
\*.ipr  
  
### NetBeans ###  
/nbproject/private/  
/build/  
/nbbuild/  
/dist/  
/nbdist/  
/.nb-gradle/  
/.metadata/