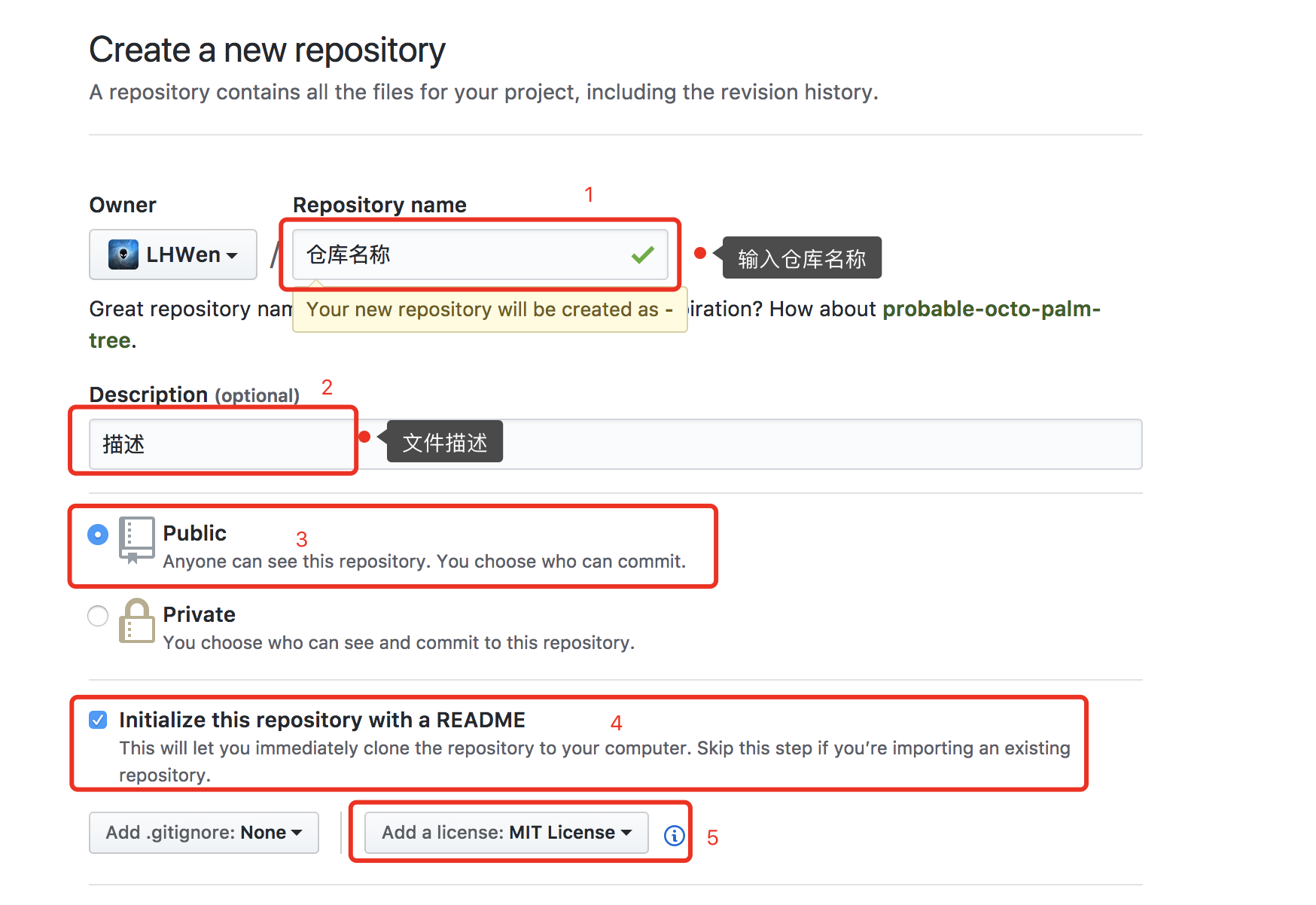
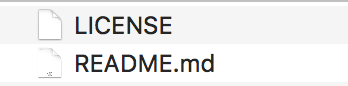
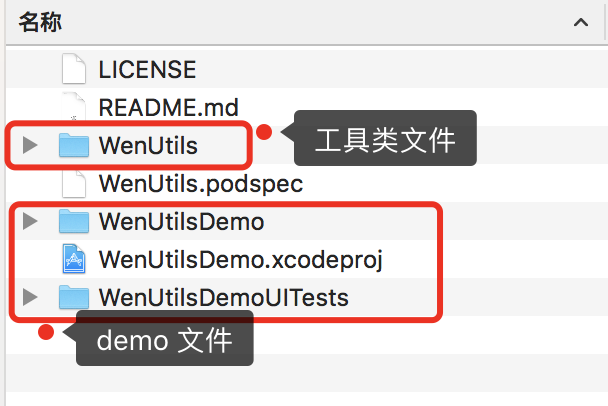
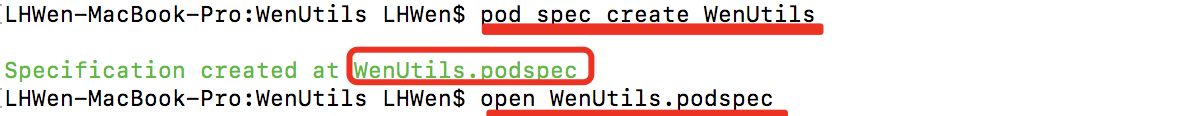
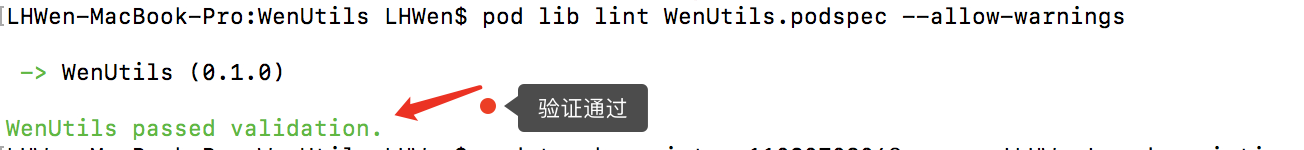
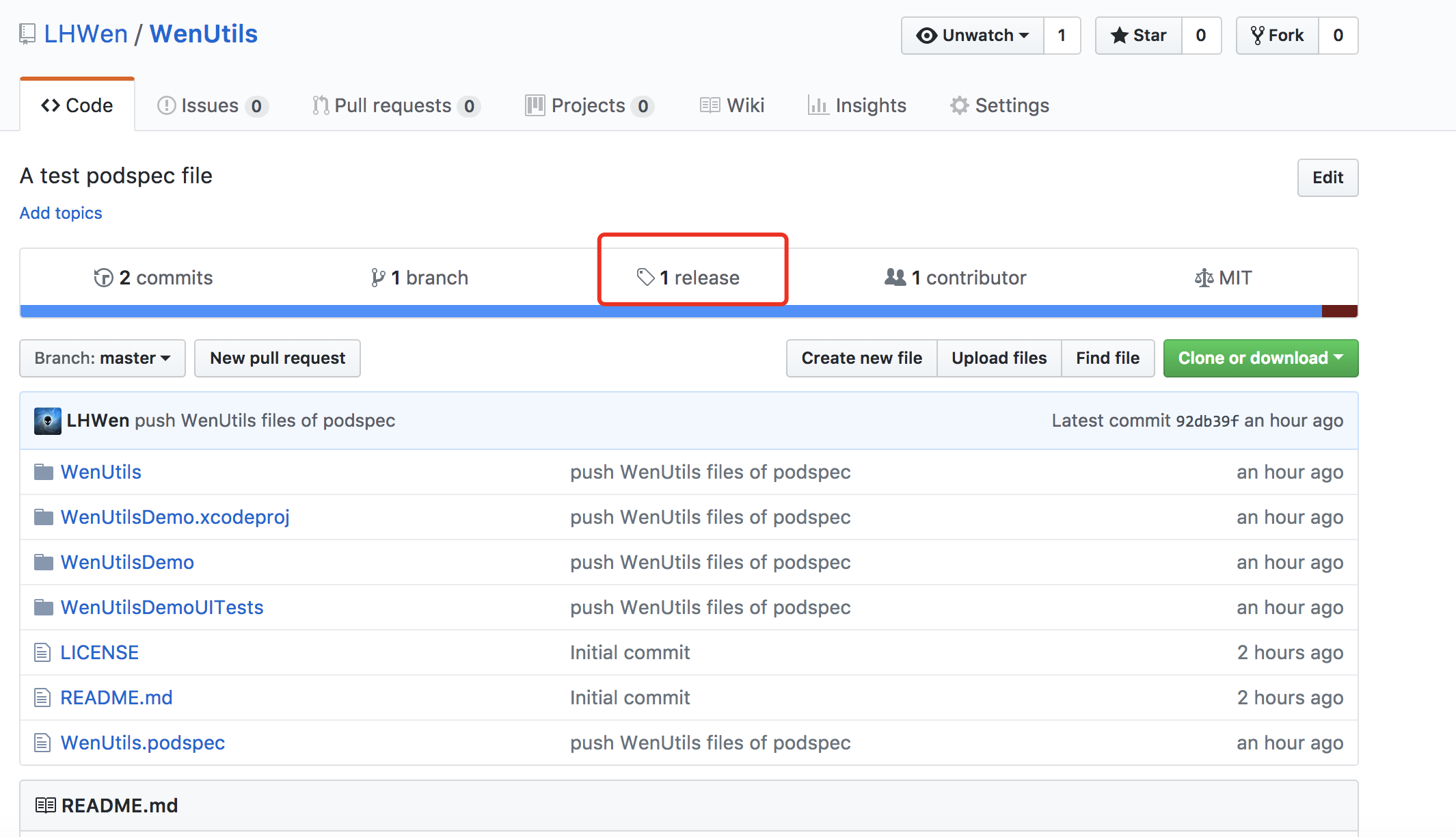
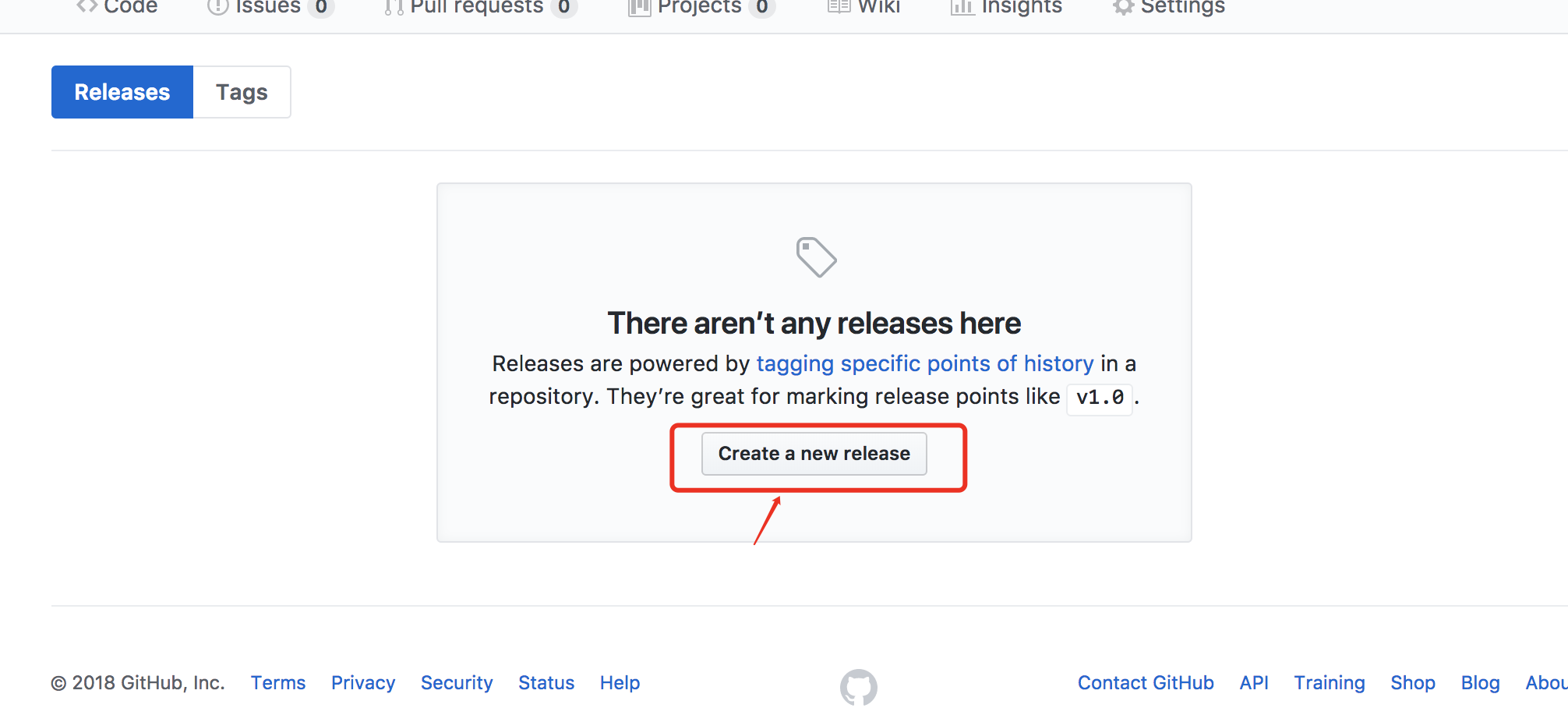
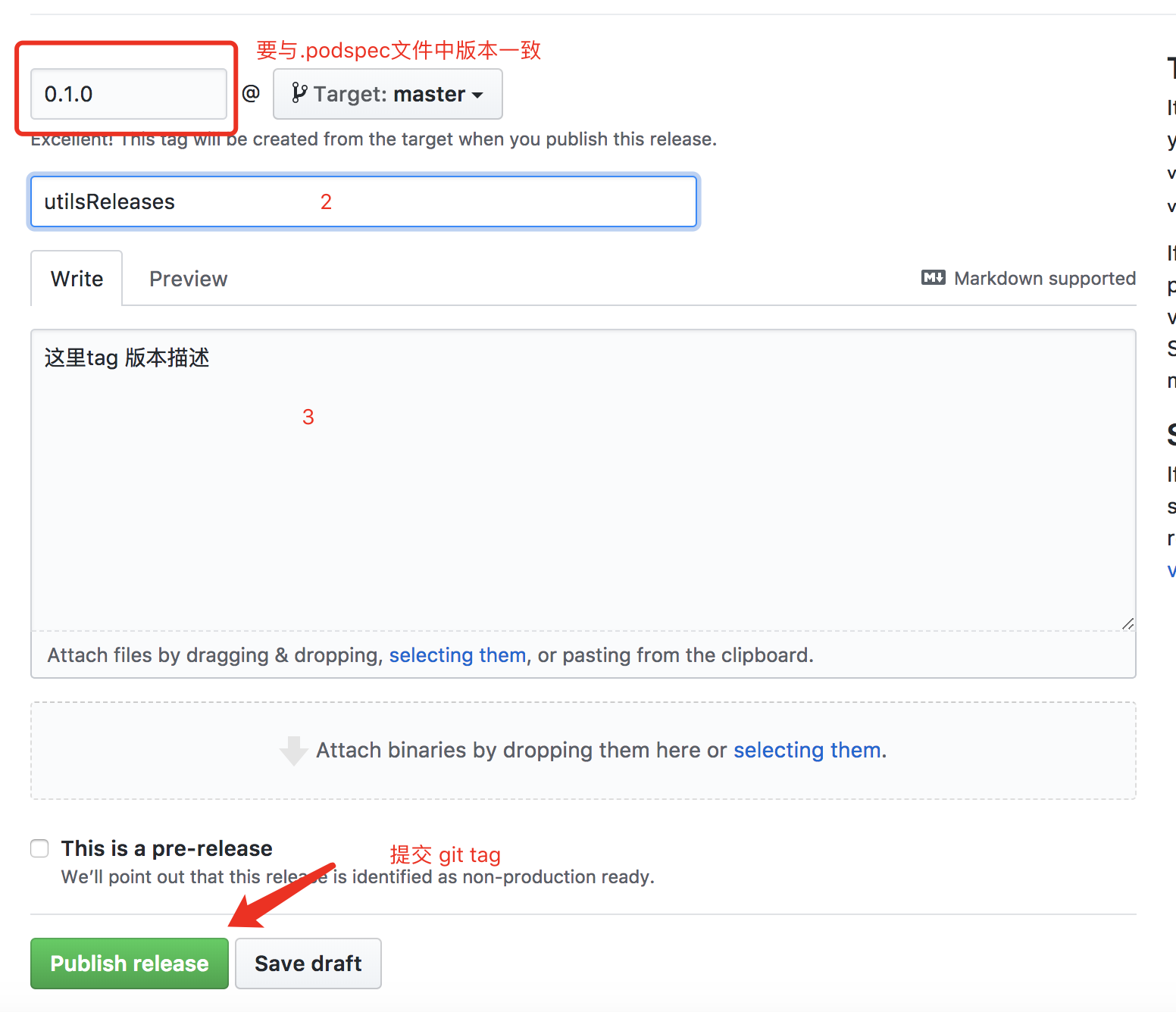
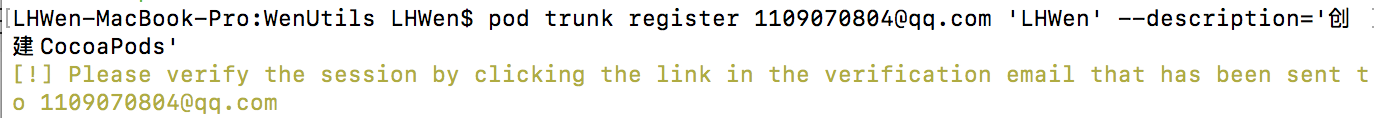
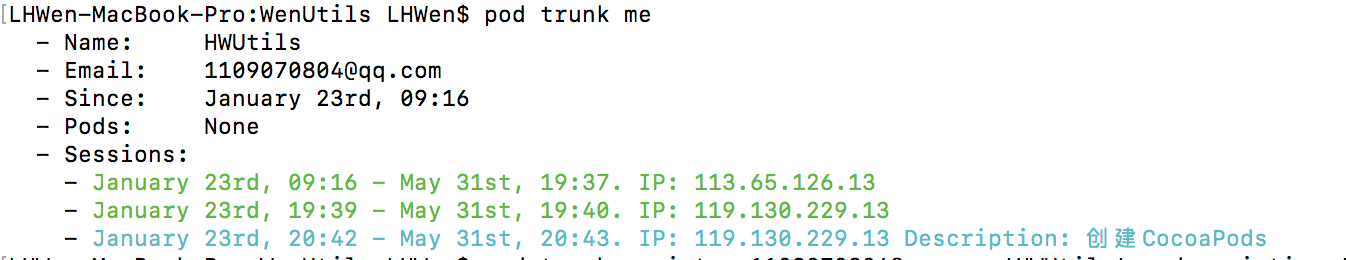
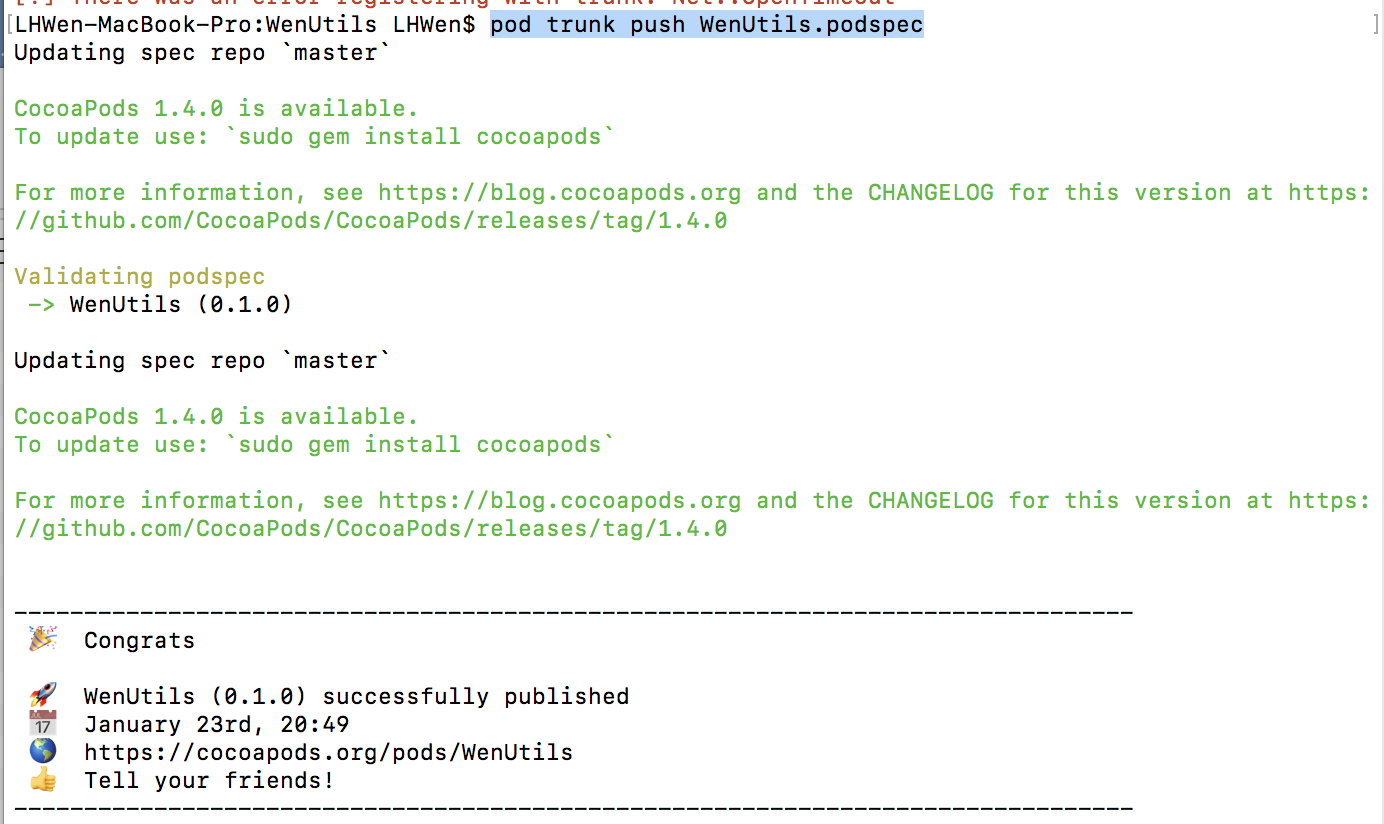
podspec 创建工具类 使用 pod来管理流程

1. 创建一个空仓库



1. 蒋git文件拷贝到本地，使用空文件（如：TestPodspec,下面会使用这个文件夹名称）接收，clon下来的文件夹中会存在以下两个文件  
   
2. 创建一个demo，蒋demo中的所有文件copy到TestPodspec文件夹中，同时在TestPodspec文件夹中创建与仓库名称一致的工具文件夹（TestTools），TestTools中添加需要生成pod的所有文件类；使用TestPodspec中的启动文件运行demo，同时在启动的demo中引用TestTools中的类，启动运行看是否报错。  
   注意：TestTools与demo启动文件是同级关系！！！  
   文件结构如下：  
   
3. 打开终端，cd 到TestPodspec文件夹运行：  
   $pod spec create 仓库名称   
     
   使用$open 仓库名称.podspec 命令
4. 编辑 .podspec 文件，如果不知道如何编辑可以看GitHub上面已有的pod的第三方框架来编辑：  
   
5. 编辑完成，保存。本地验证，执行命令：$ pod lib lint 仓库名称.podspec --allow-warnings  
   
6. 将该文件所有内容都提交到GitHub上去，并刷新自己的GitHub页面，对该次提交打上tag   
     
     
   
7. 注册trunk，例如：pod trunk register 邮箱 ‘用户名’--descripttion=‘描述’，注意：邮箱为github上的登录邮箱、用户名为github上的用户名  
   
8. 接收发送到邮箱的链接，点击进入后就会注册成功
9. 接收发送到邮箱的链接，点击进入后注册成功，使用pod trunk me 检查，只是看看有无问题，实际意义不大  
   
10. 发布podspec 使用命令:$ pod trunk push 仓库名称.podspec



1. 更新仓库$ pod repo update
2. pod search 仓库名称
3. 如果pod search 查询不到，有可能几分钟后可以使用pod search查询到刚刚创建的SDK或者等待比较长时间，有可能是一天以后。