**Egit仓库建立与推送**

**03013306 谭美秀2016-3-22**

[**一、作业要求** 2](#_Toc447030665)

[**二、egit的安装** 2](#_Toc447030666)

[**三、本地仓库建立** 2](#_Toc447030667)

[**3.1建立本地仓库** 3](#_Toc447030668)

[**3.2建立PyDev Project** 3](#_Toc447030669)

[**3.3建立Py类型文件** 4](#_Toc447030670)

[**3.4将AC\_01.py加入本地仓库repository\_DW** 5](#_Toc447030671)

[**四、** **将本地仓库推送到github** 6](#_Toc447030672)

[**五、** **将作业推送到github** 8](#_Toc447030673)

**一、作业要求**

作业三：

建立本地GIT仓库，并推送到GITHUB。客户端GIT软件建议使用EGIT，也可是其他任意，同学们感觉易上手的GIT软件，如: Github for Windows。

  要求：

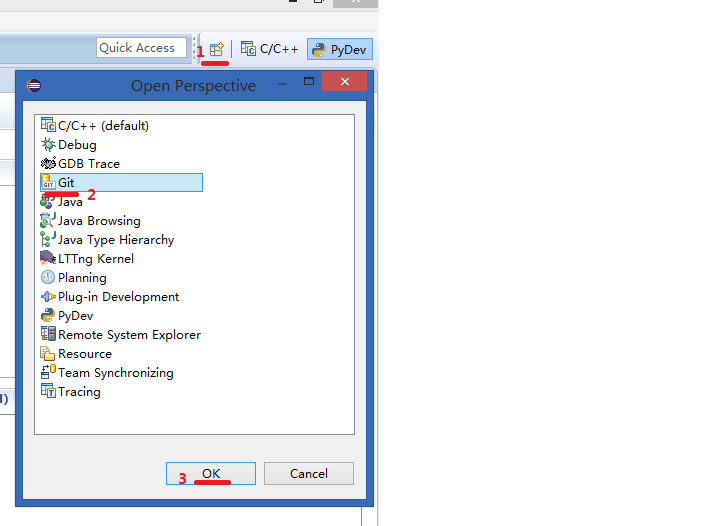
1、本地建立不少于2个仓库，对应GITHUB中的：文档和代码作业仓库；

2、将第1，2次作业文档推送到GITHUB中；

3、记录本地仓库建立和推送过程（需采用实践过程中的截图），撰写规范文档，并推送到GITHUB。

**二、egit的安装**

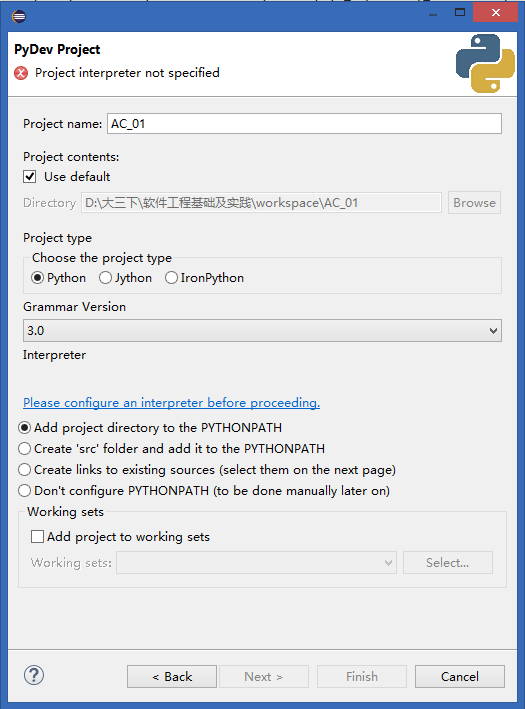
由于原先安装了Eclipse，其中自带了egit故不需要安装只需调用就好。如下图所示，调出egit。



**三、本地仓库建立**

一开始我是先create a new local Git repository，但发现这样的话无法实现在其中建立python的project将本地仓库与github中的仓库对应。故决定先建立python类型project利用py文档将他们建立联系，抱着不知是对还是错的心态一步步往下走。

不知为何，建立python.project的时候出现了bug。



然后我决定重新配置Python。

1. 安装PyDev插件，通过Windows->Eclipse Marketplaces进入市场，输入Pydev，找到Pydev安装/更新项目，在线安装。
2. 安装好后，重新启动。通过：Windows->Preference->Pydev->Interperters->Python Interperter点其中的：Advanced Auto-config。

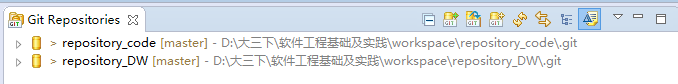
然后再建立python类型的project的时候发现成功了，至于原因我真不懂。我还是接着干活吧。

**3.1建立本地仓库**

点击如下图标

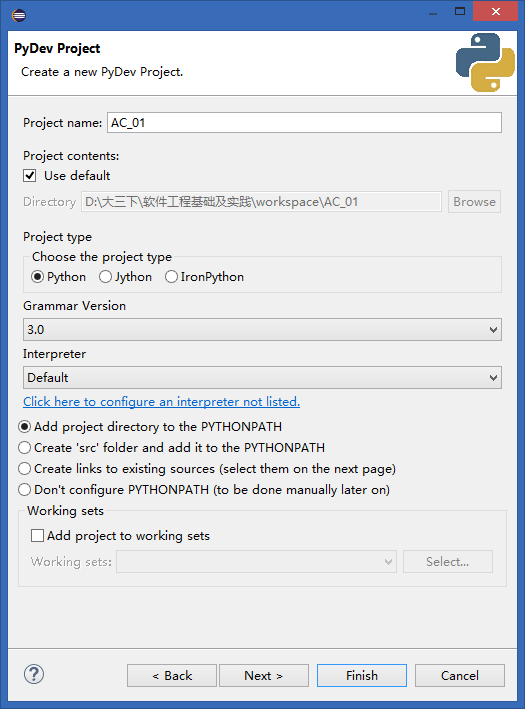
C:\Users\lenovo\Desktop\无标题.png

选择指定路径，然后建立如下仓库（此截图为完成推送后），一定不要勾选bare选项。



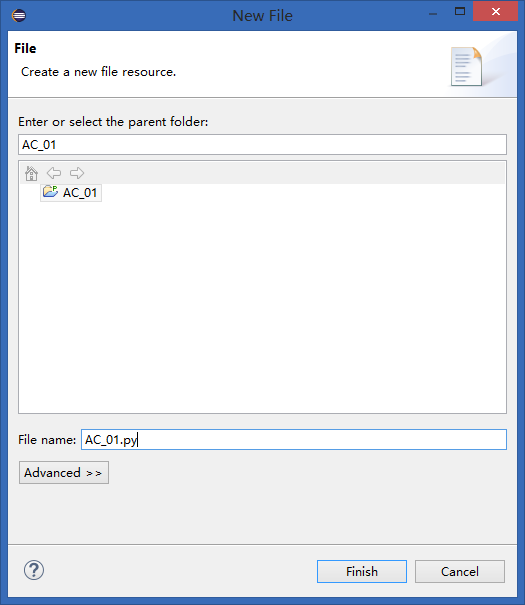
**3.2建立PyDev Project**

File🡪 New🡪PyDev Project，命名如下所示AC\_01

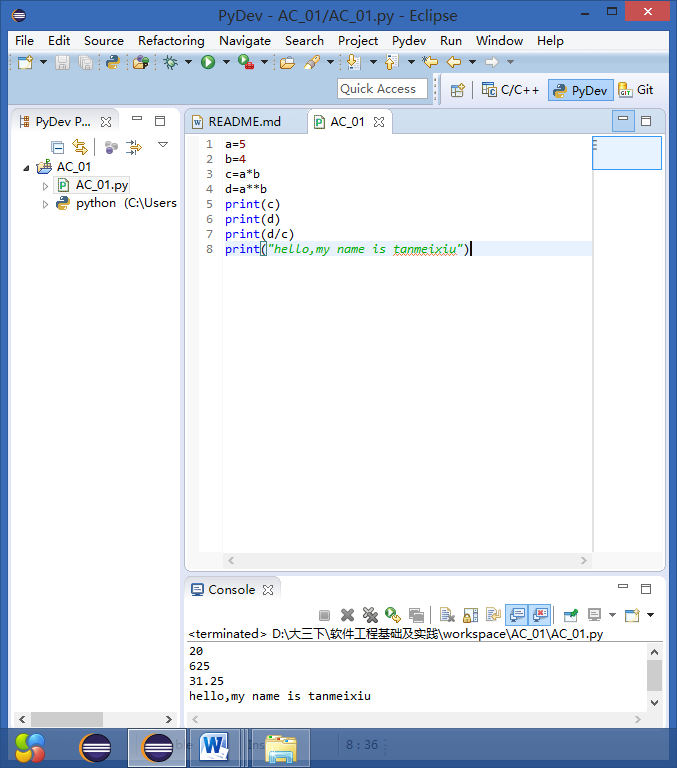


**3.3建立Py类型文件**

右击AC\_01的PyDev Project🡪File，如下图命名，记得加入.py的后缀，表明为python类型。

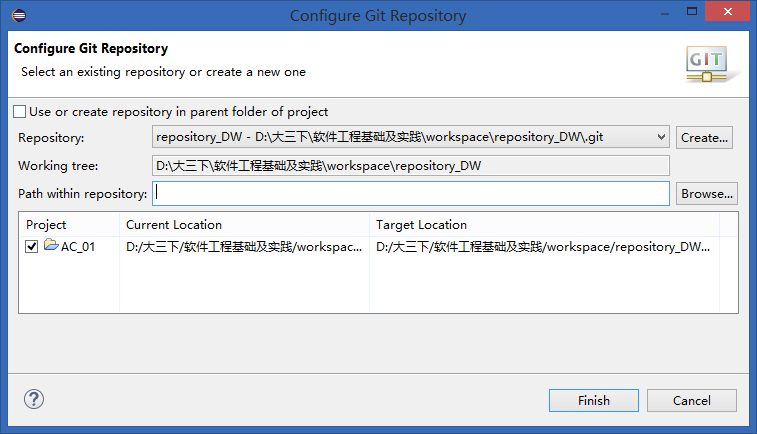


对AC\_01.py进行如下尝试：

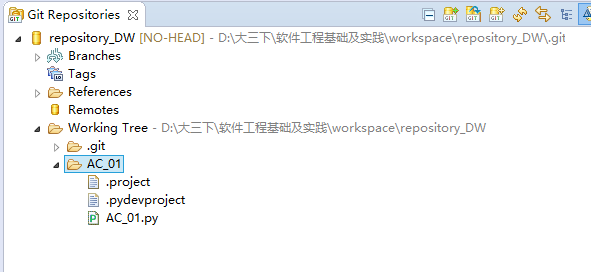


**3.4将AC\_01.py加入本地仓库repository\_DW**

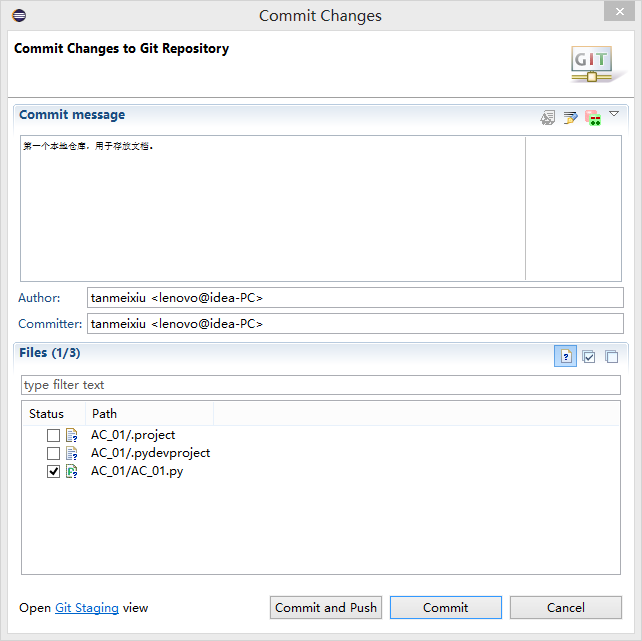
右击project AC\_01🡪team🡪share project,如下图所示



再去查看原先空着的本地仓库repository\_DW时，发现project AC\_01已成功加入。

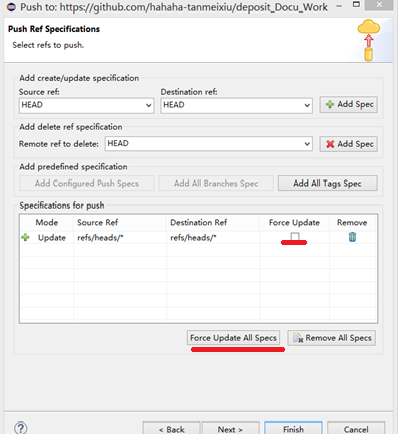
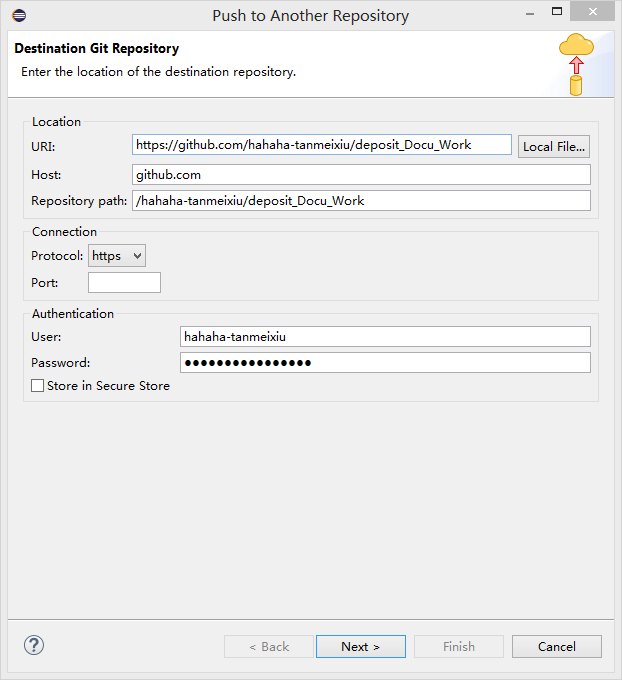


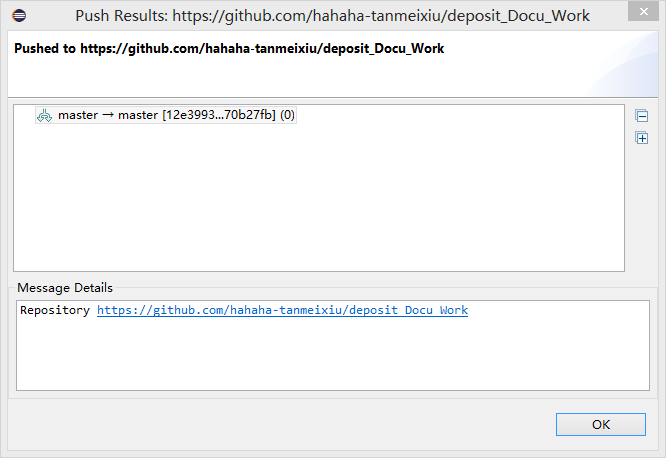
将AC\_01.py更新到仓库中，右击PyDev环境下的AC\_01.py🡪Team🡪commit,然后如下图所示，勾选所需commit的文档，这也是egit中不太方便的地方文档无法同步更新．



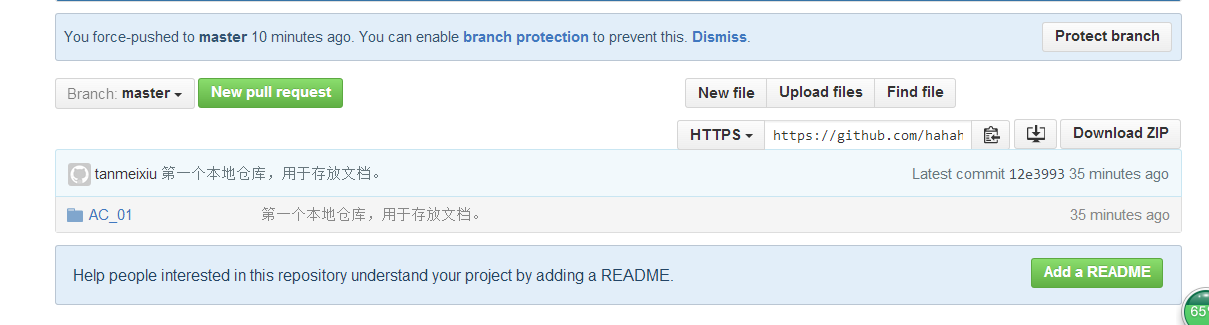
1. **将本地仓库推送到github**

正如先前猜测的那样，两个仓库之间以Pyproject.AC\_01建立联系.即将AC\_01在推送到github账户中的相应仓库repository\_Docu\_Work.此仓库在作业二中已成功建立。右击AC\_01🡪Team🡪Remove🡪push(第一个)，则如图所示，将github中的相应账户网址copy放入如图示位置。点击ＯＫ即可，记得采用force all,强行建立联系。

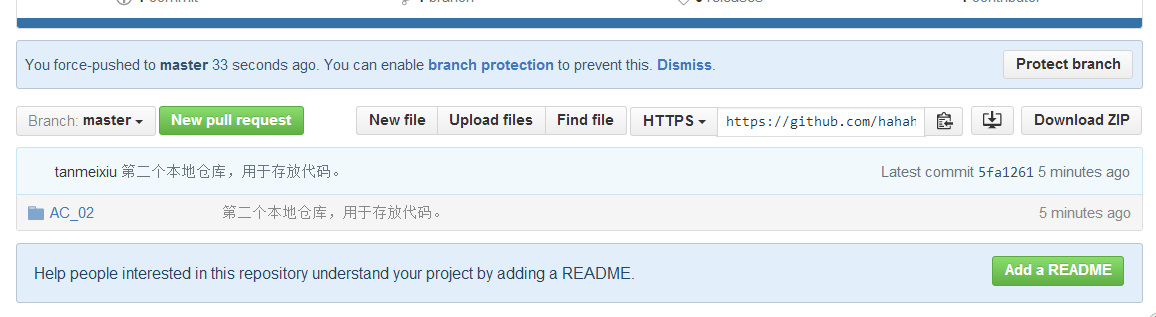




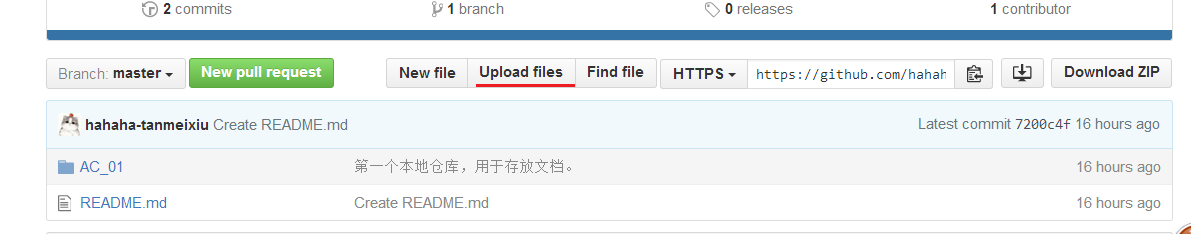
去查github账户中的repository\_Docu\_Work仓库可以看到如下图所示，故推送成功。

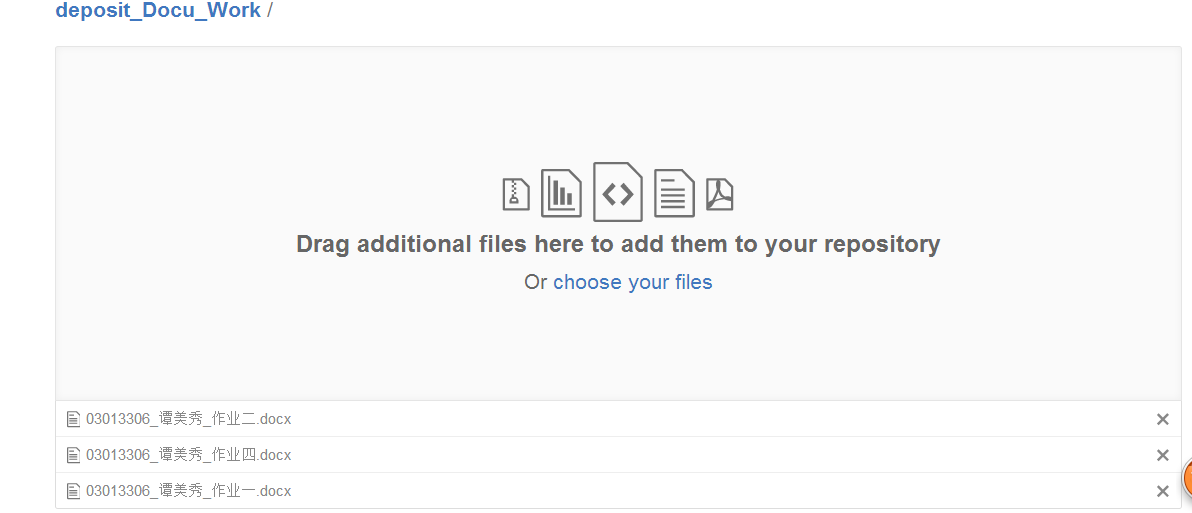


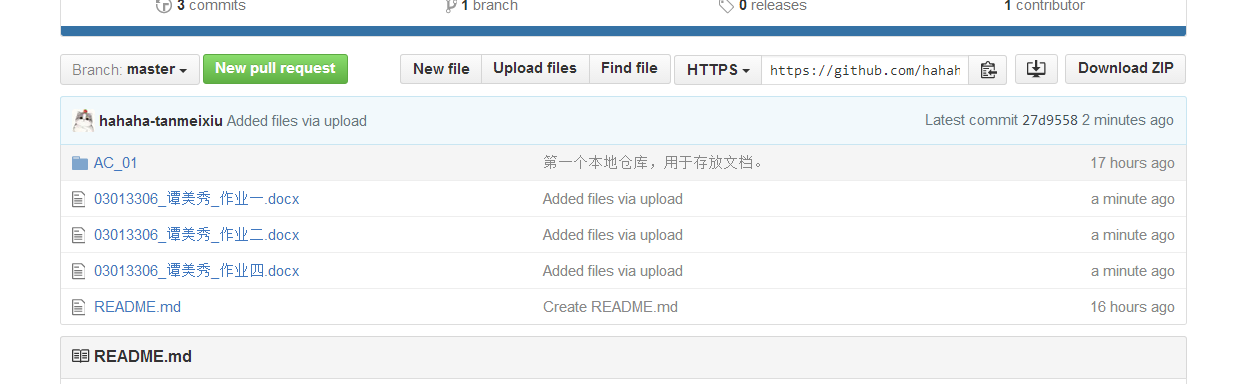
用同样的方法建立本地仓库repository\_code与推送到github账户的repository\_code，结果如下图所示。创建成功。



1. **将作业推送到github**

****





上传成功。

1. **心得与疑惑**

在完成作业三的过程中我历时了近两周，其中主要是因为出现了两个困扰我的问题。

在建立的过程中基本靠着自己的直觉与猜测在一步步往下，踩过坑但庆幸的是最后的结果总是好的。

1. 在推送的时候为什么不可以先建立本地仓库再通过本地仓库中的project与github中的相应仓库建立联系而是通过一个PyDev project作为两个仓库之间的桥梁？
2. 为什么在推送之后我原先已经存放在github中的作业一、二、四不知道跑哪了？