### Git分支划分规则

Master、UATBranch、SITBranch、DevBranch、个人分支

定义如下：

Master分支为线上稳定版

UATBranch分支为准生产环境分支

SITBranch分支为提供给测试部门的分支

DevBranch分支为开发分支

注意：平时开发需要clone DevBranch分支进行开发，开发完后merge到DevBranch分支，分支统一由Team leader 进行授权merge

### 个人分支命名规则

工号+需求简称+工程名称

例如：开发动拨工作流需求，命名个人开发分支如下：

FPM0304\_Flow\_FocsServer

### 分支合并顺序

#### 一般情况：

Master

UATBranch

SITBranch

DevBranch

......

……

FPM0304\_Flow\_FocsServer

如图1所示：个人分支一般情况都会Merge到DevBranch、DevBranch分支再Merge到SITBranch分支，以此类推。一般不允许跨分支Push代码。

#### 紧急修复上线Bug

Git仓库

Master

4

7

1

UATBranch

3

FPM0304\_Flow\_FocsServer

......

DevBranch

SITBranch

……

2

新建个人Bug分支

5

6

图2

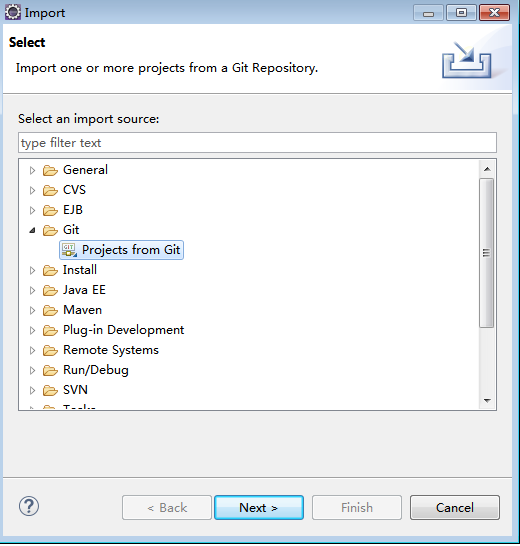
如图2所示，修复线上紧急Bug：

1. 从Master上Pull代码到新建个人分支
2. 开发完成后Push 到SITBranch,进行测试验收
3. 将SITBranch Merge到UATBranch
4. UATBranch Merger 到Master
5. SITBranch Merger 到DevBranch
6. Merger 个人分支和DevBranch分支

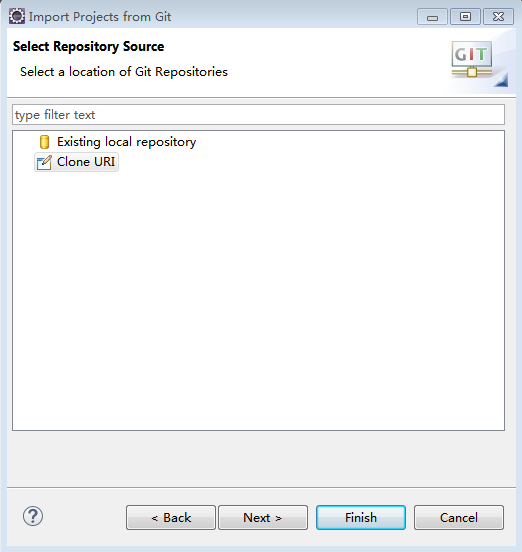
7、删除此分支并Push到远程仓库删除

### Git工程导入Eclipse操作步骤

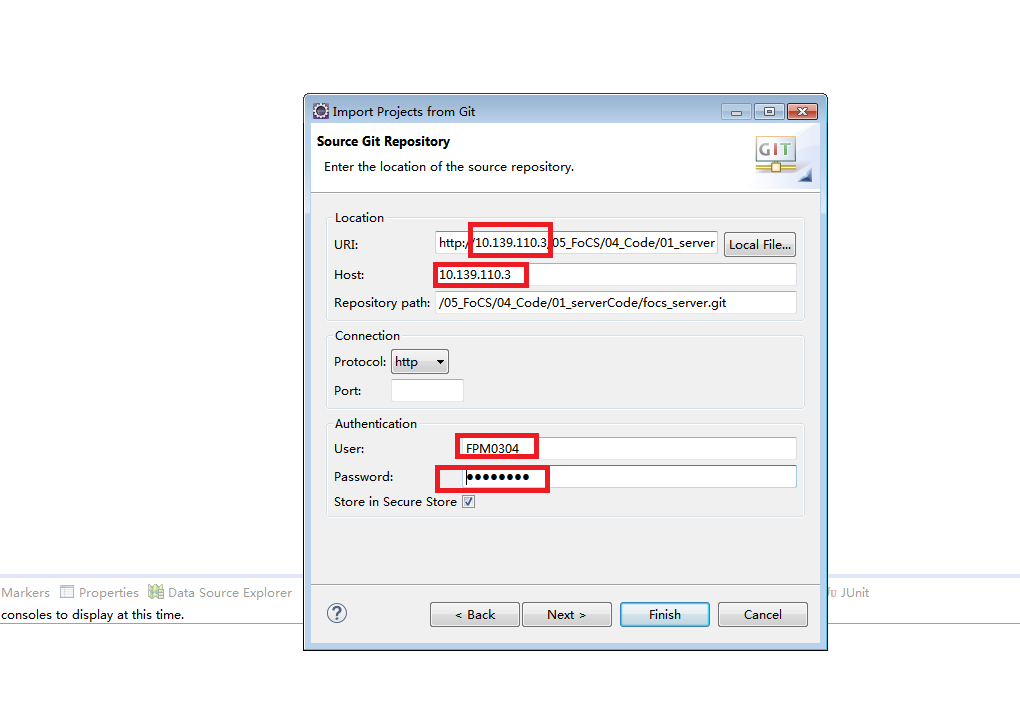
Project Explorer右键导入,选择Git



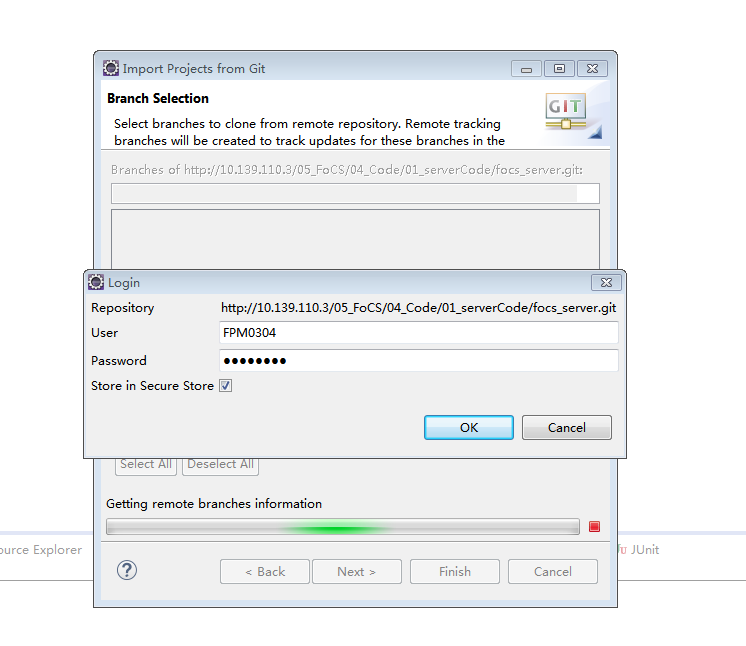
选择Clone URI，点击Next



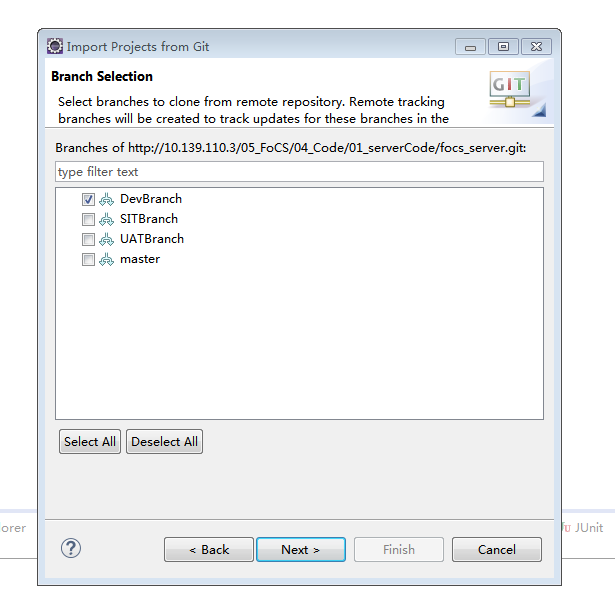
填入链接地址和账号密码，点击Next



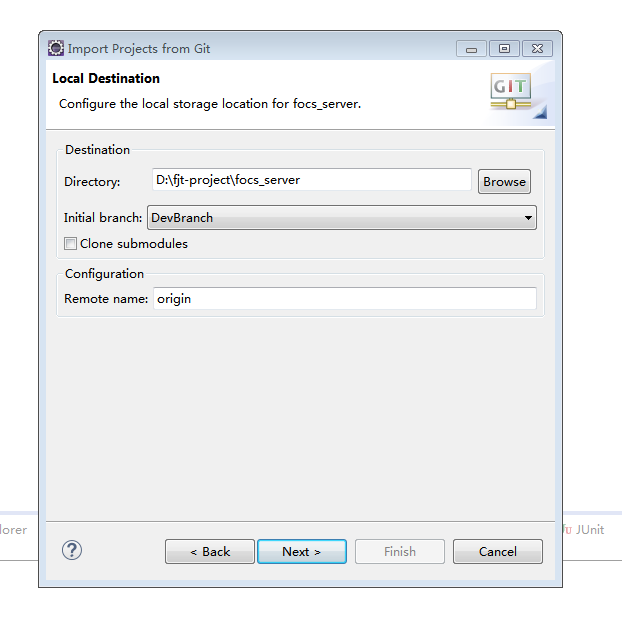
填入用户名密码，勾选 Store in Secure Store



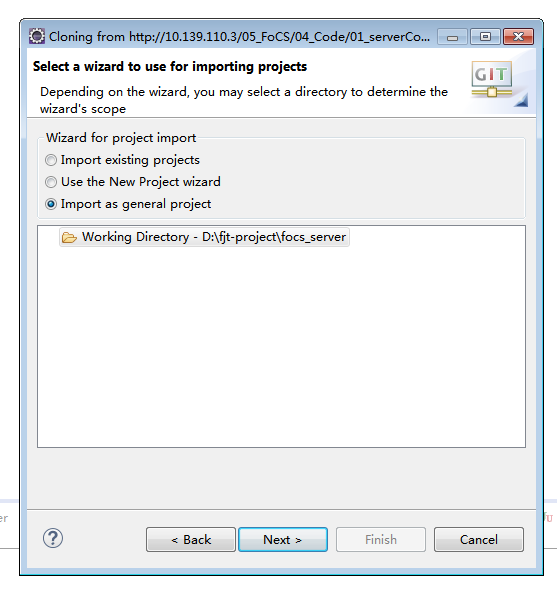
选择分支



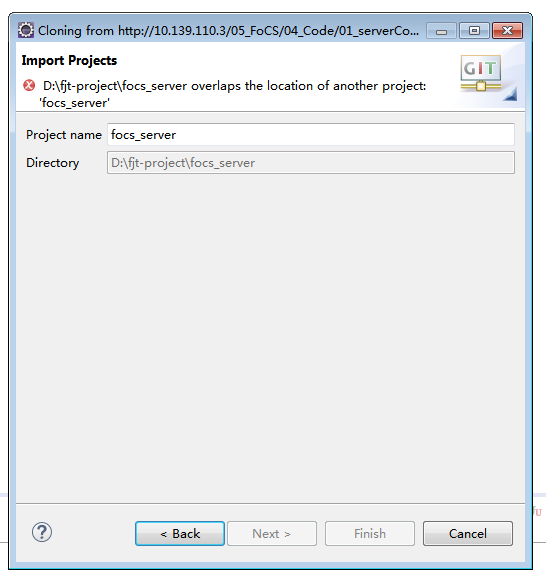
选择目录和分支，点击Next



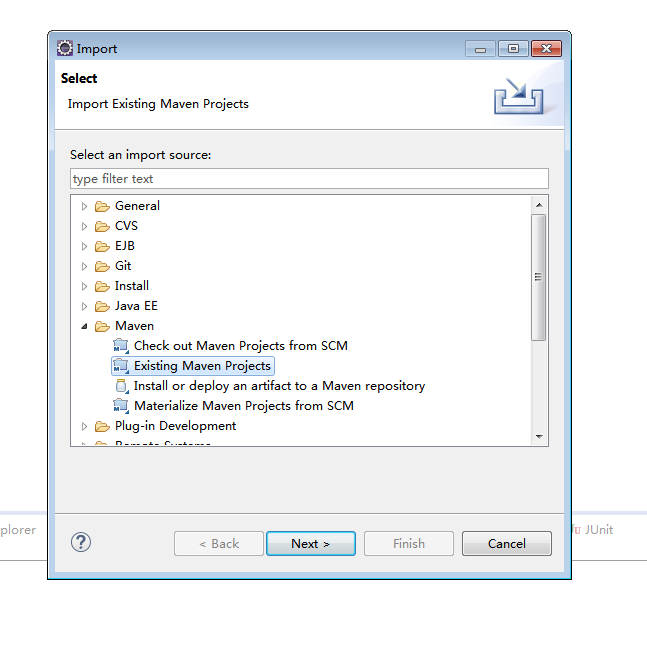
选择import as general project，点击next



下一步，因之前工作空间导入过此项目，点击 Cancel



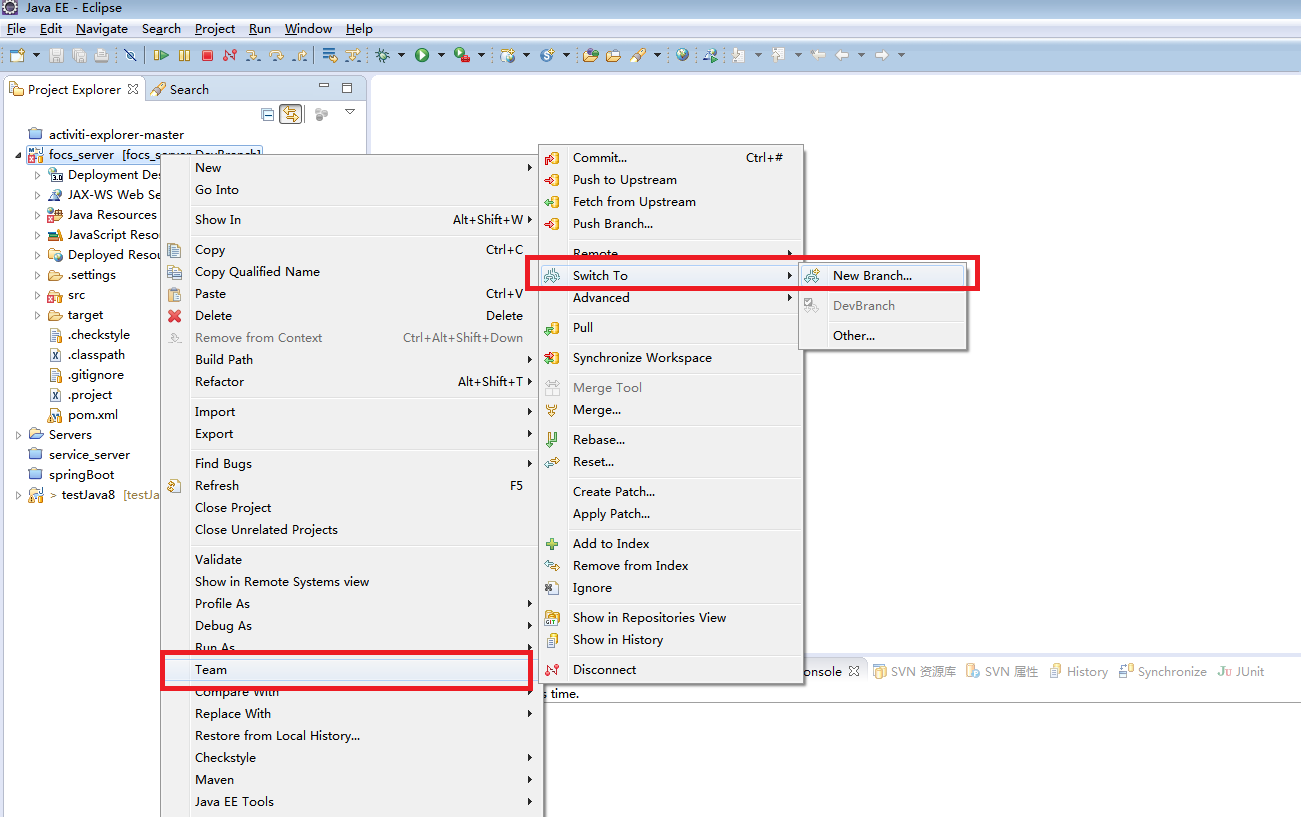
选择import Maven项目



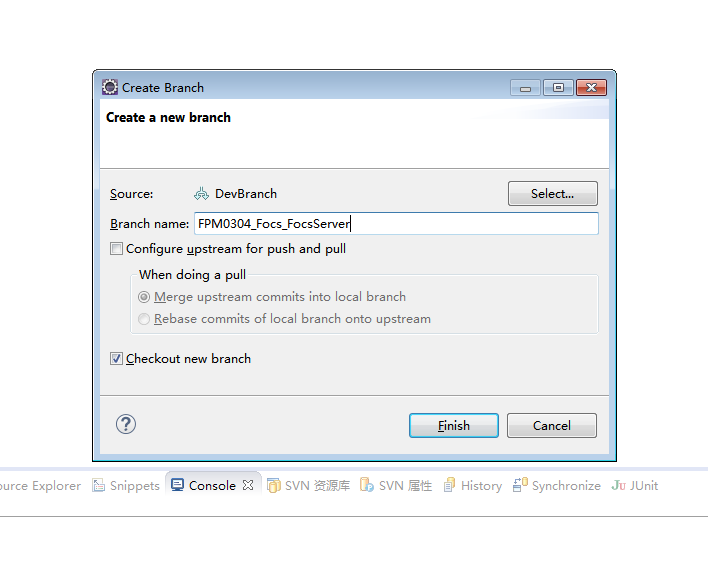
找到项目目录导入即可

#### 新建个人分支

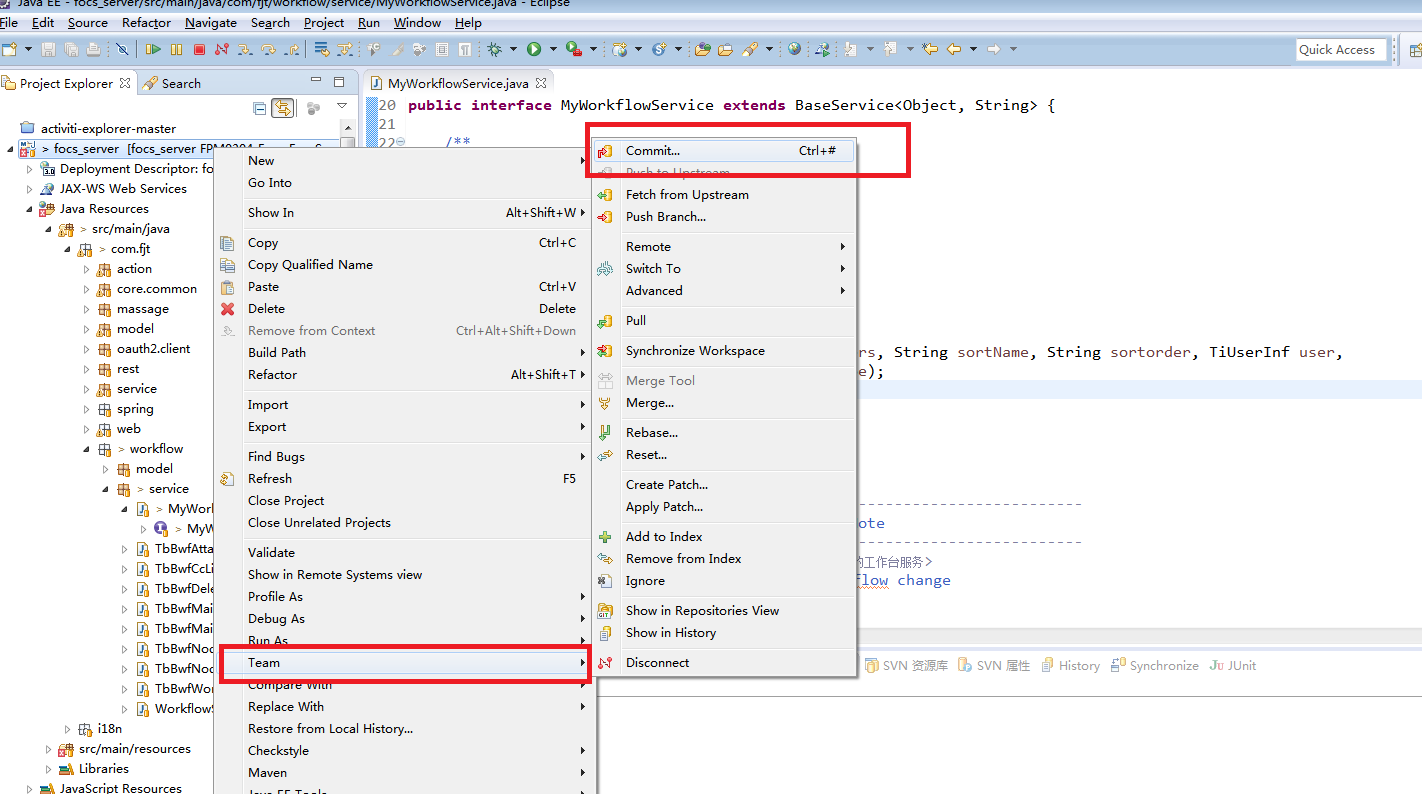
项目右键——》Team——》Switch To ——》New Branch



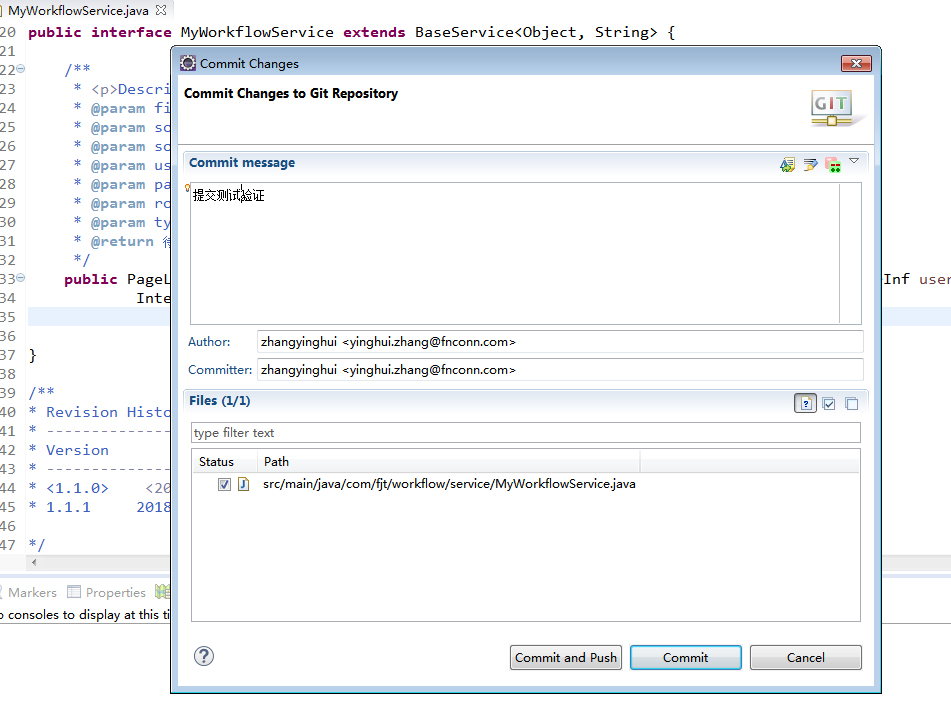
根据个人分支命名规范添加分支，点击Finish



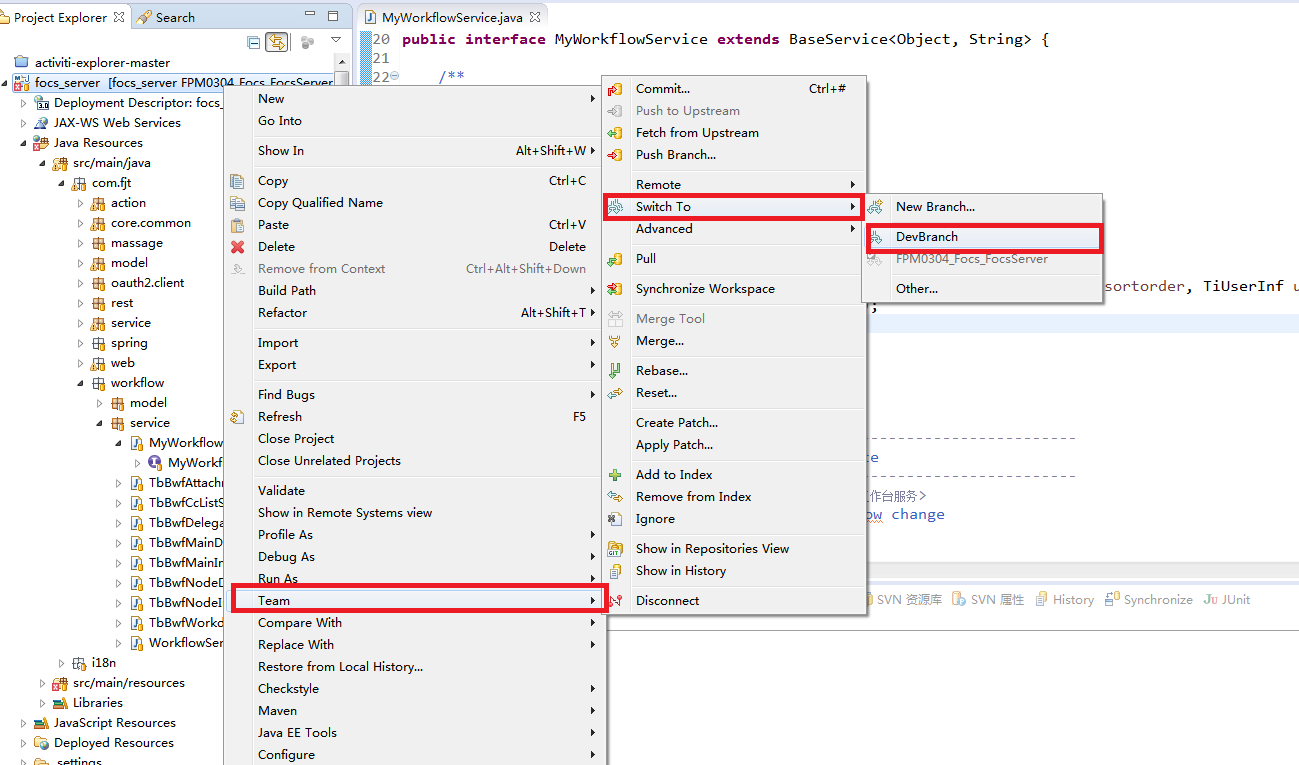
修改代码后首先进行Commit，如图：



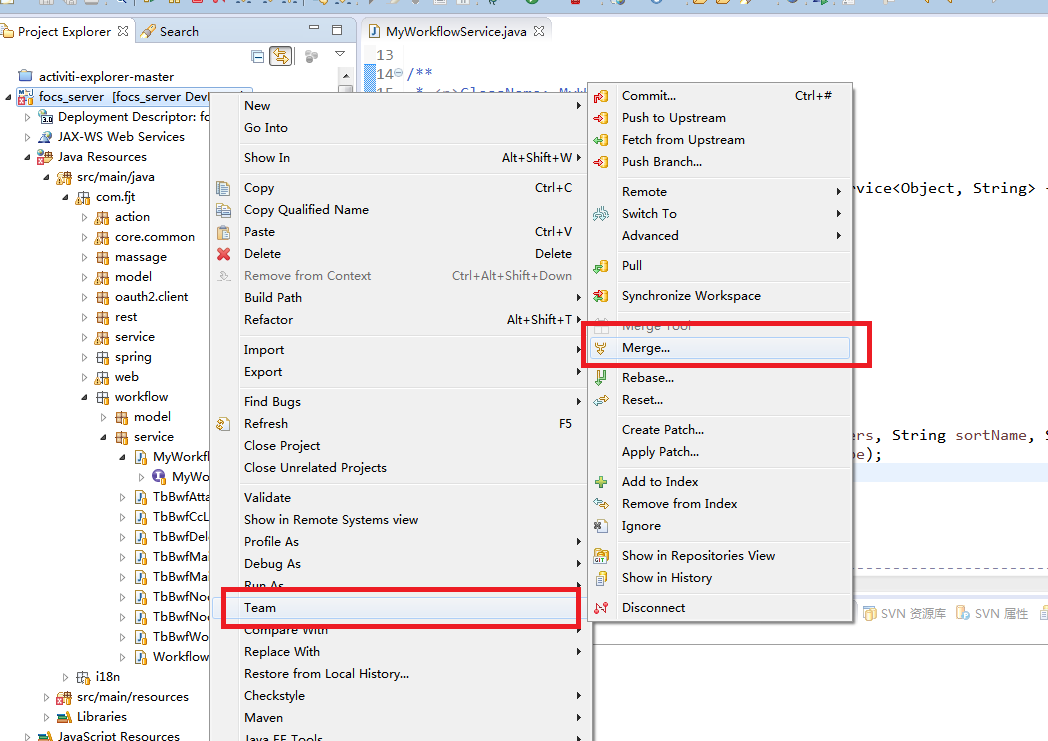
填写Commit message，点击Commit



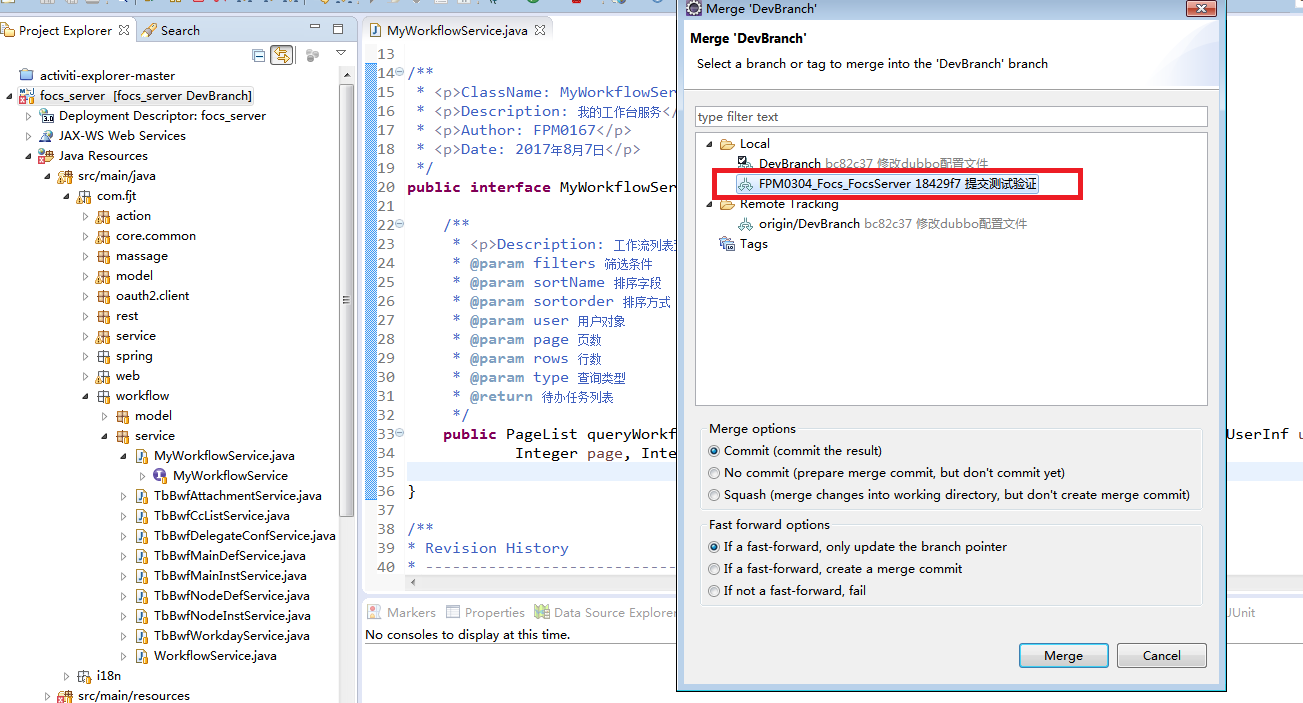
切换分支



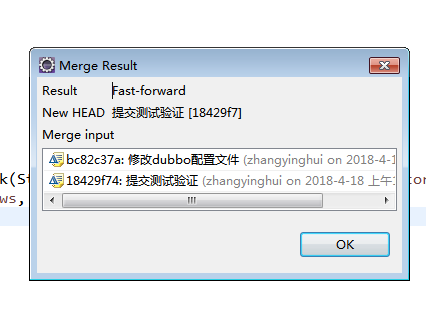
Merge分支



选择个人分支，点击Merge

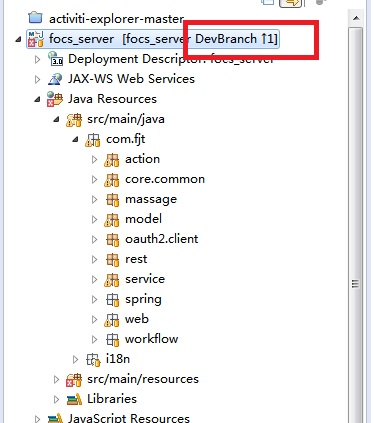


成功Merge后提示

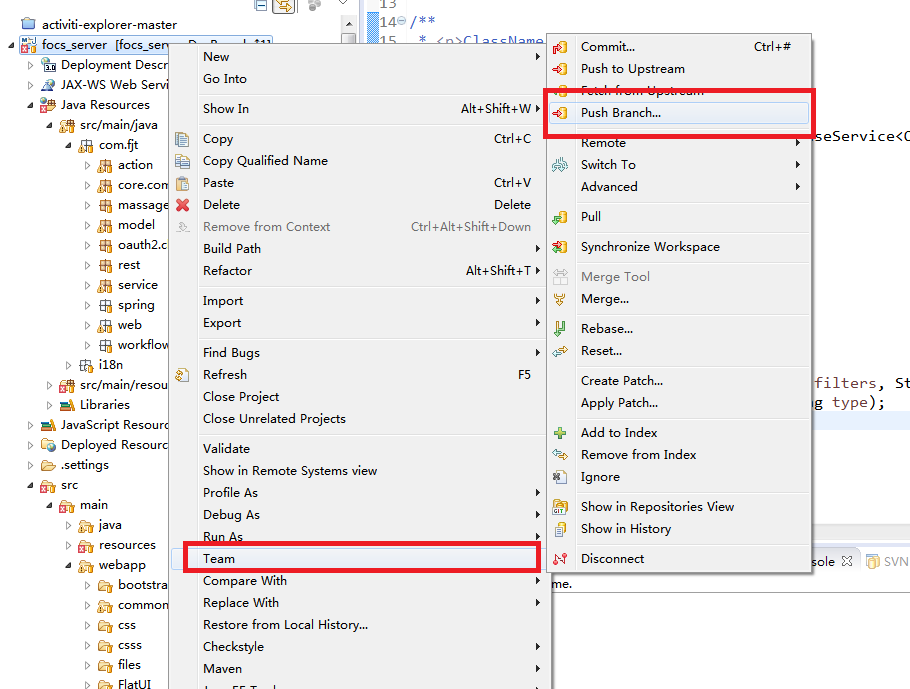


#### Push DevBranch

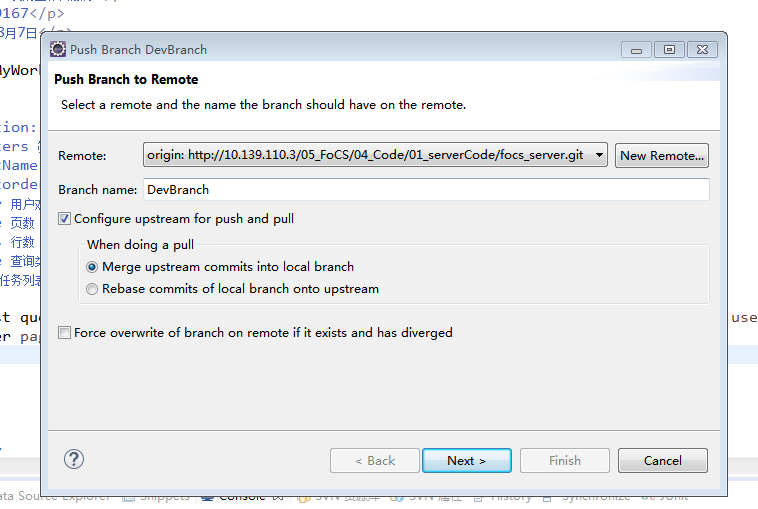
如图DevBranch有一个向上的箭头，后面的数字表示和服务器比较有多少个新的提交



选择Push Branch



点击Next



### 创建merge request

采用merge request 方式来进行codereview，步骤如下：

1. 本地创建一个本地分支，比如叫 xx\_branch。
2. 改动xx\_branch 你需要改动的代码。
3. Commit 并且 push你的分支到远程分支。
4. 在gitlab上操作，进入xx项目下，点击merge request 选项，然后选择你之前推到远程的xx\_branch和你要合并到哪个分支，比如你要合并到SITBranch上。
5. 点击 merge request。

