

Hi All，下面是使用 Git 开发过程中具体使用的步骤总结，包括常用命令的使用步骤，纯干货分享，大家可以结合之前分享的邮件花点时间熟悉下。

为了避免不必要的麻烦，请大家在本地建 Git 目录的时候与下面第二幅截图保持一致，谢谢！

git 开发通用流程总结：

1、首先请按规范配置自己在公司用的 Git 账户，参考下面链接即可

<http://conf.ctripcorp.com/pages/viewpage.action?pageId=58557925>

2、登录到 code.ctripcorp.com 上找到自己感兴趣并且有权限访问的项目代码，我们 MICE 所有服务器端源码都可以在

Project 里输入 Mice 检索

公司旅游,会议旅游,商务 x Projects | code.ctripcc x wcf 推荐书籍\_百度搜索 x 微信网页版

code.ctripcorp.com/#/admin/projects/?filter=mice

应用 Ctrip Login develop-6381 FAT - Login UAT- Login Ctrip Login 携程 - 治理中心

All My **Projects** People Documentation GitBlit

List General Branches Access Dashboards

## Projects

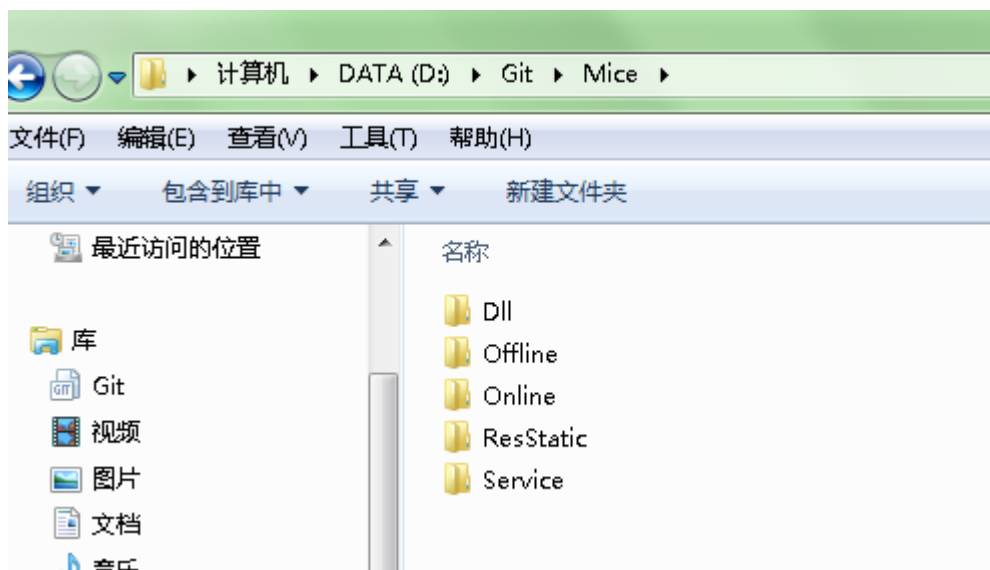
Filter

S	Project Name
	Q Mice/Dll/Library
	Q Mice/Offline/Mice.Acc
	Q Mice/Offline/Mice.BizProxy
	Q Mice/Offline/Mice.Common
	Q Mice/Offline/Mice.Input
	Q Mice/Offline/Mice.MiceOperate
	Q Mice/Offline/Mice.MiceOperate.Business
	Q Mice/Offline/Mice.Order
▶	Q Mice/Offline/Package.MiceOperate.Entity
	Q Mice/Online/Ctrip.Mice.Booking
	Q Mice/Online/Mice.BookingMeeting
	Q Mice/Online/MiceBooking
	Q Mice/ResStatic/H5/TourMice
	Q Mice/ResStatic/Online/ResTourOnline
	Q Mice/Service/H5Service
	Q Mice/Service/Jobws
	Q Mice/Service/MiceBookingServices
	Q Mice/Service/MiceInsuranceServices
	Q Mice/Service/MicePictureService
	Q Mice/Service/MiceServices

3、在 Clone 代码到本地目录之前,请先在本地 D 盘下面新建个 Git 文件夹,在 Git 文件夹下新建一个 Mice 文件夹,在 Mice

文件夹下依次新建 Dll, Online, Offline, ...等文件夹,这样本地目录就类似 D:\Git\Mice\Offline 这样的结构,这

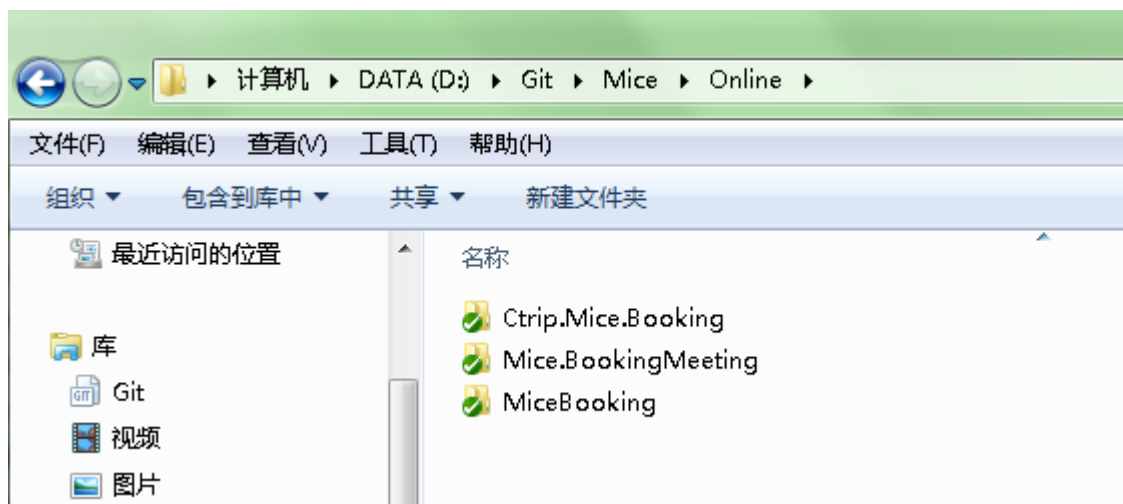
个目录结构也是和远程仓库的项目路径保持相对一致的。



4、然后本地进入相应文件夹，比如我要 Clone 的是远程 Offline 下的 Mice.Order 项目，就可以进入 D:\Git\Mice

\Offline 文件夹然后右键选择 git bash，从而打开 Git 命令行窗口，然后将选择的 git clone 代码复制到命令行后点击

Enter 就会将远程仓库对应项目的代码 Down 下来了。



5、Clone 完代码之后，我们在 VS 中打开对应的项目，然后先编译整个 Solution 一遍，没有问题就可以开始自己的开发

了，当我们开发完之后，只要进入到之前的命令行窗口，按下面的步骤操作即可完成代码的本地 Commit 和向远程仓库

push 的操作:

`git status` 查看本地目录源码文件的所有改动状态

`git add -A` 或者 `git add -- file Need to Commit` 或者 `git add .` 这一步是将本地工作区 (IDE) 中的变动的文件加入

到本地的暂存区, 在暂存区中的文件都是你确定要 Commit 到本地版本库的文件

`git commit -m "本次 Commit 的注释"`

`git pull --rebase origin RemoteBranchName`

如果在 pull 时没有冲突, 恭喜你可以直接进入 `push` 阶段, 否则请在本地用可视化工具解决冲突然后进行下一步 `push`

`git push origin RemoteBranchName:refs/for/LocalCurrentBranchName`

如果 `push` 成功, 请到 [code.ctrpcorp.com](https://code.ctrpcorp.com) 自行检查下你的提交是否已到服务器, 并指定 CodeReview 的人, 让他给你做

Codereview 并 Submit 你的代码到远程仓库。