[Git 介绍： 1](#_Toc10631)

[git add .--本地工程命令 1](#_Toc8504)

[git pull 1](#_Toc14539)

[git branch 2](#_Toc17676)

[Git command 5](#_Toc31369)

[Git diff 5](#_Toc4548)

[Double dash (--) 5](#_Toc4522)

[Git cherry-pick 5](#_Toc15143)

[Git pull 5](#_Toc5460)

[Git fetch 5](#_Toc19537)

[Git reset 5](#_Toc14971)

[Git revert 5](#_Toc19340)

[Git 撤销代码仓库的修改 6](#_Toc1756)

[遇到问题 6](#_Toc17873)

[无法运行git add . 6](#_Toc16632)

Qmtrip中依赖的子模块，无分支可言，只有commit点，

Ssh -T [git@github.com](mailto:git@github.com) 测试链接

Ssh -t rsa -C “1241282015@qq.com”

# Git 介绍：

三个组成部分：

工作目录、缓存区、提交历史

在本地的工程(可以是一个空文件夹）进行git操作。

git init --本地工程命令

初始化该工程，生成.git文件夹

git commit -a

在add前使用该语句，会自动监听文件的变化，add之后，即可直接git commit

## git add .--本地工程命令

告之git，你的工程内容的，生成snapshot。该snapshot存在于git的index临时区域

对于执行git add后需要将部分文件删除，此时可以使用git rm --cached filename, 此操作只会删除该文件在缓存区的映像，

Git rm -rf filename 删除该文件在缓存区和物理区的文件，强制删除

## git commit --本地工程命令

将project文件永久的存储到index中

## git pull

git pull时注意其他分支的pull， git pull --all

## git branch

1.to show all branch we have

2.to create a new branch

note:

Every branch will have a copy of files in that project（root folder).When

swiching to a new branch, and changes are being made, then after commiting,

files in this branch are different from others branch. Git uses header to

point differenct branches.

usage:

---git checkout feature,and make some changes in some files

---git add.

---git commit -m "decription"

---git push origin branchName

note:

if branchName is not in remote repository, then this new branch will be generated and those files having been modified will be send to remote repository.

update auth.accounts set is\_validate\_email=true where email='changlong.yu@jingli365.com'

select \* from auth.accounts as aa where aa.email='changlong.yu@jingli365.com'

git branch -a

1.to list all local and remote branch

Delete origin branch

Git push origin --delete branch\_name

或者

Git branch -r -d origin/branch-name (未测试)

=======

================develop branch======================

checkout

1. creating a new branch

2. creating a new branch and switch to new one

usage:

---git checkout -b branchName

note:

-b is short for branch

3. retreat modified part of some files

usage:

---git checkout filename

note:

when git status, some modified files that haven't sent to temporaray area will rollback to original status. Those changes will be removed in this file.

this line is newly added to differ feature branch

git show 显示修改内容

git pull origin branch\_name

git push origin branch\_name

usage:

having branch: feature/QMTRIP-25

git pull origin feature/QMTRIP-25

git ls-files

显示文档目录结构

git rm -f file\_name

remove files in temporary local storage。

usage：

--git rm -f www/ionic/trip/booking-details.html

git branch -d branch\_name

usage:

--to delete branch

如何复原已删除文件

git reset www/ionic/trip/create\_bak.ts

git reset HEAD www/ionic/trip/create\_bak.ts

git diff

查看本地版本与远程版本是否一致

usage:

---git diff origin/master

可能的结果：

diff --git a/JLTypes b/JLTypes

new file mode 160000

index 0000000..75803bc

--- /dev/null

+++ b/JLTypes

@@ -0,0 +1 @@

+Subproject commit 75803bc27ebb1a7cd632383d715e67dc457ed463

diff --git a/\_types b/\_types

new file mode 160000

index 0000000..239335d

--- /dev/null

+++ b/\_types

@@ -0,0 +1 @@

+Subproject commit 239335d7fb54f33398b4f937239176a43c77c197

diff --git a/common b/common

index 95e7999..bf9a57e 160000

--- a/common

+++ b/common

@@ -1 +1 @@

-Subproject commit 95e79997afebc175c477f4a4d217feff15b74a3f

+Subproject commit bf9a57e5cd2989fc68ab3b67b7a2610d4546b2b3

注意：当使用git diff test 时，提示ambiguous argument test，此时使用

git diff test --

使用checkout

执行tsc可以确定是否本地版本是否存在问题，然后决定是否需要push

(use "git add <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

# Git command

## Git diff

Usage:

Git diff origin/master 与远程分支文件进行标胶

Git diff --cached 比较当git add 执行过后

## Double dash (--)

不同的命令有不同的意义，in general，分离子命令的参数。如何git add -- -myfile.js：此时意为忽略git add的参数，直接使用

## Git cherry-pick

## Git pull

是git fetch和git merge的合体，

## Git fetch

当进行该操作时，此时只是获取远程内容的最新历史纪录。并不会进行远程端合并。获取的新版本会记录到FETCH\_HEAD中，若要合并，则合并FETCH\_HEAD

## Git reset

删除某分支的最前面的两个提交(commit提交)。

Usage：

---git reset --hard HEAD~~

其中~表示后退几个

参数：

--soft：表示缓存区和工作目录不会改变

--mixed：表示缓存区和指定的提交同步，工作目录不受影响

--hard：缓存区和工作目录同步提交到指定的提交。

## Git revert

取消某次提交

# Git remote

参考：<http://www.jianshu.com/p/7d0026305569>

在线创建一个工程(空的)，git pull获取后在本地创建一个对应的子工程(空工程)。此时需要将另一个工程拉下来，push到远程的空工程中。

此时需要添加多个remote工程，命令如下：(远程工程dtask-remote->origin, 第二个远程工程dtask-template)

Git remote add remoteName [git@git.l.jingli365.com:dtasks/dtask-template.git](mailto:git@git.l.jingli365.com:dtasks/dtask-template.git)

通过git pull remoteName拉取第二个仓库的文件

Git push -u origin master 将拉取下来的文件push到dtask-remote的仓库。

## Git 撤销代码仓库的修改

Git checkout

两种用法：

---切换分支

---切换到某次提交git checkout commit\_id

Git reset

前两者可以取消提交操作，可以取消特定文件

Git revert

## 子模块

两种状态：new commit和modified content

New commit:

表示线上的模块存在新的commit，及commit的节点改变。此时需要git diff出节点编号，进行checkout。

有时当使用checkout时，提示该节点编号不存在，表明此节点已被删除，无法进行checkout，此时需要git pull

Modified content: 当本地对子模块做修改时，会产生此情况。两种，开发中，要创建基于master的分支。测试修bug阶段，要创建基于test的分支。此时子模块、父模块都要进行commit， push操作。

Git update submodule --init

Git submodule add submodule\_url 例如: git submodule add git://github.com/chneukirchen/rack.git rack

# 遇到问题

## 无法运行git add .

提示.git/index.lock已经存在，需要手动删除

Test git revert