

常见git 操作查询

git status

git log -5

git status list stash

git diff

工具

git for visual studio

进入git 操作 界面

在项目上面右击 --- 选择菜单 Git Bash Here

Git提交步骤

1. git pull
2. git add .
3. git commit -m ' add sql script '
4. git config --show
5. git config --global
6. git config --global useremail 'jieqiang@caad.com'
7. git config --global username 'jieqiang'
8. git add .
9. git commit -m 'ddd'
- 10 git push 或者 git push origin develop

Git 新建分支

1. git branch develop
2. git checkout develop
3. git push origin develop

Git 获取分支更新

1. `git fetch --all`
2. `git pull origin develop`

Git non-fast-forward 冲突问题

1. `git push -f`
2. `git fetch`
3. `git config branch.master.remote origin`
4. `git config branch.master.merge refs/heads/develop`

我们常用 `-p` 选项展开显示每次提交的内容差异，用 `-2` 则仅显示最近的两次更新：

```
$ git log -p -2
```

在做代码审查，或者要快速浏览其他协作者提交的更新都作了哪些改动时，就可以用这个选项。此外，还有许多摘要选项可以用，比如 `--stat`，仅显示简要的增改行数统计：

```
$ git log --stat
```

1.查看 dev 有，而 master 中没有的：

```
git log dev ^master
```

2.查看 dev 中比 master 中多提交了哪些内容：

```
git log master..dev
```

3.不知道谁提交的多谁提交的少，单纯想知道有什么不一样：

```
git log dev...master
```

4.在上述情况下，再显示出每个提交是在哪个分支上：

```
git log --left-right dev...master
```

注意 commit 后面的箭头，根据我们在 `--left-right dev...master` 的顺序，左箭头 `<` 表示是 dev 的，右箭头 `>` 表示是 master 的。

```
git checkout . #本地所有修改的。没有提交的，都返回到原来的状态
git stash #把所有没有提交的修改暂存到stash里面。可用git stash pop回复。
git reset --hard HASH #返回到某个节点，不保留修改。
git reset --soft HASH #返回到某个节点。保留修改
```

tortoise



