2.Git分支

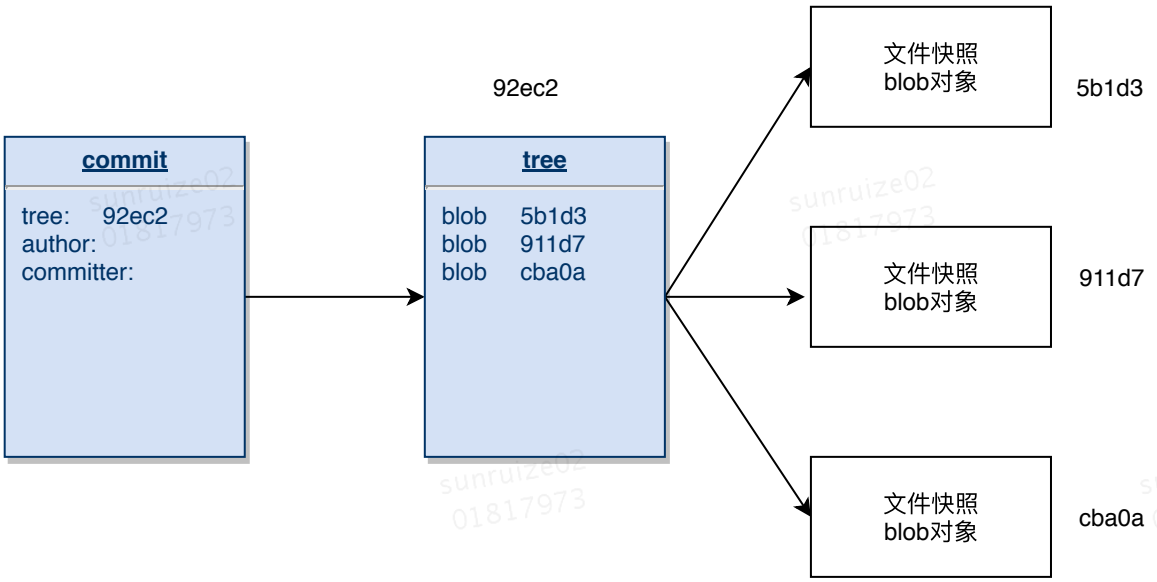
C-3 创建: 孙瑞泽, 最后修改: 孙瑞泽 昨天 18:56

Git保存数据的方式：保存不同时刻的文件快照。

当进行提交操作的时候，Git会保存一个commit对象，

commit对象中包含一个tree对象指针、作者和提交者的相关信息，如果不是第一次提交，还包含上一次commit对象的指针。

tree对象中包含暂存的文件快照的指针。



代码块

```
admindeMacBook-Pro:MyTestProjects sunruize$ git init test
Initialized empty Git repository in /Users/sunruize/MyTestProjects/test/.git/
admindeMacBook-Pro:MyTestProjects sunruize$ ls
my_pro          my_resume       test
admindeMacBook-Pro:MyTestProjects sunruize$ cd test
admindeMacBook-Pro:test sunruize$ cd .git
admindeMacBook-Pro:.git sunruize$ ls
HEAD        config      hooks       objects
branches    description     info        refs
#进行提交后，.git目录内容
admindeMacBook-Pro:.git sunruize$ ls
COMMIT_EDITMSG  config      index       objects
HEAD        description info        refs
branches    hooks       logs
```

.git目录

HEAD:指示目前被检出的分支（指向当前分支）

config:仓库的配置文件

objects:所有的数据内容

index:保存暂存区信息

refs目录:存储引用文件，指向数据（分支）的提交对象的指针

代码块

```
#向Git中存入文本，并打印哈希值
admindeMacBook-Pro:.git sunruize$  echo 'test content' | git hash-object -w —st
d670460b4b4aece5915caf5c68d12f560a9fe3e4
#查看objects目录下的内容，Git将数据的校验和的前两位作为目录，余下字符作为文件的文件名
admindeMacBook-Pro:.git sunruize$  find objects -type f
objects/d6/70460b4b4aece5915caf5c68d12f560a9fe3e4
#通过哈希值读取文件内容
admindeMacBook-Pro:.git sunruize$ git cat-file -p
d670460b4b4aece5915caf5c68d12f560a9fe3e4
test content

#存入文件
admindeMacBook-Pro:test sunruize$ echo 'test' > test.txt
admindeMacBook-Pro:test sunruize$ find .git/objects -type f
.git/objects/9d/aeafb9864cf43055ae93beb0afd6c7d144bfa4
#=====================================》》》》test.txt
.git/objects/d6/70460b4b4aece5915caf5c68d12f560a9fe3e4
admindeMacBook-Pro:test sunruize$ git cat-file -p
9daeafb9864cf43055ae93beb0afd6c7d144bfa4
test

#对test.txt作出修改，objects下将保存两个版本的数据
admindeMacBook-Pro:test sunruize$ echo 'test2' > test.txt
admindeMacBook-Pro:test sunruize$ git hash-object -w test.txt
180cf8328022becee9aaa2577a8f84ea2b9f3827
admindeMacBook-Pro:test sunruize$ find .git/objects/ -type f
.git/objects//18/0cf8328022becee9aaa2577a8f84ea2b9f3827
#=====================================》》》》修改后的test.txt
.git/objects//9d/aeafb9864cf43055ae93beb0afd6c7d144bfa4
.git/objects//d6/70460b4b4aece5915caf5c68d12f560a9fe3e4

#git只保存了文件的内容，而没有保存文件的名字，保存的对象的类型为blob
admindeMacBook-Pro:test sunruize$ git cat-file -t
180cf8328022becee9aaa2577a8f84ea2b9f3827
blob
#=====================================》》》》查询文件类型为blob对象
```

所有的数据内容都存储在objects/目录下，Git会将其内容的哈希值的前两位作为目录，余下字符作为文件名，所有文件的保存格式为blob对象

暂存修改文件

代码块

```
#接下来对test.txt执行暂存
admindeMacBook-Pro:test sunruize$ git add test.txt
admindeMacBook-Pro:test sunruize$ find .git/objects -type f
.git/objects/18/0cf8328022becee9aaa2577a8f84ea2b9f3827
.git/objects/66/8ff36f646d49868cc8ce073d153a1d08d81e61
#=====================================》》》》执行暂存后新产生的文件
.git/objects/9d/aeafb9864cf43055ae93beb0afd6c7d144bfa4
.git/objects/d6/70460b4b4aece5915caf5c68d12f560a9fe3e4
admindeMacBook-Pro:test sunruize$ git cat-file -t
668ff36f646d49868cc8ce073d153a1d08d81e61
tree
#=====================================》》》》查询文件类型为tree树对象
admindeMacBook-Pro:test sunruize$ git cat-file -p
668ff36f646d49868cc8ce073d153a1d08d81e61
100644 blob 180cf8328022becee9aaa2577a8f84ea2b9f3827
test.txt#=====================================》》》》tree对象中存储blob对象哈希值与
```

对修改的文件暂存后，Git会在objects/目录下生成一个tree对象，tree对象中包含暂存文件的具体信息（包括文件类型、文件的格式、数据内容的哈希值、文件名）

文件提交

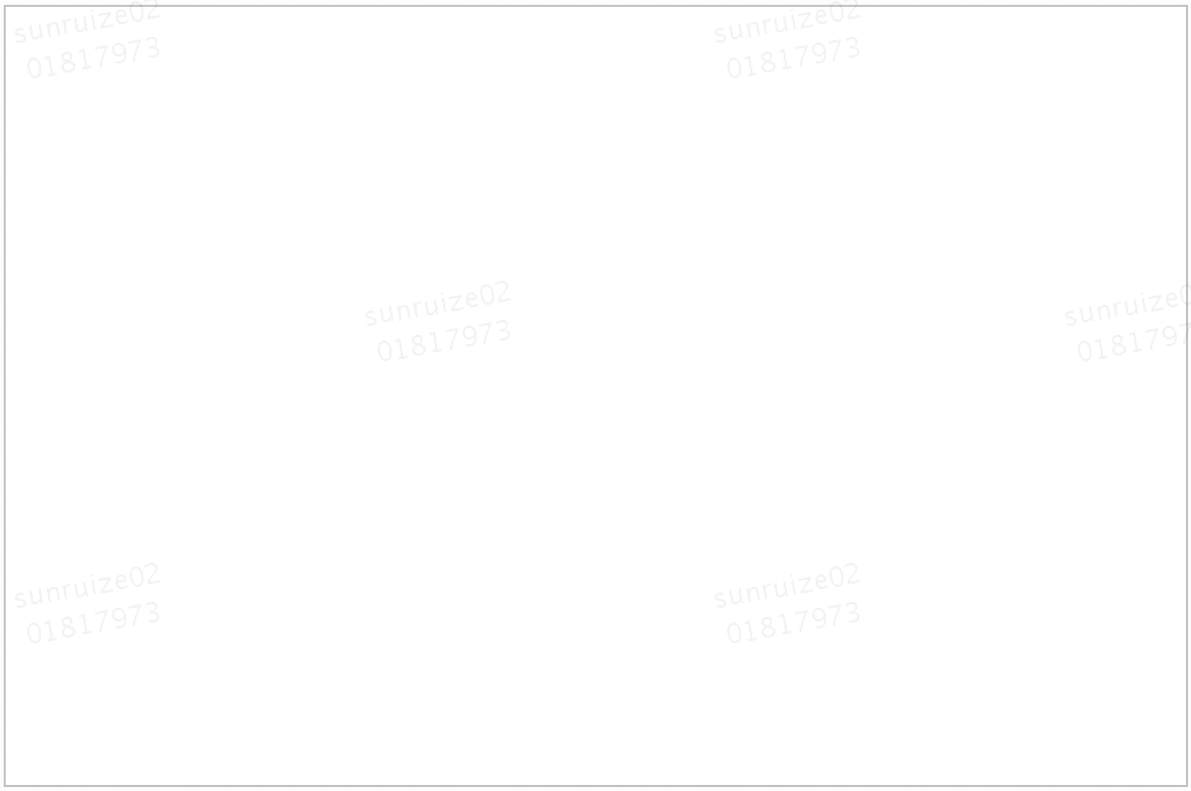
代码块

```
admindeMacBook-Pro:test sunruize$ git commit -m ‘firstCommit'
admindeMacBook-Pro:test sunruize$ find .git/objects -type f
.git/objects/18/0cf8328022becee9aaa2577a8f84ea2b9f3827
.git/objects/66/8ff36f646d49868cc8ce073d153a1d08d81e61
.git/objects/8b/682e70c1a20efa6623be058ea71c6c1226844e
#=====================================》》》》执行commit后新产生的文件
.git/objects/9d/aeafb9864cf43055ae93beb0afd6c7d144bfa4
.git/objects/d6/70460b4b4aece5915caf5c68d12f560a9fe3e4
admindeMacBook-Pro:test sunruize$ git cat-file -t
8b682e70c1a20efa6623be058ea71c6c1226844e
commit
admindeMacBook-Pro:test sunruize$ git cat-file -p
8b682e70c1a20efa6623be058ea71c6c1226844e#============》》》》查看文件的内容
tree 668ff36f646d49868cc8ce073d153a1d08d81e61
author sunruize02 <sunruize02@meituan.com> 1552964081 +0800
committer sunruize02 <sunruize02@meituan.com> 1552964081 +0800


firstCommit
```

```
#再次修改test.txt文件，查看新增的commit对象文件内容
admindeMacBook-Pro:test sunruize$  git cat-file -p 39ca23ddb95cbe413bade9aaa6a8f169dd15d365
tree cb401b5a9d6d14b89e4071d42085ee834ac86f10
parent 8b682e70c1a20efa6623be058ea71c6c1226844e
author sunruize02 <sunruize02@meituan.com> 1552973305 +0800
committer sunruize02 <sunruize02@meituan.com> 1552973305 +0800

"second commit"
```

对文件commit后，在objects/目录下会生成一个commit对象，commit对象中包含此次提交的tree对象哈希值、作者、提交者以及提交说明，如果不是第一次提交，commit对象中还含有parent指针，指向上一次commit对象

Git存储数据的模型



refs/目录和HEAD文件

代码块

```
admindeMacBook-Pro:test sunruize$ find .git/refs/
.git/refs/
.git/refs//heads
.git/refs//heads/master
.git/refs//tags
admindeMacBook-Pro:test sunruize$ cat .git/refs/heads/master
39ca23ddb95cbe413bade9aaa6a8f169dd15d365

```

```
admindeMacBook-Pro:test sunruize$ cat .git/HEAD
#=====================================》》》》HEAD文件指向当前分支引用
ref: refs/heads/master
admindeMacBook-Pro:test sunruize$ git checkout -b branchA
Switched to a new branch 'branchA'
admindeMacBook-Pro:test sunruize$ cat .git/HEAD
ref: refs/heads/branchA
admindeMacBook-Pro:test sunruize$ cat .git/refs/heads/branchA
39ca23ddb95cbe413bade9aaa6a8f169dd15d365
```

图解——远程分支

代码块

```
#当使用git clone命令克隆远程的git服务器时，会自动将其命名为origin并拉取它所有的数据，创建一的master的一个指针，本地将其命名为origin/master，
#同时Git会给用户一个和origin/master指向同一个地方的master的分支用于工作
#origin/master就是一个远程master分支的跟踪分支，跟踪分支是与远程分支有直接关系的本地分支。
admindeMacBook-Pro:my_resume sunruize$ find .git/refs/
.git/refs/
.git/refs//heads
.git/refs//heads/master
.git/refs//remotes
.git/refs//remotes/origin
.git/refs//remotes/origin/HEAD
.git/refs//tags
admindeMacBook-Pro:my_resume sunruize$ cat .git/refs/remotes/origin/HEAD
ref: refs/remotes/origin/master

admindeMacBook-Pro:my_resume sunruize$ git remote
origin
#当使用git origin命令时结果输出origin
#只要不与origin服务器进行连接，那么本地的origin/master指针就不会移动
```

图解——新建分支

仅供内部使用，未经授权，切勿外传