Дорожная карта

0. Приступаем
1. Введение в Git
2. Начало работы с Git
3. Просмотр истории
4. Ветвление
5. Слияние
6. Отмена изменений
7. Рабочий процесс

Дорожная карта (продолжение)

8. Работа в команде
9. Метки 👈
10. Последние штрихи
11. Завершаем

Метки

Суть меток

Git имеет возможность помечать (tag) определённые моменты в истории как важные.

Как правило, эта функциональность используется для отметки моментов выпуска версий (`v1.0` , и т.п.).

Просмотр меток

Просмотр имеющихся меток (tag) в Git делается просто. Достаточно набрать git tag:

```
$ git tag
v0.1
v1.3
```

Данная команда перечисляет метки в алфавитном порядке; порядок их появления не имеет значения.

Для меток вы также можете осуществлять поиск по шаблону:

```
$ git tag -l 'v1.8.5*'
v1.8.5
v1.8.5-rc0
v1.8.5-rc1
v1.8.5.1
```

Куда указывает метка

```
$ git rev-parse METKA
a236b916e8d783ccb207a1625b83e4bfec67f05a
```

Создание меток

Git использует два основных типа меток: легковесные и аннотированные.

Легковесная метка - это что-то весьма похожее на ветку, которая не меняется - это просто указатель на определённый коммит.

А вот аннотированные метки хранятся в базе данных Git как полноценные объекты. Они имеют контрольную сумму, содержат имя поставившего метку, e-mail и дату, имеют комментарий и могут быть подписаны и проверены с помощью GNU Privacy Guard (GPG). Обычно рекомендуется создавать аннотированные метки, чтобы иметь всю перечисленную информацию; но если вы хотите сделать временную метку или по какой-то причине не хотите сохранять остальную информацию, то для этого годятся и легковесные метки.

```
$ git tag -a v1.4 -m 'my version 1.4'
$ git tag
v0.1
v1.3
v1.4
```

Опция -m задаёт сообщение метки, которое будет храниться вместе с меткой. Если не указать сообщение для аннотированной метки, Git запустит редактор, чтоб вы смогли его ввести.

Вы можете посмотреть данные метки вместе с коммитом, который был помечен, с помощью команды `git show`:

```
$ git show v1.4
tag v1.4
Tagger: Ben Straub <ben@straub.cc>
Date:   Sat May 3 20:19:12 2014 -0700

my version 1.4

commit ca82a6dff817ec66f44342007202690a93763949
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Mon Mar 17 21:52:11 2008 -0700
```

Легковесные метки

Легковесная метка - это ещё один способ отметки коммитов. В сущности, это контрольная сумма коммита, сохранённая в файл - больше никакой информации не хранится. Для создания легковесной метки не передавайте опций -a, -s и -m:

```
$ git tag v1.4-lw
$ git tag
v0.1
v1.3
v1.4
v1.4-lw
v1.5
```

На этот раз при выполнении git show на этой метке вы не увидите дополнительной информации. Команда просто покажет помеченный коммит:

```
$ git show v1.4-lw
commit ca82a6dff817ec66f44342007202690a93763949
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Mon Mar 17 21:52:11 2008 -0700

    changed the version number
```

Выставление меток позже

Для отметки коммита укажите его контрольную сумму (или её часть) в конце команды:

```
$ git tag -a v1.2 9fceb02
```

Обмен метками = Записать метки на сервер

По умолчанию, команда git push не отправляет метки на удалённые серверы.

Одна метка:

```
$ git push origin v1.5
```

Все метки за раз (дубли игноируются):

```
$ git push origin --tags
Counting objects: 1, done.
Writing objects: 100% (1/1), 160 bytes | 0 bytes/s, done.
Total 1 (delta 0), reused 0 (delta 0)
To git@github.com:schacon/simplegit.git
 * [new tag]         v1.4 -> v1.4
 * [new tag]         v1.4-lw -> v1.4-lw
```

Обмен метками = Получить метки с сервера

```
git fetch --tags
```

☝️ Если метка уже существовала на клиенте и на сервере она изменилась, то метка на клиенте не будет обновлена.

Переход на метку

В действительности вы не можете переходить на метки в Git, поскольку они не могут быть перемещены. Если вы хотите установить версию вашего репозитория в рабочую директорию, которая выглядит, как определенная метка, вы можете создать новую ветку с определенной меткой:

```
$ git checkout tags/<tag_name>
```

Удалить метку локально

```
git tag -d <tagname>
```

Удалить метку удалённо

```
git push origin :refs/tags/<tagname>
```

Переместить метку на новый коммит

```
$ git tag -fa <tagname> # на HEAD
$ git tag -fa <tagname> <commit>
```

Заключение