Git fast version control system

Автор:Новопашин Антон
E-mail: antonevane@gmail.com
Twitter: @anton_evane

План презентации

- Что такое Git?
- Ключевые особенности Git
- Репозиторий Git
- Работа с репозиторием
- Работа с branch
- Коллективная работа
- Ссылки на ресурсы

Что такое Git?

Git - программное обеспечение для управления версиями, разработанное Линусом Торвальдсом для использования в управлении разработкой ядра Linux®.

Git – распределенная система контроля версий (DVCS)

Ключевые особенности Git

- Ветвление делается быстро и легко.
- Поддерживается автономная работа; локальные фиксации изменений могут быть отправлены позже.
- Фиксации изменений атомарны и распространяются на весь проект.
- Каждое рабочее дерево в Git содержит хранилище с полной историей проекта.
- Ни одно хранилище Git не является по своей природе более важным, чем любое другое.
- Скорость работы

Репозиторий Git

Git хранит информацию в структуре данных называемой – репозиторий (**repository**).

Репозиторий хранит:

Набор **commit objects**

Набор ссылок на **commit objects** называемых **heads.**

Репозиторий хранится в той же директории, что и сам проект в поддиректории *.git*.

Основные отличия от систем с центральным репозиторием (например CVS, SVN):

Существует только одна директория *.git* в корневой директории проекта

Репозиторий хранится в файлах рядом с проектом

Не существует центрального репозитория

Репозиторий Git

Commit objects

Commit objects содержат:

- Набор файлов, отображающий состояние проекта в текущую точку времени
- Ссылки на родительские **commit objects**
- SHA1 имя – 40 символьная строка которая уникально идентифицирует **commit object**. Имя представляющее собой хэш является значимым аспектом *commit* (идентичные commits всегда будут иметь одинаковое имя)

Первый commit в проекте не имеет родительского объекта.

Идея контроля версий состоит в манипулировании графом **commit objects**.

Репозиторий Git

Heads

Head – ссылка на **commit object**. Каждый **head** имеет имя.

В каждом репозитории существует head называемый – *master*.

Репозиторий может содержать любое количество heads.

Выбранный head называют – “***current head***” он имеет синоним – “***HEAD***”

Работа с репозиторием

Создание репозитория выполняется командой: `git init`. После выполнения команды в папке проекта появиться директория .git.

Для выполнения commit необходимо выполнить следующее:

Сказать Git какие файлы необходимо добавить в commit данное действие выполняется командой `git add`. Если файлы не изменились с предыдущего commit то Git добавит их в commit автоматически.

Вызвать команду `git commit` которая создаст commit object.

Команда `git commit -a` добавит все изменившиеся файлы, но не новые файлы.

Работа с репозиторием

Полезные команды:

`git log` – показывает лог commits начиная с *HEAD*

`git status` – показывает какие файлы изменились между текущей стадией и *HEAD*. Файлы разделяются на 3 категории: новые файлы, измененные файлы, добавленные новые файлы

`git mv` – используется для перемещения или переименования файла

`git rm` – удаляет файл из репозитория не затрагивая рабочую копию

`gitk` – визуальная утилита для работы с репозиторием

Получение ссылок на commit выполняется следующим образом*:*

по SHA1 имя выполнив git log

по 1м символам SHA1 имени

используя head

Работа с branch

Создание branch выполняется следующей командой:

```
git branch branch_name <base_reference>
```

Переключение веток осуществляется командой:

```
git checkout head_name
```

Данная команда выполняет 2 функции:

- Указатель на HEAD **commit object** (head_name)
- Перезапись всех файлов в директории на соответствующие родительскому HEAD (^HEAD) и формирование нового commit.

Полезные сопутствующие команды:

- `git branch` – показывает список HEAD объектов
- `git diff [head1]..[head2]` – показывает изменения между 2мя HEAD
- `git log [head1]..[head2]` – показывает историю изменений

Коллективная работа

Копирование удаленного репозитория осуществляется командой `git clone repository_url`. Данная команда выполняет следующее:

Создает директорию проекта и инициализирует репозиторий

Копирует все *commit objects* и head ссылки в новый репозиторий

Добавляет удаленные head называемые *origin/[head_name]* соответствующие head в удаленном репозитории

Для работы с удаленной веткой локально необходимо выполнить следующую команду: `git branch [local_branch] [remote-branch]`

Получение изменений из удаленного репозитория выполняется командой: `git fetch [remote-repository-reference]`, по умолчанию это ссылка на *origin*. Данную команду в большинстве случаев заменяет `git pull`.

Коллективная работа

Добавление изменений в удаленный репозиторий выполняется командой `git push [remote-repository-reference] [remote-head-name]`.

После вызова команды происходит следующее:

В удаленный репозиторий добавляются новые *commit object*.

Устанавливается head в удаленном репозитории на тот же commit.

Если выполняется `git push` без аргументов то отправляются все ветки за которыми было установлено слежение.

Ссылки на ресурсы

Бесплатный хостинг репозиториев для open source проектов:

github.com

Официальный сайт

Git: http://git-scm.com/