

Основы работы с git

Кирилл Краснощёков, МГТНИИП

5 апреля 2018 г.

Outline

- 1 Введение
 - Что такое системы управления версиями
 - Термины
- 2 Как пользоваться
 - Установка, настройка, запуск
 - Как делать резервные копии
 - Как несколькоим людям работать над одним проектом
 - Как взять обновления с сервера

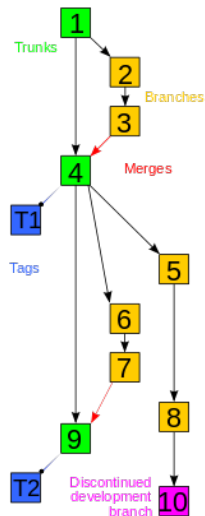
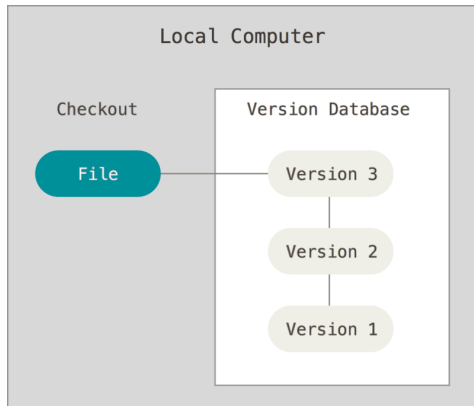
1 Введение

- Что такое системы управления версиями
- Термины

2 Как пользоваться

- Установка, настройка, запуск
- Как делать резервные копии
- Как несколькими людям работать над одним проектом
- Как взять обновления с сервера

Что такое система управления версиями



Зачем это нужно

- не терять результаты работы
- работать совместно

1 Введение

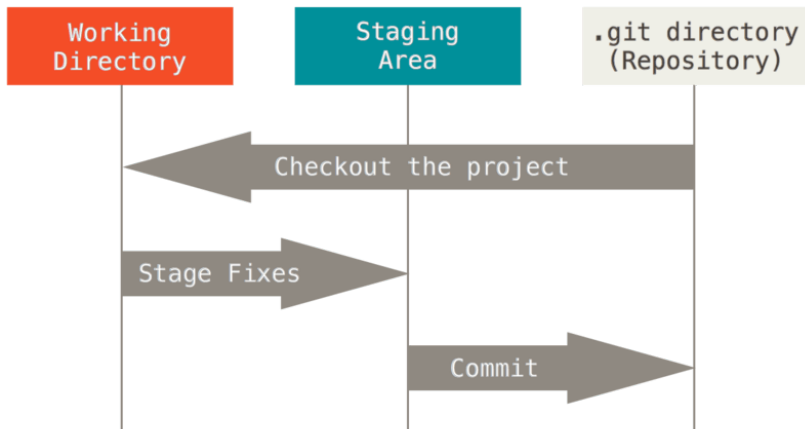
- Что такое системы управления версиями
- Термины

2 Как пользоваться

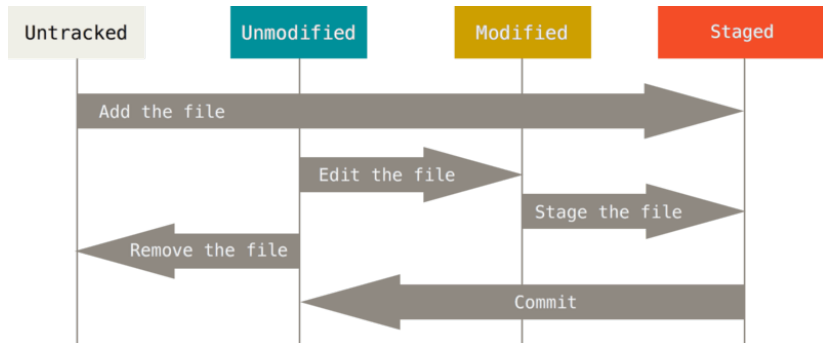
- Установка, настройка, запуск
- Как делать резервные копии
- Как несколькими людьми работать над одним проектом
- Как взять обновления с сервера

Репозиторий

Репозиторий = БД изменений в проекте



Файлы проекта



- 1 Введение
 - Что такое системы управления версиями
 - Термины

- 2 Как пользоваться
 - Установка, настройка, запуск
 - Как делать резервные копии
 - Как несколькими людьми работать над одним проектом
 - Как взять обновления с сервера

Установка

- 1 скачать последнюю версию
- 2 установить (без админских прав)

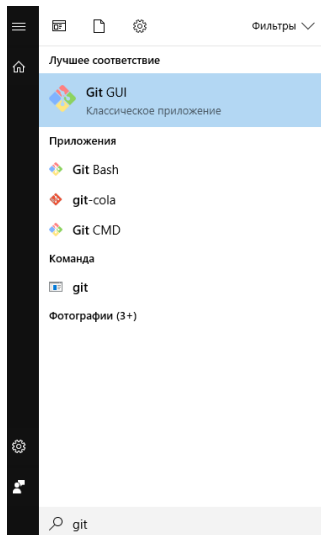
Настройка

`git config`

```
$ git config --global user.name "John Doe"
```

```
$ git config --global user.email "john@example.com"
```

Запуск



- git bash
- git GUI

- 1 Введение
 - Что такое системы управления версиями
 - Термины

- 2 Как пользоваться
 - Установка, настройка, запуск
 - Как делать резервные копии
 - Как несколькими людьми работать над одним проектом
 - Как взять обновления с сервера

Как делать резервные копии

- 1 создать репозиторий
- 2 добавить файлы в репозиторий
- 3 поработать и записать изменения в репозиторий

И повторять шаг 3.

1. Создать репозиторий

```
git init
```

```
$ mkdir project_2
```

```
$ cd project_2
```

```
$ git init
```

```
Initialized empty Git repository in project_2/.git/
```

Осмотреться

```
git status
```

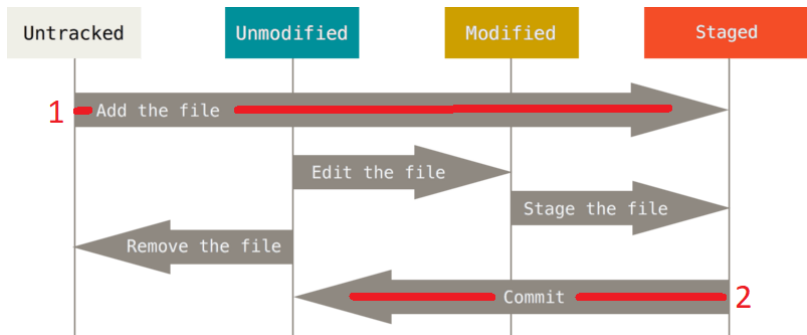
```
$ git status  
On branch master
```

```
Initial commit
```

```
nothing to commit (create/copy files  
and use "git add" to track)
```

- делайте в любой непонятной ситуации

2. Добавить файлы в репозиторий



- 1 Поместить файлы в буферную область: `git add my_file.txt`
- 2 Отправить в репозиторий: `git commit`

- В начале:

```
$ git status
```

```
On branch master
```

```
Initial commit
```

Untracked files:

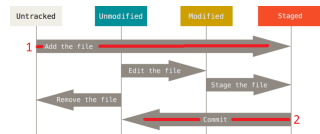
(use "git add <file>..." to include
in what will be committed)

```
my_file.txt
```

nothing added to commit

but untracked files present

(use "git add" to track)



- добавляем файл в буферную область:

```
git add
```

```
$ git add my_file.txt
```

```
$ git status
```

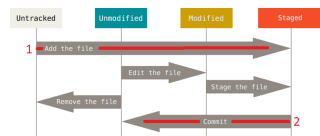
```
On branch master
```

```
Initial commit
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
new file:   my_file.txt
```



- отправляем изменения в репозиторий:

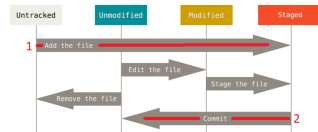
```
git commit
```

```
$ git commit -m "Very important change"
[master (root-commit) 2642eb5] Very important change
1 file changed, 12 insertions(+)
 create mode 100644 my_file.txt
```

```
$ git status
```

```
On branch master
```

```
nothing to commit, working tree clean
```



👉 Комментируйте осмысленно!

Так не надо:

	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAANDS	2 HOURS AGO

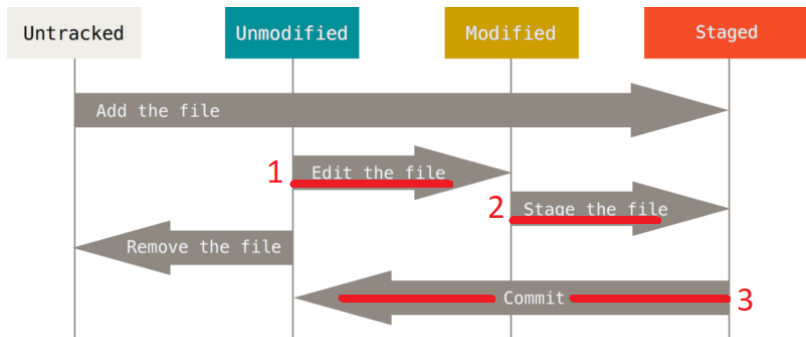
AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

👉 Комментируйте осмысленно!

Так лучше:

Commits in master		
6e343c0	* master origin/master origin/HEAD fix GTFS for MGT add print value for stops_	Titov_AY 2 weeks
51539ca	* [fix] sort stop_sequence for gtfs, import openpyxl	Titov_AY 2 weeks
a5f8ebe	* Исправлена ошибка с выбором трипов для привязки	sachasach 3 weeks
3dd6e66	* Добавлены комментарии для отслеживания прогресса	sachasach 3 weeks
b059eda	* Исправлена опечатка	sachasach 3 weeks
4c79f2d	* Добавлена установка pyuulm	sachasach 3 weeks
58c8381	* Убраны баги и tqdm	sachasach 3 weeks
58ff295	* Исправлен баг с названием переменной и базы данных	sachasach 3 weeks
8ad3fd4	* нужно изменять дату в тексте блокнота, или вынести её в конфиг	sachasach 3 weeks
17ef05f	* Merge branch 'master' of gitlab.com:mgtniip_analytics/Bd_client	sachasach 3 weeks
\		
b44ac8a	* origin/krvkir/split-into-modules [docs] Updated `hacking.md`	krvkir 3 weeks
283c4d5	* [docs] Updated architecture graph	krvkir 3 weeks
0a886ca	* [repo] Move large .png's into LFS storage	krvkir 3 weeks
8317920	* [fix] Returned database name into window title	krvkir 3 weeks
cb08752	* [refactor] Solved circular dependency by moving engine to `dbapi.engine`	krvkir 3 weeks
086ab30	* [refactor] Take username for log from the config	krvkir 3 weeks
52949ce	* [refactor] Split `dbapi/_init_.py` into `stops.py` and `log.py`	krvkir 3 weeks
c8da162	* [refactor] Take db credentials from config	krvkir 3 weeks
2ba4ff7	* [refactor] Added config and config reader	krvkir 3 weeks
7c658f4	* Merge branch 'master' into krvkir/split-into-modules	krvkir 3 weeks
\		
e47dd70	* [fix] Fixed MGTapI url & env name in `install.bat`	Titov_AY 3 weeks
5d85d2e	* [docs] Improved `hacking.md`	krvkir 3 weeks
4070884	* [docs] Fixed a typo in `hacking`	krvkir 3 weeks
d918f2b	* [docs] Added link to `hacking.md` into `README.md`	krvkir 3 weeks
b55dea3	* [docs] Added examples into `hacking.md`	krvkir 3 weeks
22104ae	* [add] Architecture: .graphml for `yed` editor + image in `hacking.md`	krvkir 3 weeks
a1d6ef7	* [docs] Added manual for developers	krvkir 3 weeks
/		
adb61d8	* [fix] Renamed virtualenv from `bdclient` to `dbclient`	krvkir 3 weeks
5fceb6f	* [fix,docs] Fixed `install.bat` and `README.md`	krvkir 3 weeks
ac766d0	* Merge branch 'krvkir/split-into-modules'	krvkir 3 weeks
\		
6371170	* [add] Install & run desktop scripts	krvkir 3 weeks

3. Записать изменения



- 1 Поработать
- 2 Поместить изменённые файлы в буферную область: `git add my_file.txt`
- 3 Отправить в репозиторий: `git commit`

Вывести историю

```
git log
```

```
$ git log
```

```
commit 2642eb5780a454a09bac4ef9113fc029cff27674
```

```
Author: krvkir <krvkir@gmail.com>
```

```
Date: Thu Apr 5 12:31:19 2018 +0300
```

```
Very important change
```

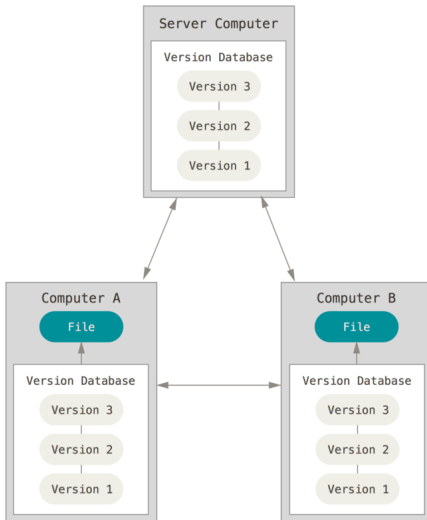

1 Введение

- Что такое системы управления версиями
- Термины

2 Как пользоваться

- Установка, настройка, запуск
- Как делать резервные копии
- Как несколькоим людям работать над одним проектом
- Как взять обновления с сервера

Центральный сервер

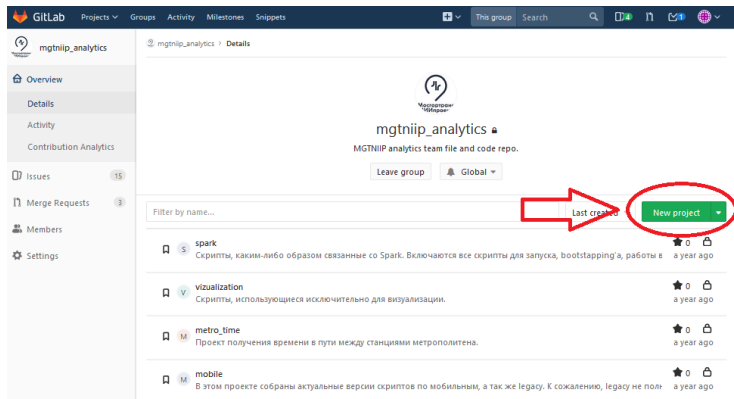


- github.com
- gitlab.com
- sourceforge.org
- ...
- внутренний сервер


Как работать с сервером

- ❶ создать репозиторий на сервере
- ❷ клонировать репозиторий с сервера в локальную папку: `git clone`
- ❸ работать, как обычно
- ❹ отправить изменения на сервер: `git push`

1. Создать репозиторий



1. Создать репозиторий

 Projects Groups Activity Milestones Snippets Search

Projects

New project

A project is where you house your files (repository), plan your work (issues), and publish your documentation (wiki), [among other things](#).

All features are enabled for blank projects, from templates, or when importing, but you can disable them afterward in the project settings.

To only use CI/CD features for an external repository, choose **CI/CD for external repo**.

Tip: You can also create a project from the command line. [Show command](#)

Blank projectCreate from templateImport projectCI/CD for external repo

Project path

Project name

Project description (optional)

Visibility Level

Create projectCancel

https://gitlab.com/1mgtntip_analytics2my_shining_project

Want to house several dependent projects under the same namespace? [Create a group](#)

Description of my shining project.
Maybe few more lines.
Yeah!

Private

Internal

Public

Project access must be granted explicitly to each user.

This project cannot be internal because the visibility of `mgtntip_analytics` is private. To make this project internal, you must first [change the visibility](#) of the parent group.

This project cannot be public because the visibility of `mgtntip_analytics` is private. To make this project public, you must first [change the visibility](#) of the parent group.

1. Создать репозиторий

The screenshot shows the GitLab web interface. At the top, there's a navigation bar with 'GitLab' logo and links for 'Projects', 'Groups', 'Activity', 'Milestones', and 'Snippets'. Below this, a sidebar on the left contains a list of project features: 'Overview', 'Details', 'Activity', 'Cycle Analytics', 'Registry', 'Issues', 'Merge Requests', 'CI / CD', 'Wiki', 'Snippets', and 'Settings'. The main content area displays the details of a project named 'my_shining_project'. A blue banner at the top of the main area states 'Project 'my_shining_project' was successfully created.' Below this, the project name 'my_shining_project' is shown with a 'master' branch. A red circle highlights the repository URL: 'git@gitlab.com:krvklr/my_shining_'. Below the URL, it says 'The repository for this project is empty'. Further down, there are instructions on how to push files using command line instructions, and a note about protected branches. At the bottom, there's a section about enabling Auto DevOps (Beta) and adding a Kubernetes cluster.

2. Клонировать репозиторий с сервера

```
git clone
```

```
$ git clone https://github.com/krvkir/notes_dev_culture.git
Cloning into 'notes_dev_culture'...
remote: Counting objects: 9, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 9 (delta 2), reused 9 (delta 2), pack-reused 0
Unpacking objects: 100% (9/9), done.
```

```
$ cd notes_dev_culture/
```

```
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working tree clean
```

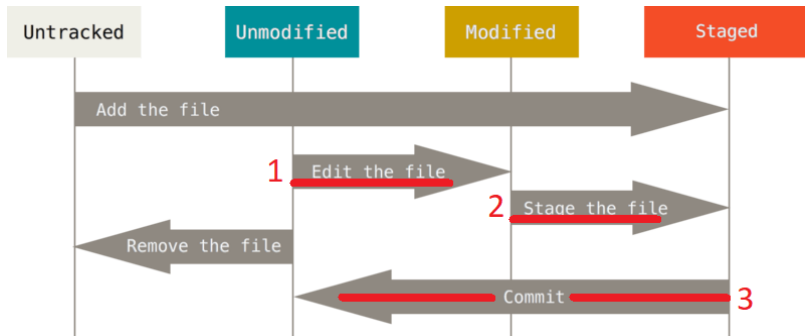
Список серверов

```
git remote
```

```
$ git remote  
origin
```

```
$ git remote -v  
origin  git@github.com:krvkir/notes_dev_culture.git (fetch)  
origin  git@github.com:krvkir/notes_dev_culture.git (push)
```


3. Работать, как обычно



- ❶ Поработать
- ❷ `git add`
- ❸ `git commit`

4. Отправить изменения на сервер

```
git push
```

```
git push origin master
```

- ввести логин и пароль от сервера

- 1 Введение
 - Что такое системы управления версиями
 - Термины

- 2 Как пользоваться
 - Установка, настройка, запуск
 - Как делать резервные копии
 - Как несколькими людьми работать над одним проектом
 - Как взять обновления с сервера

Как взять обновления с сервера

- 1 Скопировать репозиторий с сервера: `git clone`
- 2 Обновить: `git pull`

```
$ git pull origin master
```

Материалы

- Pro Git (книга)
- Git Visual (интерактивная обучалка)