# GIT

**Документация**

<http://ru.wikipedia.org/wiki/GIT>  
[Git How To](http://githowto.com/ru/)  
[GIT man page](http://linux.die.net/man/1/git)

Оглавление документа

[Документация](https://wiki.yandex-team.ru/belyan/git#dokumentacija)

[Настройка](https://wiki.yandex-team.ru/belyan/git#nastrojjka)

[Алиасы](https://wiki.yandex-team.ru/belyan/git#aliasy)

[Основные команды](https://wiki.yandex-team.ru/belyan/git#osnovnyekomandy)

[Операции с метками](https://wiki.yandex-team.ru/belyan/git#operaciismetkami)

[Операции с ветками](https://wiki.yandex-team.ru/belyan/git#operaciisvetkami)

[Работа с внешними репозиториями](https://wiki.yandex-team.ru/belyan/git#rabotasvneshnimirepozitorijami)

[Сценарии](https://wiki.yandex-team.ru/belyan/git#scenarii)

[Клонировать внешний репозиторий](https://wiki.yandex-team.ru/belyan/git#klonirovatvneshnijjrepozitorijj)

[Связь локальной ветки с внешним репозиторием](https://wiki.yandex-team.ru/belyan/git#svjazlokalnojjvetkisvneshnimrepozitoriem)

[Добавить внешний репозиторий](https://wiki.yandex-team.ru/belyan/git#dobavitvneshnijjrepozitorijj)

[Добавить файл в игнор](https://wiki.yandex-team.ru/belyan/git#dobavitfajjlvignor)

[Создать метку](https://wiki.yandex-team.ru/belyan/git#sozdatmetku)

[Переименовать ствол в старую ветку, а новую ветку сделать стволом](https://wiki.yandex-team.ru/belyan/git#pereimenovatstvolvstarujuvetkuanovujuvetkusdelatstvolom)

[Внести вклад в открытый проект](https://wiki.yandex-team.ru/belyan/git#vnestivkladvotkrytyjjproekt)

[История изменений](https://wiki.yandex-team.ru/belyan/git#istorijaizmenenijj)

[Сравнение команд GIT и SVN](https://wiki.yandex-team.ru/belyan/git#sravneniekomandgitisvn)

[Перенос репозитория из SVN](https://wiki.yandex-team.ru/belyan/git#perenosrepozitorijaizsvn)

**Настройка**

Установка имени и электронной почты

|  |  |
| --- | --- |
| 1  2 | git config --global user.name "Yuriy Belyakov"  git config --global user.email belyan@yandex-team.ru |

Текстовый редактор по умолчанию

git config --global core.editor vim

Постраничный вывод длинных команд

git config --global core.pager more

Параметры установки окончаний строк

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | git config --global core.autocrlf false  # true - запретить записывать CRLF в репозиторий  # input - конвертация CRLF в LF только при коммитах  # false - разрешить записывать CRLF в репозиторий  git config --global core.safecrlf true |

Утилита сравнения

git config --global merge.tool vimdiff

Цвета

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | # Для Git версии 1.5.5 и выше  git config --global color.ui true  # Для более старых версий  git config --global color.branch true  git config --global color.diff true  git config --global color.interactive true  git config --global color.status true |

Проверка настроек

git config --list

или

cat ~./gitconfig

**Алиасы**

Добавьте следующее в файл .gitconfig в вашем $HOME каталоге.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | [alias]  co = checkout  ci = commit  st = status  br = branch  hist = log --pretty=format:'%h %ai | %an <%ae> | %s %b' --max-count=25  histg = log --pretty=format:'%h %ai | %an <%ae> | %s' --graph --max-count 25  info = config --list  type = cat-file -t  dump = cat-file -p |

**Основные команды**

|  |  |
| --- | --- |
| git init | создать репозиторий из текущего каталога |
| git status | проверить состояние |
| git diff | посмотреть локальные изменения |
| git checkout -- <file> | отменить локальные изменения |
| git stash | спрятать локальные изменения |
| git stash apply | вернуть спрятанные изменения |
| git add/mv/rm <file> | проиндексировать изменения |
| git diff --staged | посмотреть проиндексированные изменения |
| git reset HEAD <file> | отменить индексацию изменений |
| git commit -m <comment> | закоммитить изменения |
| git commit --amend | изменить последний коммит |
| git revert HEAD | отменить последний коммит |
| git revert <hash> | отменить произвольный коммит |
| git reset --hard <hash> | удалить произвольный коммит |

**Операции с метками**

|  |  |
| --- | --- |
| git tag | посмотреть метки |
| git tag <tag> | создать метку |
| git tag -a <tag> -m <message> | создать аннотированную метку |
| git tag -s <tag> -m <message> | создать подписанную метку (с помощью GPG) |
| git show <tag> | посмотреть данные метки |
| git checkout <tag> | переключиться на метку |
| git tag -d <tag> | удалить метку |

**Операции с ветками**

|  |  |
| --- | --- |
| git branch | посмотреть ветки |
| git branch -v | посмотреть последний коммит в каждой ветке |
| git branch <branch> | создать ветку |
| git checkout <branch> | переключиться в ветку |
| git checkout -b <branch> | создать ветку и сразу переключиться в нее |
| git merge <branch> | слияние с веткой |
| git rebase <branch> | перебазирование ветки |
| git branch -m <branch> <new\_branch> | переименовать ветку |
| git branch -d <branch> | удалить ветку |
| git stash branch <branch> | создать ветку из спрятанных изменений |

**Работа с внешними репозиториями**

В примерах в качестве внешнего репозитория используется origin

|  |  |
| --- | --- |
| git remote -v | посмотреть внешние репозитории |
| git remote show origin | получить подробную информацию о внешнем репозитории |
| git remote add origin <url> | добавить внешний репозиторий |
| git remote rename origin <name> | переименовать внешний репозиторий |
| git remote rm origin | удалить внешний репозиторий |
| git branch -a | посмотреть внешние ветки |
| git branch --track <branch> origin/<branch> | создать ветку, связанную с внешней |
| git fetch origin | извлечь изменения из внешнего репозитория |
| git merge origin/<branch> | слияние с внешней веткой |
| git pull | извлечение и слияние изменений |
| git push origin <branch> | отправка изменений |
| git checkout -b <branch> origin/<branch> | создать локальную ветку на основе удаленной ветки |
| git push origin :<branch> | удалить ветку из внешнего репозитория |

**Сценарии**

**Клонировать внешний репозиторий**

git clone git+ssh://git.yandex.ru/market/pda.git market-pda

**Связь локальной ветки с внешним репозиторием**

|  |  |
| --- | --- |
| 1  2 | git config branch.master.remote origin  git config branch.master.merge refs/heads/master |

*или*

git branch --set-upstream master origin/master

**Добавить внешний репозиторий**

git submodule add git+ssh://git.yandex.ru/market/common.git common.hid

**Добавить файл в игнор**

touch .gitignore

**Создать метку**

|  |  |
| --- | --- |
| 1  2 | git tag v1.0-16 fc2c776 # создаем метку для коммита с указанным хешом  git push origin v1.0-16 # отправляем метку на удаленный репозиторий |

**Переименовать ствол в старую ветку, а новую ветку сделать стволом**

Переименовываем в локальном репозитории master в ветку old, а ветку new в master

|  |
| --- |
| git br -m master old  git br -m new master |

Удаляем старую версию master из удаленного репозитория

git push origin :master

Связываем новую и старую версии master с удаленным репозиторием

|  |
| --- |
| git push origin master:refs/heads/master  git push origin old:refs/heads/old |

Удаляем ветку new из удаленного репозитория

git push origin :new

**Внести вклад в открытый проект**

Сначал нужно склонировать основной репозиторий, создать тематическую ветку для одного или нескольких патчей, которые вы собираетесь внести в проект, и выполнить свою работу в ней.

|  |  |
| --- | --- |
| 1  2  3  4 | git clone <url> project  cd project  git checkout -b featureA  git commit # (выполнение работы) |

**pull request**  
Если вы закончили работу со своей веткой и готовы поделиться наработками с мейнтейнерами, перейдите на страницу исходного проекта и нажмите кнопку «Fork», создав таким образом свою собственную копию проекта доступную на запись. Вам нужно отправить свои наработки в этот репозиторий.

|  |  |
| --- | --- |
| 1  2 | git remote add myfork <url>  git push myfork featureA |

Когда ваши наработки будут отправлены в ваш форк, вам нужно будет послать уведомление мейнтейнеру. Его часто называют запросом на включение (pull request), вы можете либо сгенерировать его через сайт — на GitHub’е есть кнопка «pull request», автоматически уведомляющая мейнтейнера, либо выполнить команду git request-pull и вручную отправить её вывод по почте мейнтейнеру.

git request-pull origin/master myfork

**patch**  
Вместо того, чтобы создавать ответвление (fork) от проекта и отправлять наработки в свой собственный репозиторий с правами на запись, вы генерируете e-mail версию каждой серии коммитов и отправляете её в список рассылки для разработчиков.

git format-patch -M origin/master

**История изменений**

[gitk](http://gitk.sourceforge.net/) - графический инструмент для визуализации истории коммитов

git log

**Параметры**

|  |  |
| --- | --- |
| -n | Показать последние n коммитов |
| -p | Выводит патч (заплатку/diff) внесенный каждым коммитом |
| --since  или  --after | Ограничить коммиты теми, которые сделаны после указанной даты |
| --until  или  --before | Ограничить коммиты теми, которые сделаны до указанной даты |
| --author | Показать только те коммиты, автор которых соответствует указанной строке |
| --stat | Выводит статистику по файлам измененным в каждом коммите |
| --shortstat | Отображает только строку с changed/insertions/deletions от вывода команды  --stat |
| --name-only | Выводит список измененных файлов после каждого коммита |
| --name-status | Выводит список файлов вместе с информацией о добавлении/изменении/удалении |
| --abbrev-commit | Выводит только первые несколько символов контрольной суммы SHA-1 вместо всех 40 |
| --relative-date | Выводит дату в относительном формате (например, “2 недели назад”) |
| --graph | Выводит ASCII граф истории ветвлений и слияний рядом с выводом лога |
| --pretty=format | Выводит коммиты в альтернативном формате %an - Имя автора %ae - Электронная почта автора %ad - Дата автора (формат соответствует параметру  --date  ) %ar - Дата автора, относительная %h - Сокращенный хеш коммита %s - Комментарий |

**Примеры**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | git log -p -2 # разница (diff) двух последних коммитов  git log --stat # краткая статистика по каждому коммиту  git log --pretty=oneline # вывод коммитов в одну строку  git log --since=2.weeks # список коммитов, сделанных за последние две недели  git log --author=belyan # коммиты определенного автора  git log --pretty=format:"%h %s" --graph  git log --pretty=format:'%h %ad | %an | %s' --since='7 days ago' --date=short |

**Сравнение команд GIT и SVN**

|  |  |
| --- | --- |
| **SVN** | **GIT** |
| svn co svn+ssh://... | git clone git+ssh://... |
| svn info | git config --list |
| svn switch | git checkout |
| svn up | git pull |
| svn st | git status |
| svn remove | git remove |
| svn add | git add |
| svn diff | git diff |
| svn ci | git commit + git push |
| debcommit | debcommit -a |

**Перенос репозитория из SVN**

1. Чекатуим репозиторий из SVN

svn co svn+ssh://svn.yandex.ru/market/verstka/mobile market-mobile

2. Получаем из changelog список логинов

|  |
| --- |
| cd market-mobile  svn log -q | cut -s -d'|' -f2 | sort | uniq > logins |

3. На основе списка логинов составляем список пользователей в формате:  
belyan = Yuri Belyakov <belyan@yandex-team.ru>

\* фамилию и имя пользователя можно взять со [стаффа](http://staff.yandex-team.ru/belyan/en/)

Либо можно воспользоваться [скриптом](https://git.yandex.ru/kbakba/kbakba.git/?a=blob_plain;f=authors-prog.py;hb=HEAD), который делает это автоматически

export PASSPORT\_SESSION\_ID=<Session\_id>

cat logins | xargs -n1 ~/projects/git-svn/authors-prog.py > users

export PASSPORT\_SESSION\_ID=

\* где <Session\_id> - кука Session\_id в домене **yandex-team.ru** (нужна для доступа в center.yandex-team.ru)

4. Выполняем миграцию с помощью git svn

|  |
| --- |
| mkdir ~/projects/market-mobile-gitsvn && cd ~/projects/market-mobile-gitsvn  git svn init --no-metadata --stdlayout svn+ssh://svn.yandex.ru/market/verstka/mobile  git config svn.authorsfile ~/projects/market-mobile/users  git svn fetch |

5. Импорт svn:ignore

git svn show-ignore -itrunk >> .gitignore

6. Создаем "чистый" git-репозиторий

cd ..

git clone market-mobile-gitsvn market-mobile-git

7. Привязка нового репозитория к git.yandex.ru

|  |  |
| --- | --- |
| 1  2 | git remote set-url origin git+ssh://git.yandex.ru/market/mobile.git # меняем путь к origin  git push origin master # отправляем все в новую удаленную ветку master |

8. Подключить внешние репозитории

git submodule add git+ssh://git.yandex.ru/market/common.git common.hid