

Публикация на тему

Git

git, git rebase, git merge

Автор

[Михалькевич Александр Викторович](#)

Публикация

Наименование Git

Автор А.В.Михалькевич

Специальность git, git rebase, git merge,

Анотация

Anotation in English

Ключевые слова

Количество символов 7785

Содержание

[Введение](#)

1 [Основы GIT](#)

2 [Merge request на практике](#)

[Заключение](#)

[Список использованных источников](#)

[Приложения](#)

Введение

1 Основы GIT

Git — распределённая система управления версиями.

Скачать git можно по этой ссылке <https://git-scm.com/downloads>

Благодаря Git и системы ветвления у разработчиков имеется возможность работать в команде над одним проектом.

Обычно разработка начинается с клонирования репозитория, либо создания репозитория в

папке с проектом. Далее, при необходимости, создаются ветки, в которых хранится разное состояние проекта, также часто используются команды `git pull` - скачать изменения из удалённого git-репозитория и `git push` - залить изменения в удалённый репозиторий.

Предположим, у нас имеется git-репозиторий с проектом, который необходимо развернуть локально - https://github.com/mikhalkevich/laravel_blog

```
cd domain // перешли в папку, где будем разворачивать проект
git clone https://github.com/mikhalkevich/laravel_blog
```

Далее открываем проект в своей любимой IDE и решаем поставленные задачи. После чего снова в консоли:

```
git add *
git commit -m "first commit"
git push
```

Другие часто используемые команды git:

GIT Cheatsheet



```

// Repository
git init // Initialize a new Git repo
git clone <repo-url> // Clone a repo from a URL

// Basics
git status // Show changes status
git add <file> // Add changes to staging
git commit -m "Message" // Commit changes with a message
git log // View commit history

// Branching
git branch // List branches
git branch <branch-name> // Create a new branch
git checkout <branch-name> // Switch to a branch
git merge <branch-name> // Merge changes from a branch
git branch -d <branch-name> // Delete a branch

// Remote Repositories
git remote // List remotes
git remote add <name> <url> // Add a remote
git push <remote> <branch> // Push changes to a remote
git pull <remote> <branch> // Pull changes from a remote

// Undoing Changes
git pull // Fetch and merge changes
git fetch // Fetch changes without merging
git reset --hard HEAD // Discard changes
git revert <commit-hash> // Revert changes in a commit
```



Ram Maheshwari

rammcodes



2 Merge request на практике

Итак, мы клонировали проект, и начинаем работать над задачей.

По умолчанию исходный код находится в ветке `master`. Поэтому прежде чем приступить к задаче, необходимо создать ветку под текущую задачу. Название ветки имеет значение:

- префикс `feat/` означает `feature`, он всегда должен присутствовать в начале названия;
- номер задачи, который идёт сразу после префикса и
- через знак `"_"` нижнее подчёркивание описываем одним/двумя словами на английском языке суть задачи.

Пример:

```
git branch feat/4546_form
git checkout feat/4546_form
```

Теперь можно приступать к задаче. Вносим необходимые изменения в проект. Например, в файл `test.php` добавили код

```
phpinfo();
```

Примечание. Если приступили к задаче и забыли переключиться на нужную ветку - не проблема. Это можно сделать и на текущем этапе. Главное следить за тем, чтобы локально была установлена текущая версия проекта (почаще делаем `git pull` из ветки `master`), иначе могут возникнуть конфликты с `merge` (слиянии веток). Если в удалённом репозитории настроен `merge request`, то попытка залить в `master` ни к чему не приведёт.

Далее необходимо залить текущую ветку с изменениями на удалённый репозиторий. Команды:

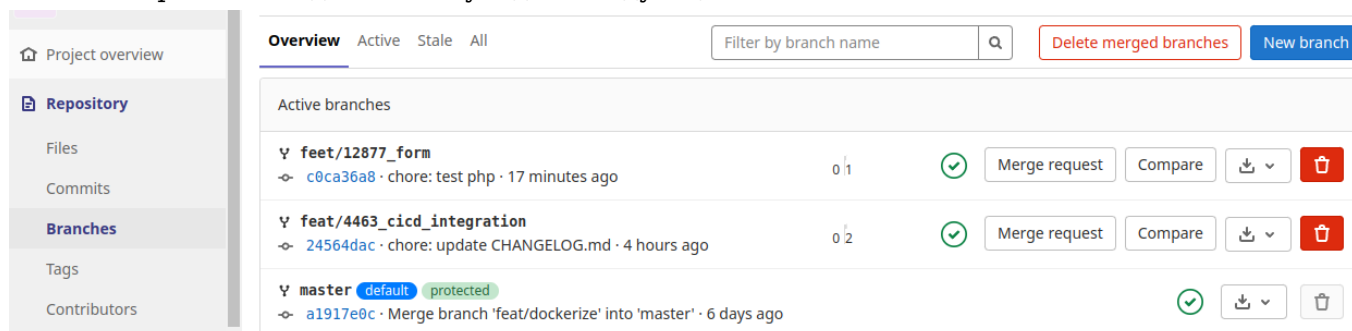
```
git add *
git commit -m "chore: test php"
```

Примечание. Если задача ещё выполнена, то в начале коммита необходимо прописать `chore`, если задача выполнена - то, в `message` пишем что угодно: что делали, то и пишем (желательно использовать английские буквы).

Теперь необходимо изменения из ветки задачи залить в `master` - тогда мы сможем протестировать задачу на удалённом сервере. Для этого необходимо авторизоваться в [gitlab](#) и перейти на страницу с ветками проекта.

Repository -> branches

Если всё правильно сделали, то увидим следующее:



The screenshot shows the GitLab web interface. On the left, the 'Repository' sidebar has 'Branches' selected. The main content area is titled 'Active branches' and shows a list of three branches:

Branch Name	Commit Hash	Commit Message	Time Ago	Branch Status	Actions
feat/12877_form	c0ca36a8	chore: test php	17 minutes ago	0 1	Merge request, Compare, Download, Delete
feat/4463_cicd_integration	24564dac	chore: update CHANGELOG.md	4 hours ago	0 2	Merge request, Compare, Download, Delete
master (default, protected)	a1917e0c	Merge branch 'feat/dockerize' into 'master'	6 days ago		Merge request, Compare, Download, Delete

Далее необходимо сделать Merge request. Нажимаем на кнопку Merge request своей ветки и переходим на соответствующую страницу.

Примечание. Merge request можно сделать также перейдя по соответствующей

ссылке в репозитории, но тогда в выпадающем списке веток необходимо выполнить `compare branches` для своей ветки, которую предварительно нужно выбрать из выпадающего списка.

Если всё правильно сделали, увидим следующее:

New merge request

From `feat/12877_form` into `master` [Change branches](#)

Title

Start the title with `Draft:` to preview

Меняем Title на следующий:

[12877] Form adding

Находясь на этой же странице, чуть ниже, выбираем Assignee (текущий пользователь, т.е. тот, кто делает задачу), Reviewer (тот, кто будет проверять задачу) и нажимаем Merge request

Assignee

Павел Салтыков

Reviewer

Павел Салтыков

Milestone

Milestone

Labels

Labels

Merge options

☒ Delete source branch when merge request is accepted.

☐ Squash commits when merge request is accepted. [?](#)

Please review the [contribution guidelines](#) for this project.

Create merge request

Cancel

Merge request создан. Но если задача не доделана, помечаем её как черновик Mark as draft

Open

Created just now by



Павел Салтыков

Developer

Edit

Mark as draft



chore: test php

Overview 0

Commits 1

Pipelines 1

Changes 1



Request to merge [feet/12877](#) form [feet/12877](#)
into [master](#)

Open in Web IDE

Check out branch



Merge request в состоянии draft позволяет отредактировать Title и Description и проверить какие изменения были внесены в код.

Если задача выполнена - переводим Merge request в состояние Mark as ready. После чего возвращаемся в локальный репозиторий и создаём ещё один комит, но уже с другим названием: `feet: [4546](href_link_to_task)`, где в квадратных скопках - название, в круглых - ссылка на задачу.

Заключение

Список использованных источников

Приложения