

Git - część 3

Podstawowa obsługa systemu



Cześć!

Daniel Kossakowski

DevOps Engineer @ CloudPanda.io

1. Tagi

Tagi O co chodzi?

- Tag to czytelny znacznik w historii
- `git tag` - lista znaczników
- `git tag 'nazwa'` - nowy znacznik na aktualnym commicie
- `git push origin --tags` - przesłanie znaczników do serwera
- Popularna praktyka: Wersjonowanie kodu przy pomocy tagów

Tagi Tworzenie

```
linux@iSA ~ $ git tag
linux@iSA ~ $ git tag 1.0
linux@iSA ~ $ git tag
1.0
linux@iSA ~ $ git log
commit a7a8d41234f515fa7a5e0f27cf10f25ec8fa7997 (HEAD -> master, tag: 1.0)
Author: Daniel Kossakowski <daniel@krolnet.pl>
Date: Sun Nov 18 23:03:38 2018 +0100
```

Add test file

Tagi Przełączanie na tag

- `git checkout <tag>`

```
linux@iSA ~ $ git checkout 1.0
```

```
Note: checking out '1.0'.
```

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using `-b` with the checkout command again. Example:

```
git checkout -b <new-branch-name>
```

```
HEAD is now at a7a8d41 Add test file
```

2.

Git Flow - rozszerzenie

Git Flow Dodatek do gita

- Źródła: <https://github.com/nvie/gitflow>
- Instalacja:
 - Ze źródeł
 - Z oficjalnych repozytoriów

```
linux@iSA ~ $ sudo apt-get update
(...)
linux@iSA ~ $ sudo apt-get install git-flow
Czytanie list pakietów... Gotowe
Budowanie drzewa zależności
Odczyt informacji o stanie... Gotowe
(...)
linux@iSA ~ $ git flow version
1.11.0 (AVH Edition)
```


Git Flow Włączenie obsługi

```
linux@iSA ~ $ git flow init
```

Which branch should be used for bringing forth production releases?

- develop
- master

Branch name for production releases: [master]

Which branch should be used for integration of the "next release"?

- develop

Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?

Feature branches? [feature/]

Bugfix branches? [bugfix/]

Release branches? [release/]

Hotfix branches? [hotfix/]

Support branches? [support/]

Version tag prefix? []

Hooks and filters directory? [/tmp/jjdd5-materialy-git/.git/hooks]

Git Flow Pierwszy feature

```
linux@iSA ~ $ git flow feature start ISA-123  
Switched to a new branch 'feature/ISA-123'
```

Summary of actions:

- A new branch 'feature/ISA-123' was created, based on 'develop'
- You are now on branch 'feature/ISA-123'

Now, start committing on your feature. When done, use:

```
git flow feature finish ISA-123  
linux@iSA ~ $ git branch  
develop  
* feature/ISA-123  
master
```

Git Flow Zakończenie pracy

```
linux@iSA ~ $ git flow feature finish ISA-123
Switched to branch 'develop'
Updating 852c180..b0acee5
Fast-forward
 feature.file | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 feature.file
Deleted branch feature/ISA-123 (was b0acee5).
```

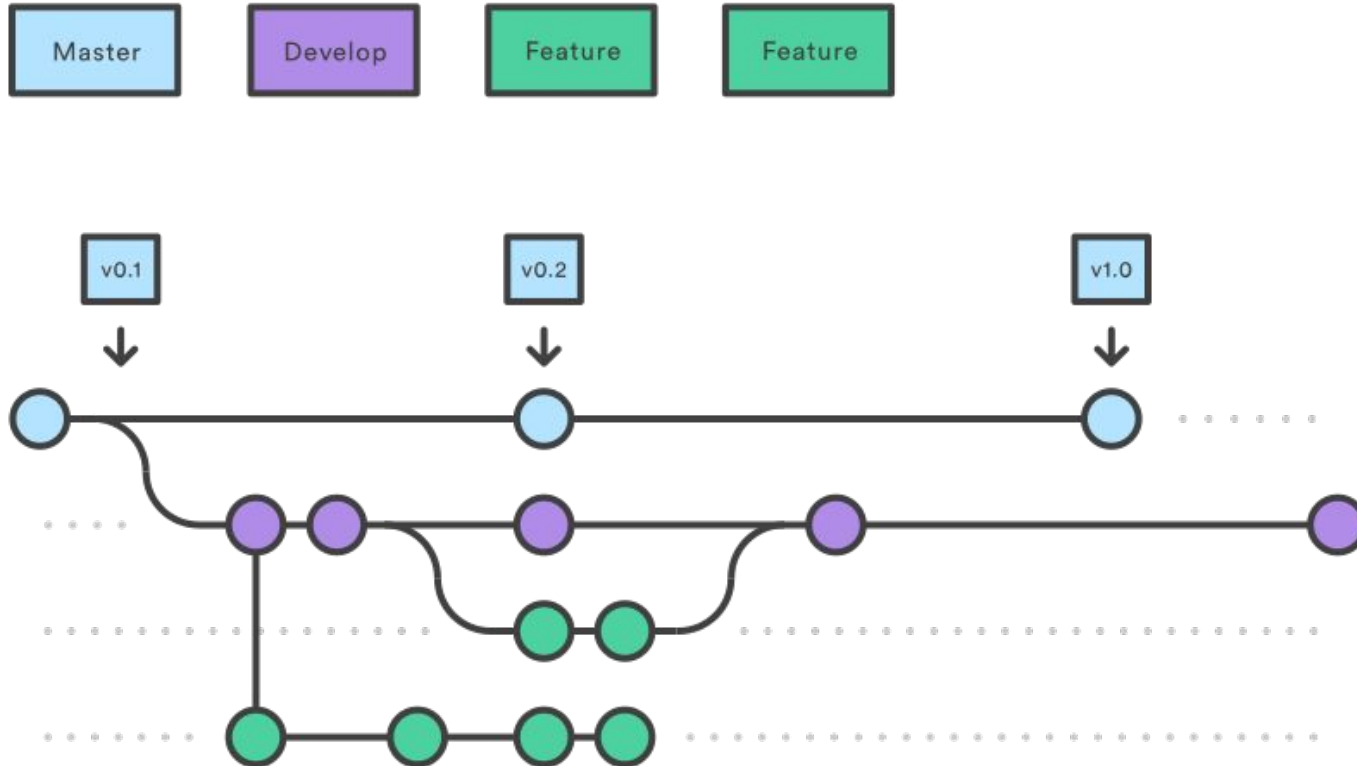
Summary of actions:

- The feature branch 'feature/ISA-123' was merged into 'develop'
- Feature branch 'feature/ISA-123' has been locally deleted

You are now on branch 'develop'

```
linux@iSA ~ $ git branch
* develop
master
```

Git Flow Feature branches



ĆWICZENIE

- Sklonuj repozytorium (adres na Slacku)
- Zainicjuj git flow z domyślnymi ustawieniami
- Utwórz nowy feature branch o nazwie imie.nazwisko
- Będąc na feature branchu utwórz w repo plik o nazwie test.imie.nazwisko
- Utwórz commit i zamknij feature branch
- Prześlij zmiany z brancha develop do githuba

Git Flow Release (start)

```
linux@iSA ~ $ git flow release start 1.1
```

Branches 'develop' and 'origin/develop' have diverged.

And local branch 'develop' is ahead of 'origin/develop'.

Switched to a new branch 'release/1.1'

Summary of actions:

- A new branch 'release/1.1' was created, based on 'develop'
- You are now on branch 'release/1.1'

Follow-up actions:

- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

```
git flow release finish '1.1'
```

```
linux@iSA ~ $ git branch
```

```
develop
```

```
master
```

```
* release/1.1
```

Git Flow Release (finish)

```
linux@iSA ~ $ git flow release finish 1.1
```

Branches 'master' and 'origin/master' have diverged.

And local branch 'master' is ahead of 'origin/master'.

Branches 'develop' and 'origin/develop' have diverged.

And local branch 'develop' is ahead of 'origin/develop'.

Already on 'master'

Your branch is ahead of 'origin/master' by 3 commits.

(use "git push" to publish your local commits)

Switched to branch 'develop'

Already up to date!

Merge made by the 'recursive' strategy.

Deleted branch release/1.1 (was b0acee5).

Summary of actions:

- Release branch 'release/1.1' has been merged into 'master'
- The release was tagged '1.1'
- Release tag '1.1' has been back-merged into 'develop'
- Release branch 'release/1.1' has been locally deleted
- You are now on branch 'develop'

Git Flow Release (finish)

```
linux@iSA ~ $ git log
commit 2eb6e73551191b1bfee8f3dd971d0610335681e7 (HEAD -> develop)
Merge: b0acee5 fb78f64
Author: Daniel Kossakowski <daniel@krolnet.pl>
Date: Mon Nov 19 10:00:22 2018 +0100
```

```
    Merge tag '1.1' into develop
    Release 1.1
```

```
commit fb78f6472651ad72fdaa8394e4ae581907f34621 (tag: 1.1, master)
Merge: b98ef5e b0acee5
Author: Daniel Kossakowski <daniel@krolnet.pl>
Date: Mon Nov 19 10:00:04 2018 +0100
```

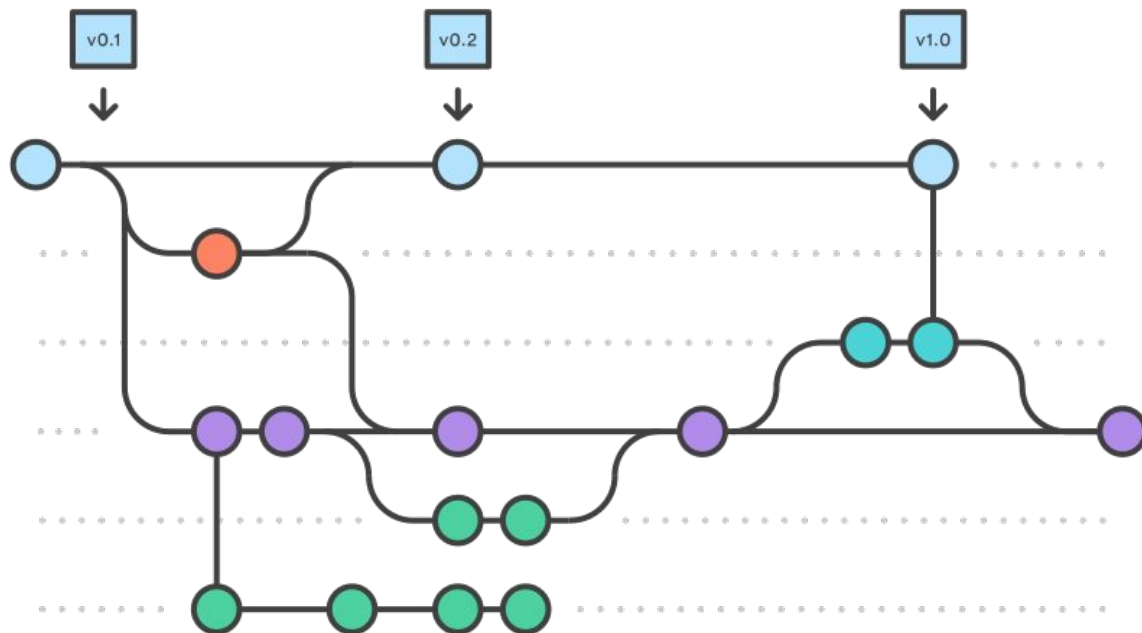
```
    Merge branch 'release/1.1'
```

```
commit b0acee5cd6176a50ec9bf39d870c464bf4c2e72b
Author: Daniel Kossakowski <daniel@krolnet.pl>
Date: Mon Nov 19 09:22:59 2018 +0100
```

```
    Add feature.file (using git flow)
```

```
commit 852c180f376e4ecaba2f42487b5b99427fe12b41 (origin/develop)
```


Git Flow Release (finish)



3. Schowek

Schowek Jak to działa?

- Warto użyć gdy nie chcemy commitować nieskończonej pracy
- Zachowuje aktualny stan katalogu roboczego (niezatwierdzony)

- `git stash` - zapisanie stanu do schowka
- `git stash list` - lista zapisanych stanów

- `git stash apply [...]` - wczytanie ze schowka
- `git stash drop [...]` - usunięcie ze schowka

Schowek Zapisywanie

```
linux@iSA ~ $ git status
```

On branch master

Your branch is up to date with 'origin/master'.

Changes not staged for commit:

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

(use "git checkout -- <file>..." to discard changes in working directory)

zmodyfikowany: config/application.config.php

no changes added to commit (use "git add" and/or "git commit -a")

```
linux@iSA ~ $ git stash
```

Saved working directory and index state WIP on master: 9dd90d6 Add missing translations

```
linux@iSA ~ $ git status
```

Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

Schówek Odczytywanie

```
linux@iSA ~ $ git stash apply
```

On branch master

Your branch is up to date with 'origin/master'.

Changes not staged for commit:

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

(use "git checkout -- <file>..." to discard changes in working directory)

zmodyfikowany: config/application.config.php

no changes added to commit (use "git add" and/or "git commit -a")

```
linux@iSA ~ $ git stash list
```

stash@{0}: WIP on master: 9dd90d6 Add missing translations

stash@{1}: WIP on master: 494c8d7 Add support to language subdomains

Schówek Usuwanie

```
linux@iSA ~ $ git stash list
stash@{0}: WIP on master: 9dd90d6 Add missing translations
stash@{1}: WIP on master: 494c8d7 Add support to language subdomains
linux@iSA ~ $ git stash drop
Dropped refs/stash@{0} (fc1c495ee10aba15aad6d634c27d840eac30f633)
linux@iSA ~ $ git stash drop
stash@{0}: WIP on master: 494c8d7 Add support to language subdomains
```

ĆWICZENIE

- Zainicjuj nowe repozytorium
- Utwórz plik o nazwie test i stwórz nowy commit
- Zmodyfikuj plik test i uruchom komendę git status
- Zapisz swoją pracę do schowka i ponownie uruchom git status
- Jeszcze raz zmodyfikuj plik test, tym razem z inną zawartością i stwórz commit
- Wczytaj swoją poprzednią pracę ze schowka

4. .gitconfig

.gitconfig Plik konfiguracyjny

- Ustawia parametry gita lub repozytorium
- Ścieżka użytkownika: ~/.gitconfig (/home/user/.gitconfig)
- Ścieżka lokalna: .git/config

.gitconfig Plik konfiguracyjny

```
linux@iSA ~ $ cat .git/config
[core]
  repositoryformatversion = 0
  filemode = true
  bare = false
  logallrefupdates = true
[remote "origin"]
  url = ssh://git@github.com:dkossako/test-repo
  fetch = +refs/heads/*:refs/remotes/origin/*
[branch "master"]
  remote = origin
  merge = refs/heads/master
[pack]
  buildbitmaps = false
```

.gitconfig Interfejs

- `git config [...]` - ustawienie lokalne
- `git config --global [...]` - ustawienie globalne
- `git config --global user.name 'Jan Nowak'`
- `git config --global user.email 'jannowak@example.com'`
- `git config user.email` - wyświetlenie zawartości zmiennej
- Ścieżka użytkownika: `~/.gitconfig` (`/home/user/.gitconfig`)
- Ścieżka lokalna: `.git/config`

ĆWICZENIE

- Wyświetl zawartość pliku ~/.gitconfig
- Ustaw poprawne i globalne wartości user.name i user.email
- Ponownie wyświetl zawartość pliku ~/.gitconfig
- Zainicjalizuj nowe repozytorium i ustaw wartość user.name lokalnie
- Wyświetl plik .git/config

5.

Git oraz SSH

Git oraz SSH Czym jest SSH?

- Protokół do zarządzania serwerami
- Zapewnia bezpieczne połączenie
- Uwierzytelnianie przez hasło lub klucz
- Szyfrowane od początku do końca

Git oraz SSH Jak działają klucze?

- Klucz prywatny – tajny, nie można nikomu udostępniać
- Klucz publiczny
- Należy dopasować klucz prywatny do publicznego
- Logowanie bez hasła

Git oraz SSH Generowanie kluczy

```
linux@iSA ~ $ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:2yn3eiKM5P9pPTekOWDMhEU+0T7nGbtIrUafnS0tDyc root@daniel-pc
The key's randomart image is:
+---[RSA 2048]----+
|      .o.      |
|      ....     |
|     oo.       |
|    ..o o      |
|   S+  = +     |
|    o=.o *     |
|   o oo.+ = BE=+|
|   o 0000@ O=+ |
|   ..++=. + =. |
+----[SHA256]-----+
```


Git oraz SSH Github - Dodanie klucza

Personal settings

[Profile](#)[Account](#)[Emails](#)[Notifications](#)[Billing](#)[SSH and GPG keys](#)[Security](#)[Sessions](#)[Blocked users](#)

SSH keys

[New SSH key](#)

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

**SSH****daniel.kossakowski@droptica.pl**

43:62:fd:2c:1a:67:f4:0a:60:00:4c:80:b9:f5:89:21

Added on 15 Nov 2018

Last used within the last week — Read/write

[Delete](#)

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#).

GPG keys

[New GPG key](#)

There are no GPG keys associated with your account.

Git oraz SSH Github - Dodanie klucza

Personal settings

[Profile](#)[Account](#)[Emails](#)[Notifications](#)[Billing](#)[SSH and GPG keys](#)[Security](#)[Sessions](#)[Blocked users](#)[Repositories](#)

SSH keys / Add new

Title

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDFnNwyRNI2zzXPjslGXKTVML9utOGXeXro/UV23e28pl91kt5hC67Aml
9l5mJjm011L6yw
/JIUGKXUakUs6L+WyMhvWk2EDt1aWIMkBf5PWNuejxOU2XucDqYv9WRtv0WxARQvQX2D2GIkx7ItJg0jZwuRX+
OiyZNv9GrPDgmjUUFAlO9dyttuBuWCs9RQ8QdHmY0+dhTaR0Ew6NS9xAjKIAQuwdXBG
/PUQ4EFze868wKGYC2UsD
/Pi1YKWCXY5ERrA07SCmq5oQ4YzV0ZNzV80+YjFIO0gNhfhvRV3DyTaYsrlfHngec7h+ZW+dmUNygKrWEIrd9xVynf
zufthzBxR daniel@krolnet.pl
```

Add SSH key

Git oraz SSH Github - Dodanie klucza

Personal settings

[Profile](#)[Account](#)[Emails](#)[Notifications](#)[Billing](#)[SSH and GPG keys](#)[Security](#)[Sessions](#)[Blocked users](#)

SSH keys

[New SSH key](#)

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.



SSH

daniel.kossakowski@droptica.pl

43:62:fd:2c:1a:67:f4:0a:60:00:4c:80:b9:f5:89:21

Added on 15 Nov 2018

Last used within the last week — Read/write

[Delete](#)

SSH

daniel@krolnet.pl

d4:50:dd:ed:2f:68:0f:1f:02:d1:f7:99:b4:33:16:c3

Added on 18 Nov 2018

Never used — Read/write


[Delete](#)




Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#).

Git oraz SSH Test połączenia

```
linux@iSA ~ $ ssh git@github.com
PTY allocation request failed on channel 0
Hi dkossako! You've successfully authenticated, but GitHub does not provide shell access.
Connection to github.com closed.
```

Git oraz SSH Github - Dodanie klucza






 **dkossako / liquidsoap**
forked from savonet/liquidsoap

 Unwatch ▾ 1  Star 0  Fork 49

[Code](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Insights](#) [Settings](#)


Audio and video streaming language <http://liquidsoap.fm/> [Edit](#)


[Manage topics](#)


 **2,841** commits  **19** branches  **18** releases  **15** contributors  GPL-2.0


Branch: **master** ▾ [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download ▾](#)



This branch is 5 commits behind savonet:master.

 **rhizome** and **toots** Reformat dependency lists ([savonet#644](#))

 [doc](#) Remove mentions of --dynamic-plugins-dir

 [examples](#) Update versions, use liquidsoap.info

 [gui](#) Bump copyright years.

Clone with SSH  [Use HTTPS](#)
Use an SSH key and passphrase from account.
`git@github.com:dkossako/liquidsoap.git` 
[Download ZIP](#)
5 months ago

ĆWICZENIE

- Wygeneruj swoją parę kluczy SSH
- Zaloguj się do Githuba i dodaj klucz publiczny
- Sklonuj dowolne swoje repozytorium przy pomocy SSH

6. Zdalne gałęzie

Zdalne gałęzie Jak i dlaczego?

- Wspólna praca nad kodem
 - Praca zdalna w projekcie
 - Rozproszony tryb repozytorium
-
- git remote - lista zdalnych gałęzi
 - git remote add - dodanie nowej
 - git remote remove - usunięcie istniejącej (odpięcie)
-
- Można ustawić adresy dla operacji fetch i push
 - Mogą być inne (używane przy mirrorach)

Zdalne gałęzie Jak i dlaczego?

```
linux@iSA ~ $ git remote -v
origin  ssh://git@github.com:dkossako/fscom-zf3 (fetch)
origin  ssh://git@github.com:dkossako/fscom-zf3 (push)
linux@iSA ~ $ git remote add remote ssh://git@bitbucket.org:dkossako/fscom-backup
origin  ssh://git@github.com:dkossako/fscom-zf3 (fetch)
origin  ssh://git@github.com:dkossako/fscom-zf3 (push)
remote  ssh://git@bitbucket.org:dkossako/fscom-backup (fetch)
remote  ssh://git@bitbucket.org:dkossako/fscom-backup (push)
linux@iSA ~ $ git push remote master
remote: Counting objects: 43, done.
remote: Compressing objects: 100% (36/36), done.
remote: Total 43 (delta 10), reused 31 (delta 5)
Unpacking objects: 100% (43/43), done.

* [new branch]    master    -> remote/master
```

ĆWICZENIE

- Zainicjuj nowe puste repozytorium
- Utwórz commit z plikiem testowym
- Zaloguj się do Githuba i utwórz nowy projekt
- Zobacz wynik komendy `git remote -v`
- Dodaj gałąź zdalną remote w lokalnym repozytorium
- Ponownie zobacz wynik komendy `git remote -v`
- Wyślij zawartość swojego repo do Githuba



Koniec!

Dziękuję za uwagę

Pytania?