Developer Academy Poland

Let’s GiT OUT!

By Patryk Podolski 2nd of November 2023

SMARTBEAR

Agenda

- Introduction
- Why so much fuss about Git
- Git's way of work
- May the “git -- force” be with you (live coding)
- Questions
- Summary

Introduction

SMARTBEAR

Introduction

- **Patryk Podolski**
  - Senior software Engineer
  - LoadNinja

Why so much fuss about Git

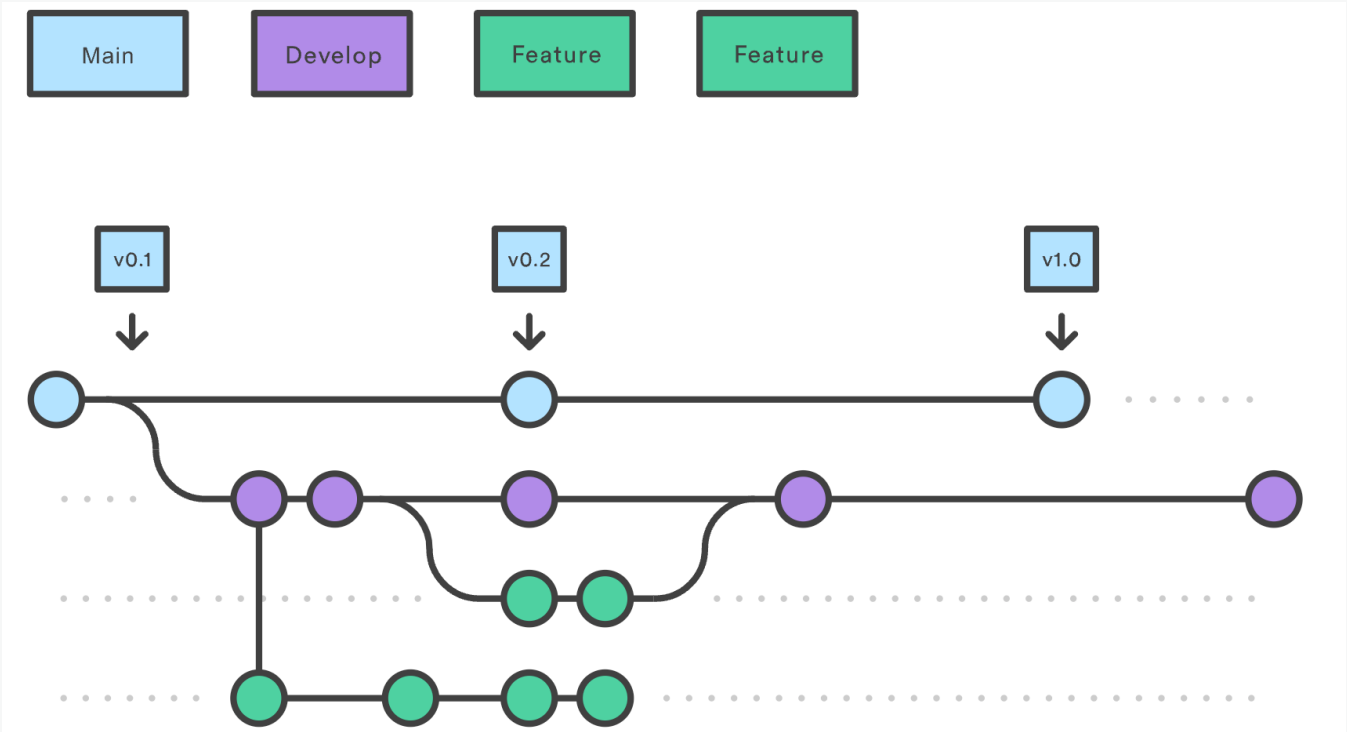
Why so much fuss about Git

- Version control system
- Distributed
- Great work history
- Branching
- Free / fast / small
- Easy to learn & use
- Big community/docs

Git's way of work

SMARTBEAR

Git's way of work



Repository

Branch

Commit

Git's way of work

- What is repository
  - Online/local
  - Source of truth
  - Enables distribution

Git's way of work – Repository commands

git clone (*init*) => copy/ initiate repository locally

git fetch => check and displays online changes

git pull => check and downloads online changes

Git's way of work – Operations on branches

git checkout <branch_name> => move between branches

git checkout –b <new_ branch_name> => create new branch and loads it

git branch –D <branch_name> => deletes branch

git merge <branch_to_marge> => update feature branch

Git's way of work – Dealing with code changes

git status => displays info about any un-commited changes

git add –all (-A) => add all (including untracked files) to staging, before commit state

git commit –m "short commit message" => final commit of changes that creates changes in history

git push => pushes local changes to remote repository

git push –u origin <branch_name> => pushes & created connection to remote repository

Git's way of work – Work in progress (1/2)

git stash –u => save all changes in git memory without commiting

git stash apply (stash@{index}) => apply last saved changes from stashed stack LIFO

git stash drop => deletes last saved changes from stashed stack

git stash list => print all stashed changes

git stash clear => clear entire history of stashed changes

git checkout -f => reverts all local changes that are not staged or commited

Git's way of work – Work in progress (2/2)

git branch => prints list of all local branches

git log -{last_commits_number} => print last commits info from current branch LIFO

git reflog => print all latest git actions that were performed locally

git diff <hash_1> <hash_2>=> clear entire history of stashed changes

Git's way of work

- **GitHub**
  - Host repository
  - Creating Pull-Request (PR, MR)
  - Reviewing changes
  - Approving changes
  - Closing PR
  - Browse commit history
  - An many many more…

May the “git -- force” be with you (live coding)

May the “git -- force” be with you (live coding)

Cheat-sheet

Repository

- git clone/init
- git fetch
- git pull

Branches

- git checkout -b
- git branch -D <branch_name>
- git merge <branch_to_merge>

Code changes

- git status
- git add --all
- git commit -m "commit message"
- git push
- git push -u origin <branch_name>

Work in progress

- git stash -u
- git stash apply
- git stash drop
- git stash clear
- git checkout -f
- git branch
- git log
- git diff <hash_1> <hash_2>

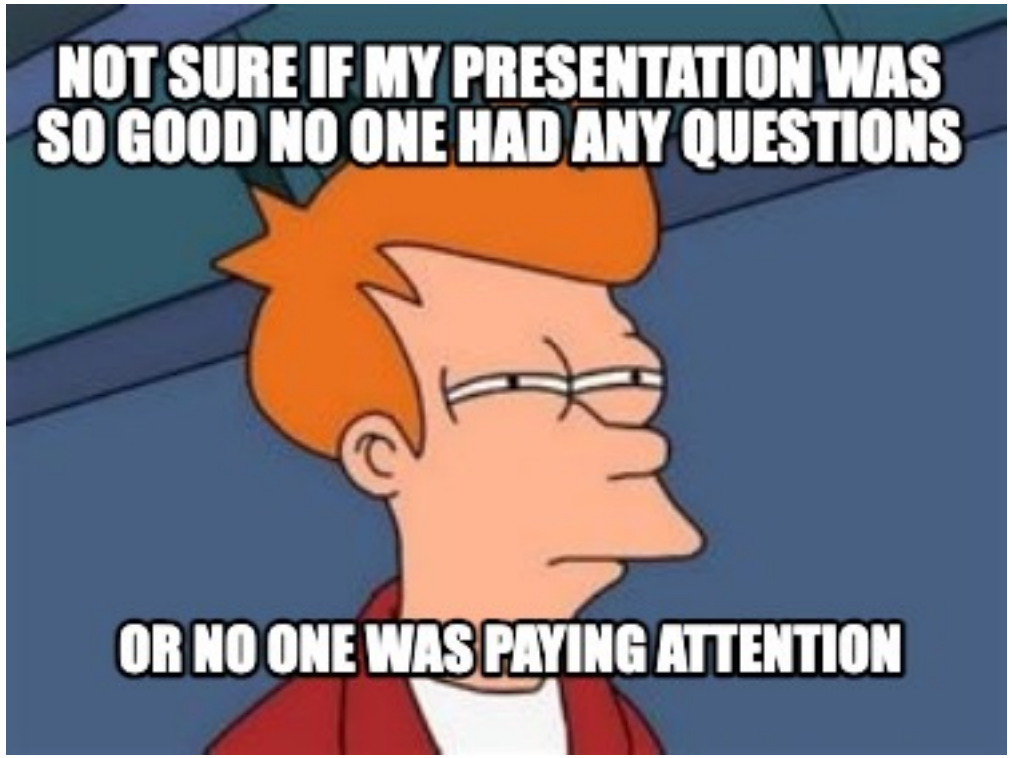
Combine

- git add -all && git commit -m "xyz" && git push

Questions

SMARTBEAR

Questions



Summary

SMARTBEAR

Summary

- Survey
- Go Git
- Dig in – Documentation

links

- https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token
- https://www.atlassian.com/git
- https://git-scm.com/doc

That’s all folks!

SMARTBEAR