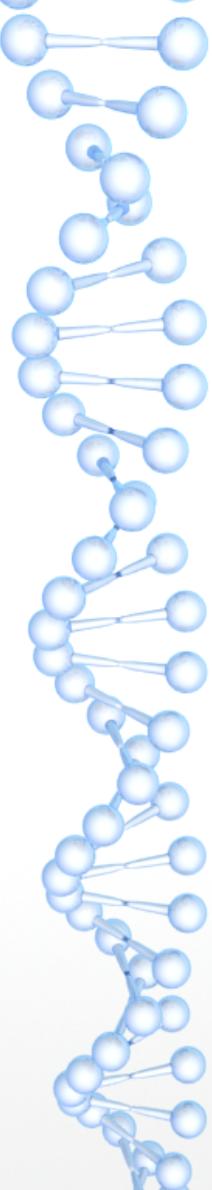


GIT

GIT primer

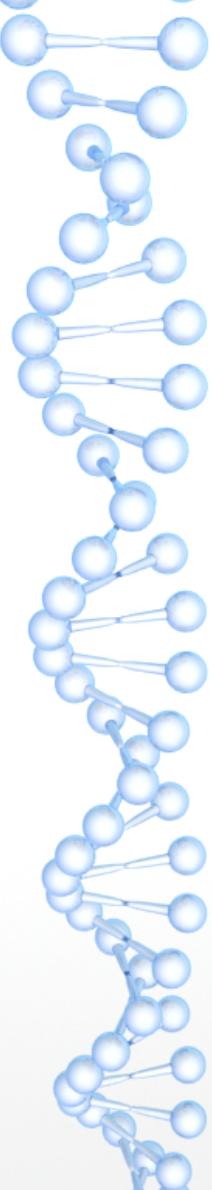
*Ako na GIT aj pre
úplných začiatočníkov*

Miloš Korenčiak (wezeo)



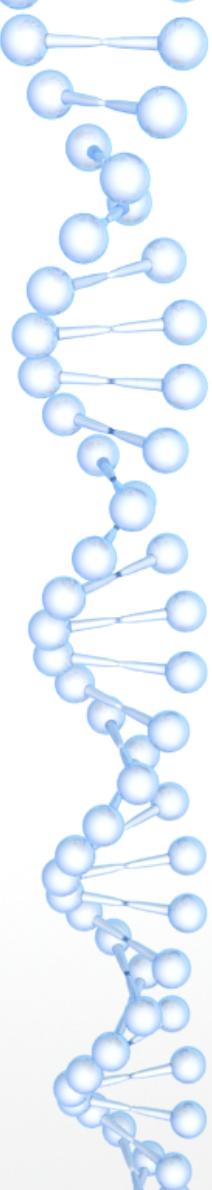
Obsah

1. Čo (ne)urobí GIT za mňa?
2. Čo je vlastne GIT zač?
3. Tri stromy/databázy v GIT
4. Hands on lab (praktická ukážka toho najbežnejšieho na konzole)
5. Rôzne GUI pre GIT, aby sme to mali jednoduchšie
6. Sumár
7. Záver



1. Čo (ne)urobí GIT za mňa?

- Neurobí za mňa prácu
- Nevyrieši komunikáciu v tíme ani nezhody v návrhu programu
- Nevyrieši každú kolíziu s kolegovým kódom
- Nevyrieši kóderove zabúdanie zálohovať /verzovať svoj program
- Manžérovi nepomôže zistiť, či sa projekt stíha

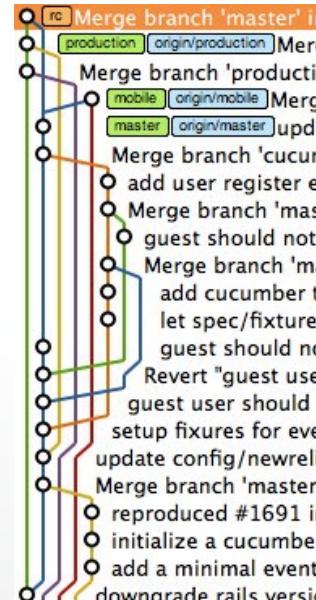
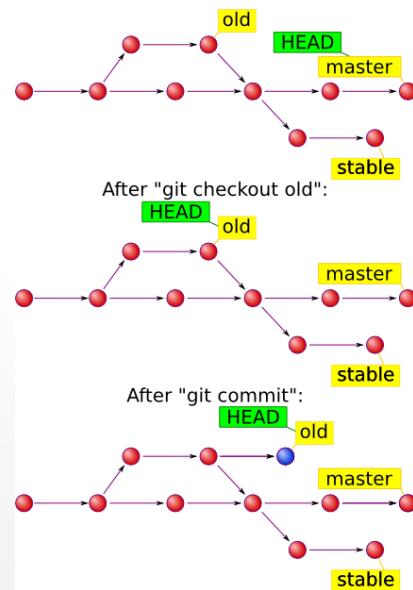


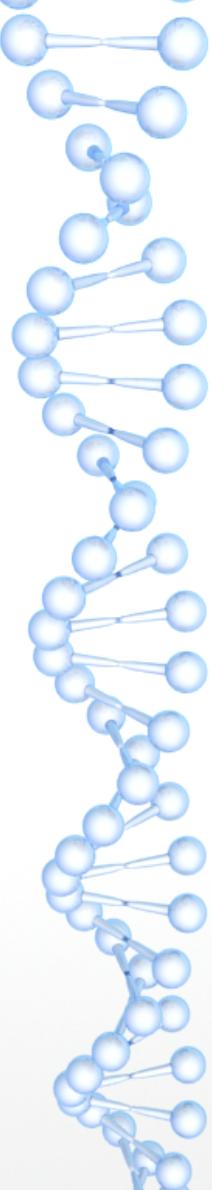
2. Čo je vlastne GIT zač?

- *a distributed revision control and source code management (SCM) system with an emphasis on speed*
(Wikipedia)
- Umožňuje manažment kódu (SCM)
- Verzuje zdrojový kód
- Je distribuovaný
- Je rýchly :-) (za cenu...)
- Umožňuje mať rozrobených viac vecí naraz +prepínanie

2. Čo je vlastne GIT zač?

- Je to vlastne „strom“ commitov (check-point kódu)
- Priponá „stanice metra“
- Je uložený v internej databáze

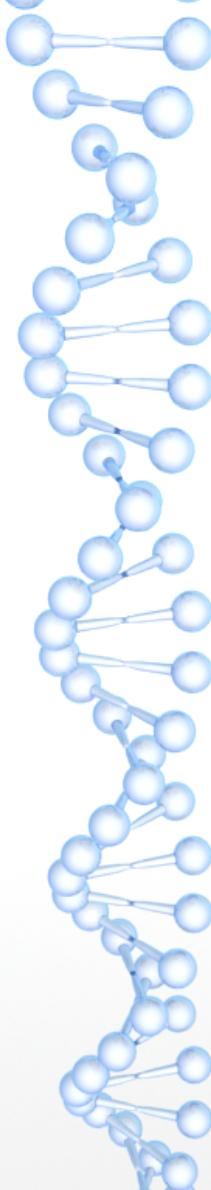




2. Čo je vlastne GIT zač?

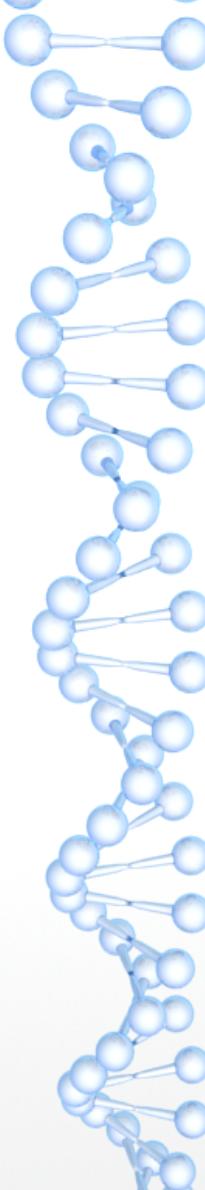
Software tools for revision control are essential for the organization of multi-developer projects.

(Wikipedia)



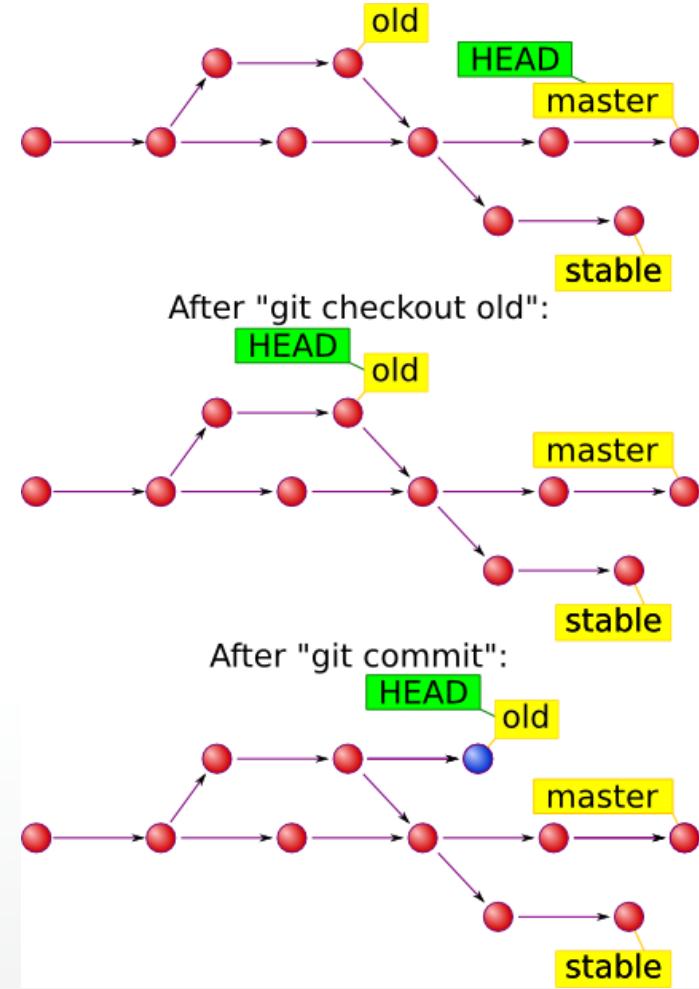
3. Tri stromy/databázy v GIT -working dir

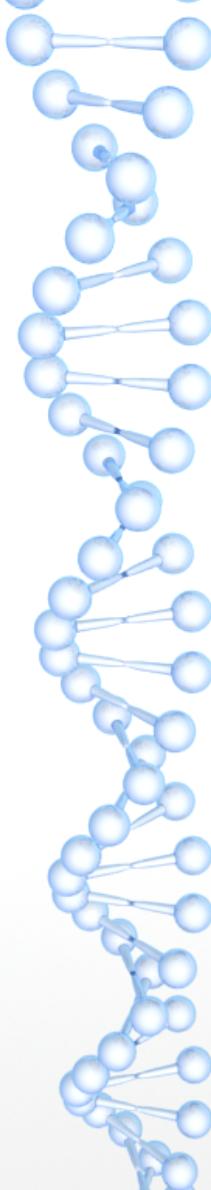
- Ten adresár v ktorom priamo programujeme
- Je v ňom teda fyzicky každý súbor projektu
- GIT prehľadáva aktuálny adresár a rekurzívne aj všetky podadresáre
- Zistuje, ktoré súbory oproti poslednému commitu pribudli (a potenciálne by sa mali pridať do repozitára), a ktoré súbory v repozitári (tzv. verzované /tracked súbory) sa zmenili
- GIT ho edituje pri checkout-e



3. Tri stromy/databázy v GIT -HEAD

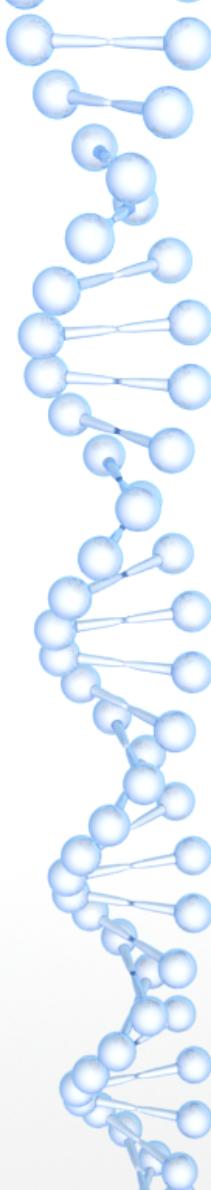
- Je to posledný commit v aktuálnej vetve „stromu“ (strom je celý GIT repozitár)
- Ak urobíme commit, stane sa synom aktuálneho HEAD commitu; HEAD ukazovateľ sa potom posunie na náš najnovší commit
- Ukazuje kde pracujeme





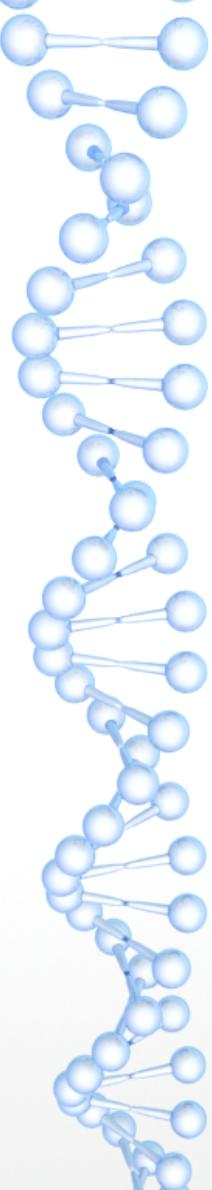
3. Tri stromy/databázy v GIT -index

- Je to miesto prípravy commitu (akýsi medzisklad)
- Pridávame tu zmeny, ktoré chceme zahrnúť do najbližšieho commitu
- Môžme tu pridať všetky zmeny zo súboru alebo len ich časť
- Môžeme pridať jeden alebo viac súborov
- Využívajú ho najmä GUI nástroje



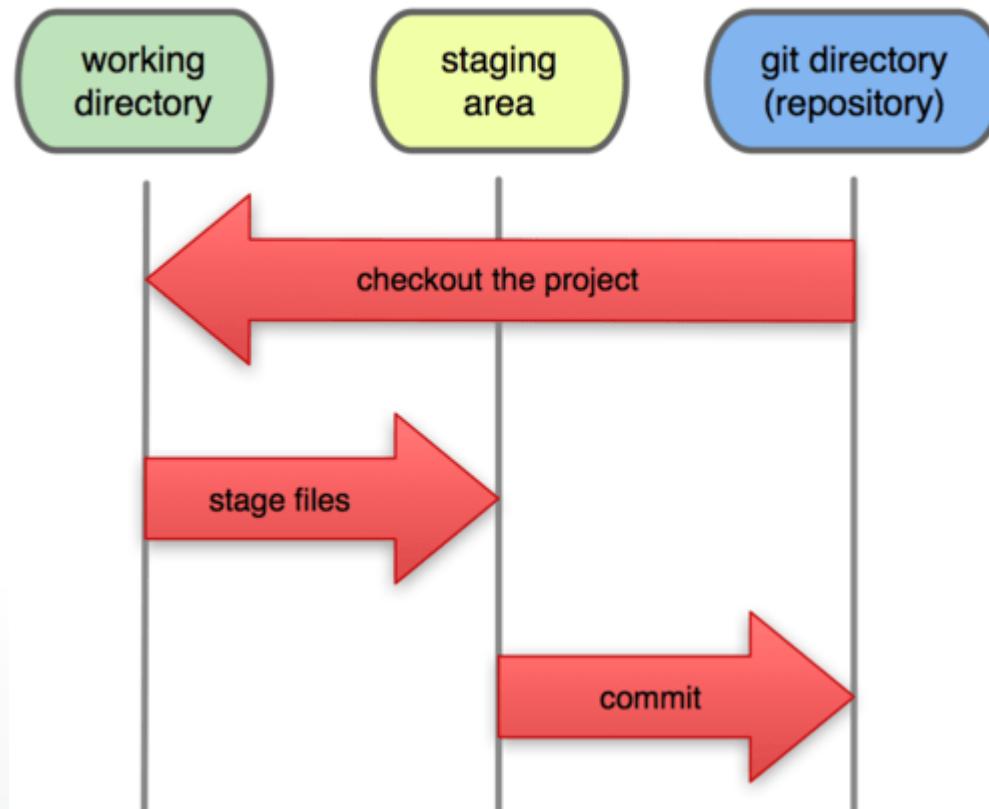
3. Tri stromy/databázy v Git -ich spolupráca

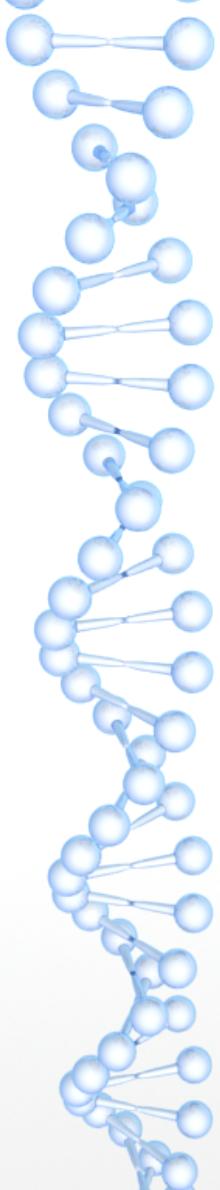
- Programovaním meníme working dir
- Príkazom git add (bude ukázaný neskôr) prenášame working dir zmeny do indexu
- Príkazom git commit (neskôr) prenášame index alebo aj working dir do stromu repozitára (až ten je verzovaný)
- IBA strom GIT repozitára je schopný priamo komunikovať so vzdialenými GIT repozitármi



3. Tri stromy/databázy v Git -ich spolupráca

Local Operations

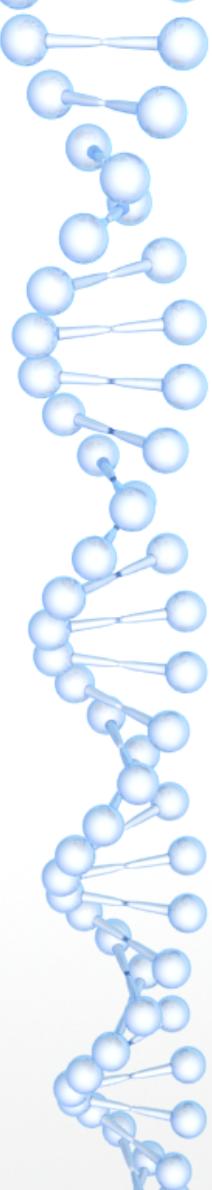




4. Hands on lab

(Príkazy ako by boli v *git bash* konzole;
so screenshotmi GitExtensions)

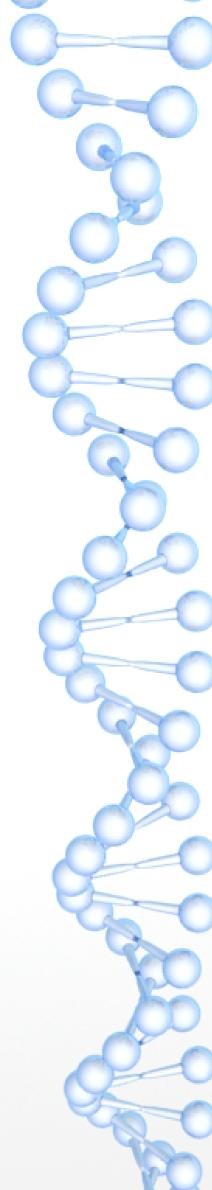
Predved'me si to na jednoduchom
projekte ...



4.a) Vytvorenie repozitára

\$ git init

- Vytvorí v aktuálnom adresári lokálny git repozitár
- Jeho všetky dáta sú v jedinom *.git* skrytom podadresári



Stroj View Zariadenia Pomocník

Git Extensions

Start Dashboard Github Plugins Tools Help

Common Actions

- [Open repository](#)
- [Clone repository](#)
- [Clone SVN repository](#)
- [Clone Github repository](#)
- [Create new repository](#)

Recent Repositories

Initialize new repository

Directory G:\aa

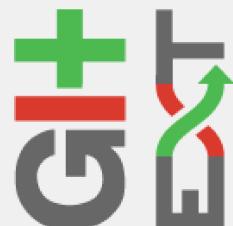
Repository type

Personal repository

Central repository, no working dir (--bare --shared=all)

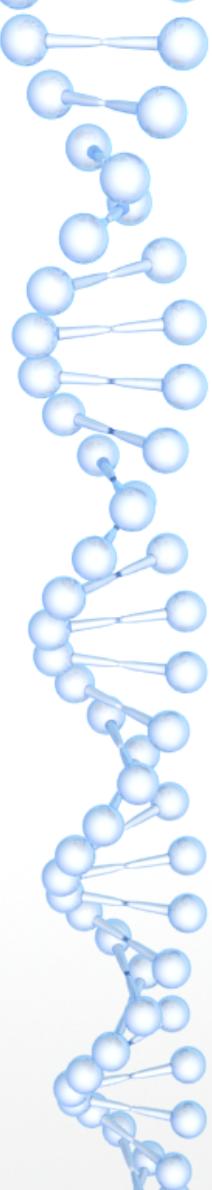
Contribute

- [Develop](#)
- [Donate](#)
- [Translate](#)
- [Issues](#)



1:17 PM
3/25/2014

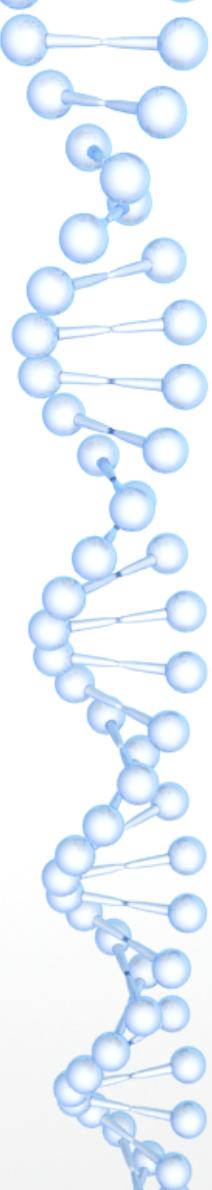
Pravy ctrl



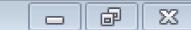
4.b) Stiahnutie repozitára

```
$ git clone https://github.com/vishnevskiy/battlenet.git  
$ git clone /home/m/kodenie/effective-django-tutorial
```

- Naklonuje repozitár z daného URI
- Zvyčajne sa používa zabezpečený protokol
- Zvyčajne tento protokol vyžaduje autorizáciu pri posielaní zmien pri budúcom *push*

- 
- My použijeme Git tutoriál pre angularJS:
(keywords do google):

git angularjs tutorial phonecat



master hahaha

babbbbrak_poradny

1 minute ago

Clone

Repository to clone: G:\aa

Browse

Destination: G:\bb

Browse

Subdirectory to create: aa

Branch: master

The repository will be cloned to a new directory located here:

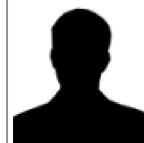
G:\bb\aa (New directory)

Repository type

 Personal repository Public repository, no working dir (--bare) Initialize all submodules

Clone

Commit File tree

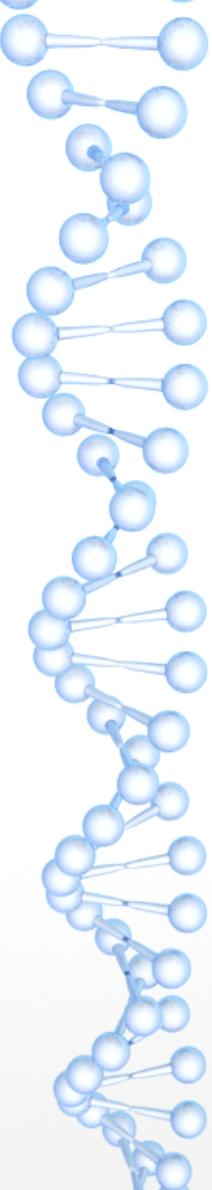


hahaha

Contained in branches: [master](#)

Contained in no tag





4.c) Commit +Add

\$ git status

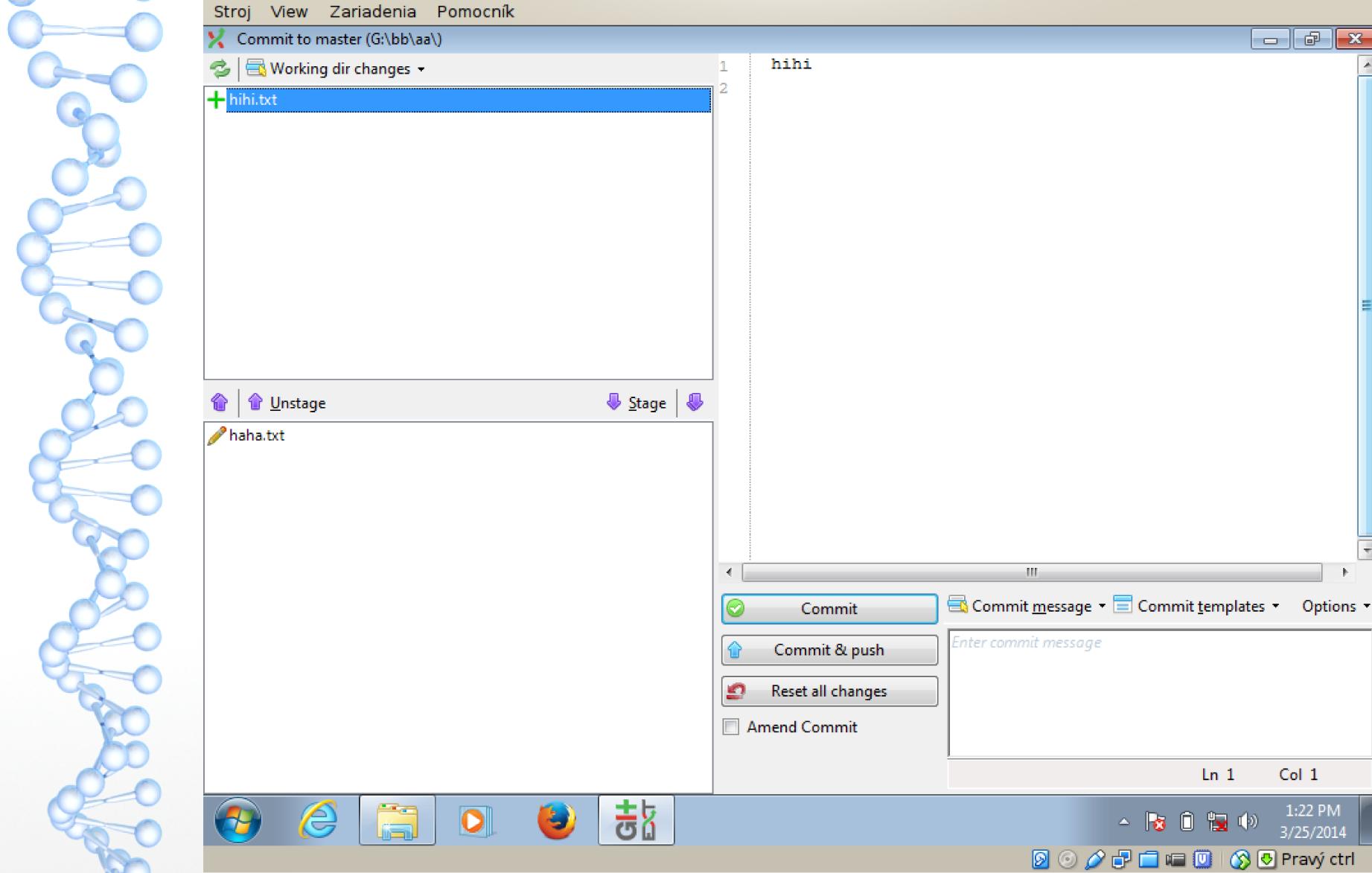
\$ git add haha.txt

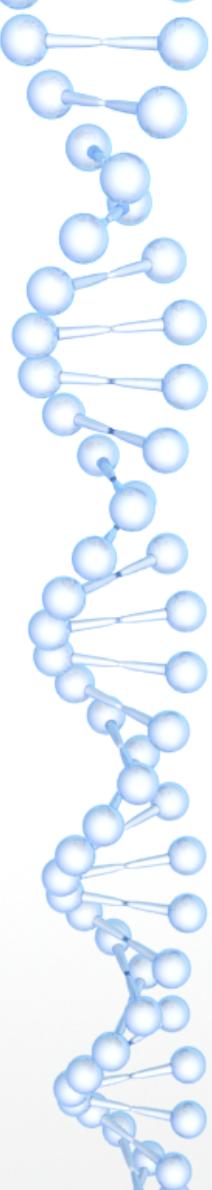
\$ git commit

\$ git commit -a

\$ git commit -a -m „ahoj -som popis commitu“

- Preskúmame stav repozitára
- Pridáme súbor, aby sa oň git staral
- Commit-neme jeho zmeny do git-u („Tri stromy git-u“)

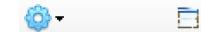




4.d) Push

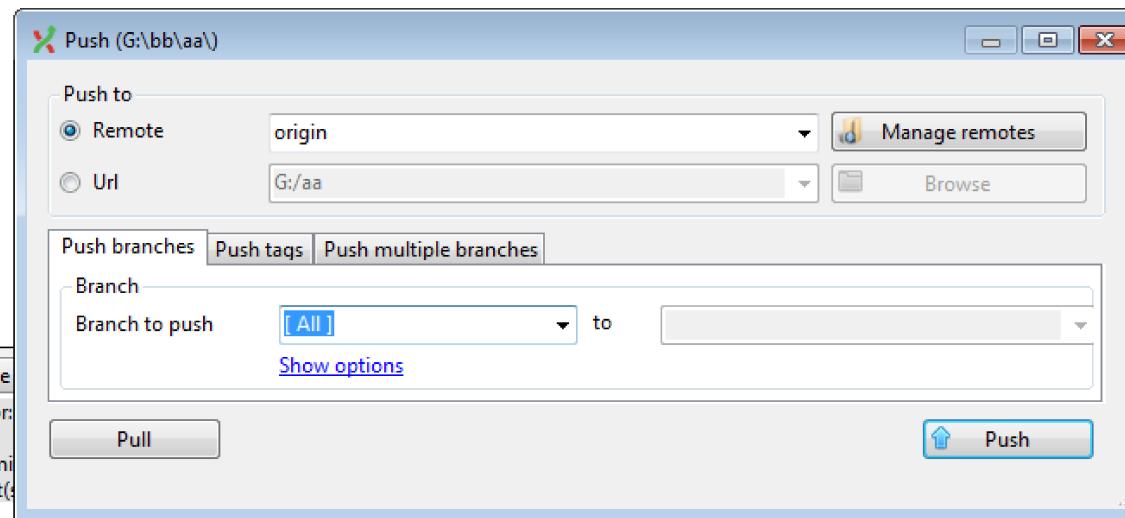
\$ git push

- Pošle naše commity do vzdialeného repozitára
- Ak nemáme nijaké nové dátu, sme o tom informovaní
- Môže skončiť neúspešne (povieme pri Merge-ovaní)



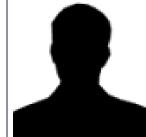
▶ master hihi added babbbrak_poradny 4 seconds ago

▷ origin/master hahaha babbbrak_poradny 7 minutes ago



Commit File

Author:
Date:
Commit:
Parent(s):

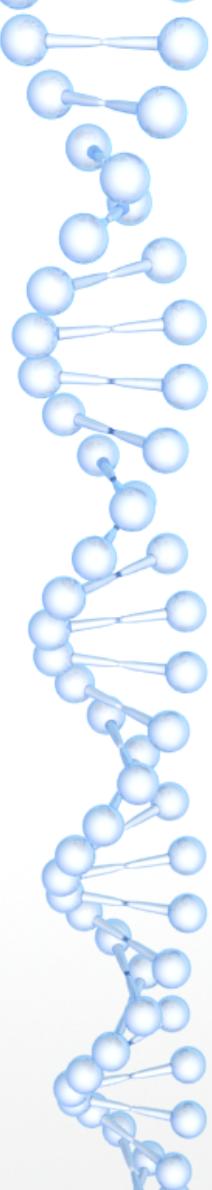


hihi added

Contained in branches: [master](#)

Contained in no tag

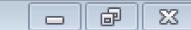




4.e) Fetch

\$ git fetch

- Získá nové commity zo vzdialeného repozitára
- Nič nezmení u nás
- Nikdy nespôsobí „problémy“



▶ master ➤ origin/master hihi added babbbrak_poradny 3 minutes ago
origin/aaaa hahaha babbbrak_poradny 11 minutes ago

remote: Compressing objects: 100% (2/2), done.+[K]

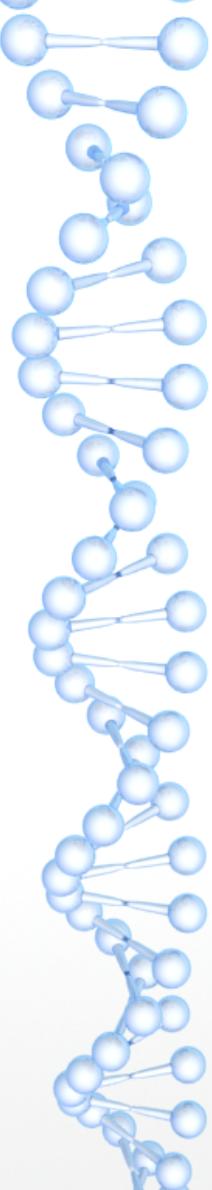
```
"C:\Program Files (x86)\Git\bin\git.exe" fetch --progress "origin"
remote: Total 3 (delta 0), reused 0 (delta 0)+[K]
From G:/aa
  a4bc140..b9c7a45  aaaa  -> origin/aaaa
Done
```

Commit File tree +
Author:
Date:
Commit hash:
Parent(s):
hihi added

Keep dialog open Abort OK

Contained in branches: [master](#)
Contained in no tag



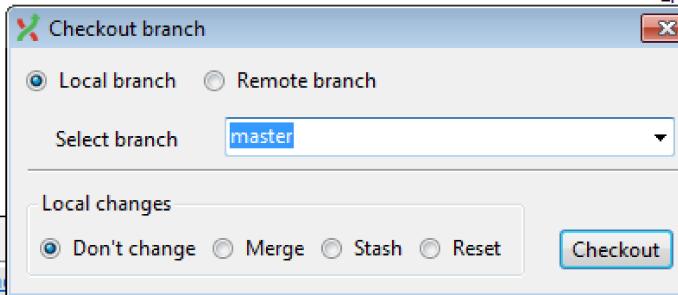


4.f) Checkout

```
$ git checkout {meno vetvy}
```

- Nikdy nespôsobuje komunikáciu so vzdialeným servrom
- Prepne nás na inú lokálne prístupnú vetvu
- *Working dir* sa upraví podľa vetvy, na ktorú sa checkout-ujeme
- Môžeme prísť o dátu napr. vo *working dir* -zvykne sa (podľa prepínačov) opýtať, či to tak chceme

stash	WIP on master: c76ca5a added hihi double	babbbrak_poradny	33 seconds ago
	untracked files on master: c76ca5a added hihi double	babbbrak_poradny	33 seconds ago
	index on master: c76ca5a added hihi double	babbbrak_poradny	34 seconds ago
master	added hihi double	babbbrak_poradny	12 minutes ago
aaaa	hihi added second time	babbbrak_poradny	16 minutes ago
	hihi added	babbbrak_poradny	20 minutes ago
	hahaha	babbbrak_poradny	27 minutes ago



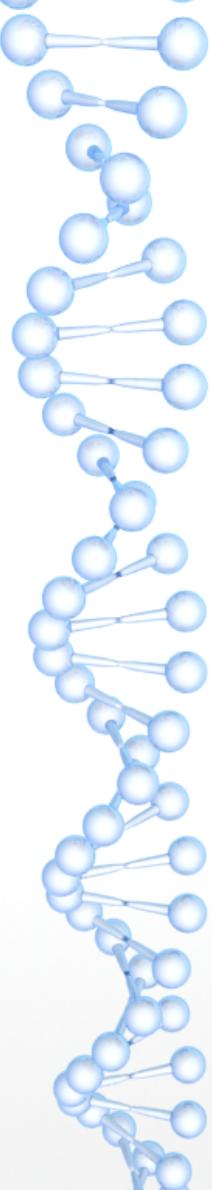
Commit File tree Diff

Author: babbbrak_pora
Date: 16 minutes ago (3/25/2014 1:26:17 PM)
Commit hash: b9c7a451b10211613a50b94e341b8eb1f45d272a
Parent(s): a4bc140d21

hihi added second time

Contained in branches: aaaa

Contained in no tag



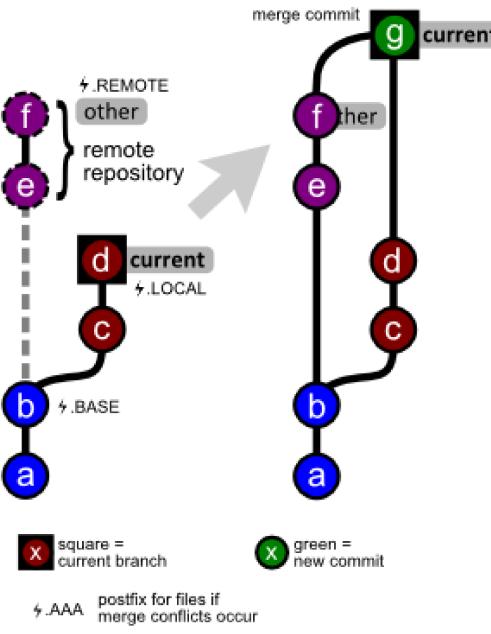
4.g) Pull

\$ *git pull*

- Urobí niečo ako fetch (stiahnutie) a následný checkout na najnovší commit vo vzdialenom repozitári
- Tiež detekuje zmenu vo *working dir* -ak máme niečo necommitnuté, varuje nás
- **Jeho cieľom je čo naintelligentnejšie ZOSYNCHRONIZOVАŤ našu lokálnu a vzdialenú vetvu**
- Robí ešte skrytý Merge...

[Hide help](#)

Hover to see scenario when fast forward is possible.



Pull from

 Remote

origin

 Url

G:/aa

Branch

Local branch

master

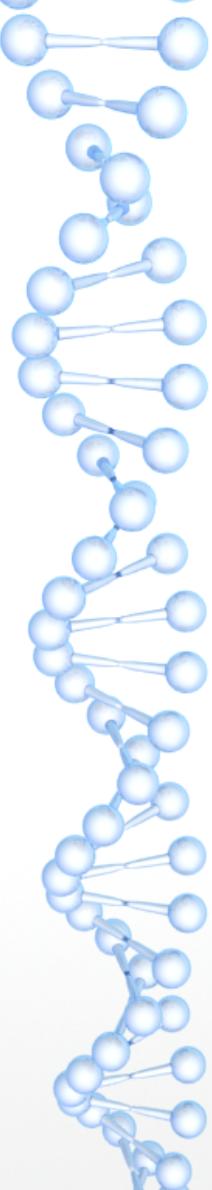
Remote branch

Merge options

 Merge remote branch into current branch Rebase current branch on top of remote branch, creates linear history (use with caution) Do not merge, only fetch remote changes

Tag options

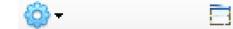
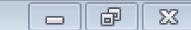
 Follow tagopt, if not specified, fetch tags reachable from remote HEAD Fetch no tag Fetch all tags[Solve conflicts](#)[Stash changes](#) Auto stash [Pull](#)



4.h) Merge-ovanie cez kdiff3

`$ git merge {iná vetva}`

- Spojí aktuálnu a {inú vetvu}
- Musí riešiť (prípadné) **kolízie kódu** -triviálne prípady rieši sám; neautomatizovateľné necháva kóderovi =**konfliktný kód**
- Konfliktný kód prenecháva špecializovaným programom -napr. *kdiff3* (konfigurovateľné)
- Je často najobávanejšou operáciou začiatočníkov



master added hihi added

aaaa hihi added sec

hihi added

hahaha

Resolve merge conflicts

Unresolved merge conflicts

Filename

hihi.txt

Open in kdiff3

Start mergetool

Rescan

Abort merge

The file has been modified locally(ours) and deleted
remotely(theirs). Choose to delete the file or keep the
modified version.

Merge

Local hihi.txt

Base hihi.txt

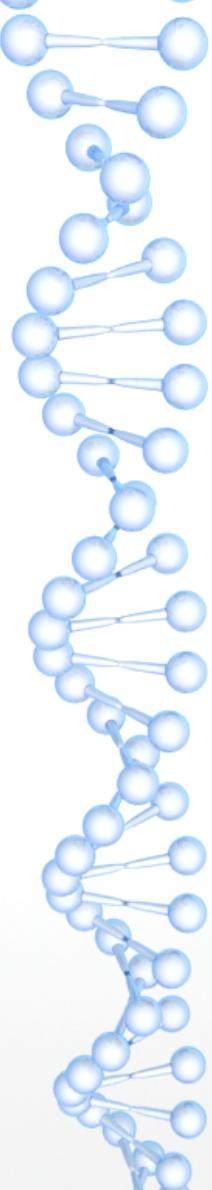
Remote deleted

Help

Contained in branches: [master](#)

Contained in no tag





4.i) Stash

`$ git stash`

- Umožňuje narýchlo odsunúť aktuálne ešte necommittedé working dir zmeny
- Dá sa k nim hocikedy vrátiť' cez `$ git stash pop`
- Funguje na princípe zásobníka
- Vhodný pri operatívnych zásahoch
(použiteľný na rýchle prepínanie sa medzi viac rozrobenými vecami)



master added hihi double

babbbbrak_poradny 11 minutes ago

aaaa hihi added second time babbbbrak_poradny 16 minutes ago

hihi added

hahaha

minutes ago

minutes ago

Process (G:\aa\)

```
"C:\Program Files (x86)\Git\bin\git.exe" stash save -u  
Saved working directory and index state WIP on master: c76ca5a added hihi double  
HEAD is now at c76ca5a added hihi double  
Done
```



Commit



Author:
Date:
Commit hash:
Children:
Parent(s):

hihi added

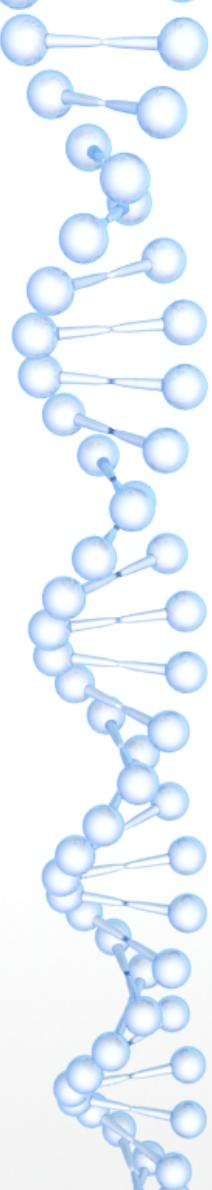
 Keep dialog open

Abort

OK

Contained in branches: [master](#)
Contained in no tag



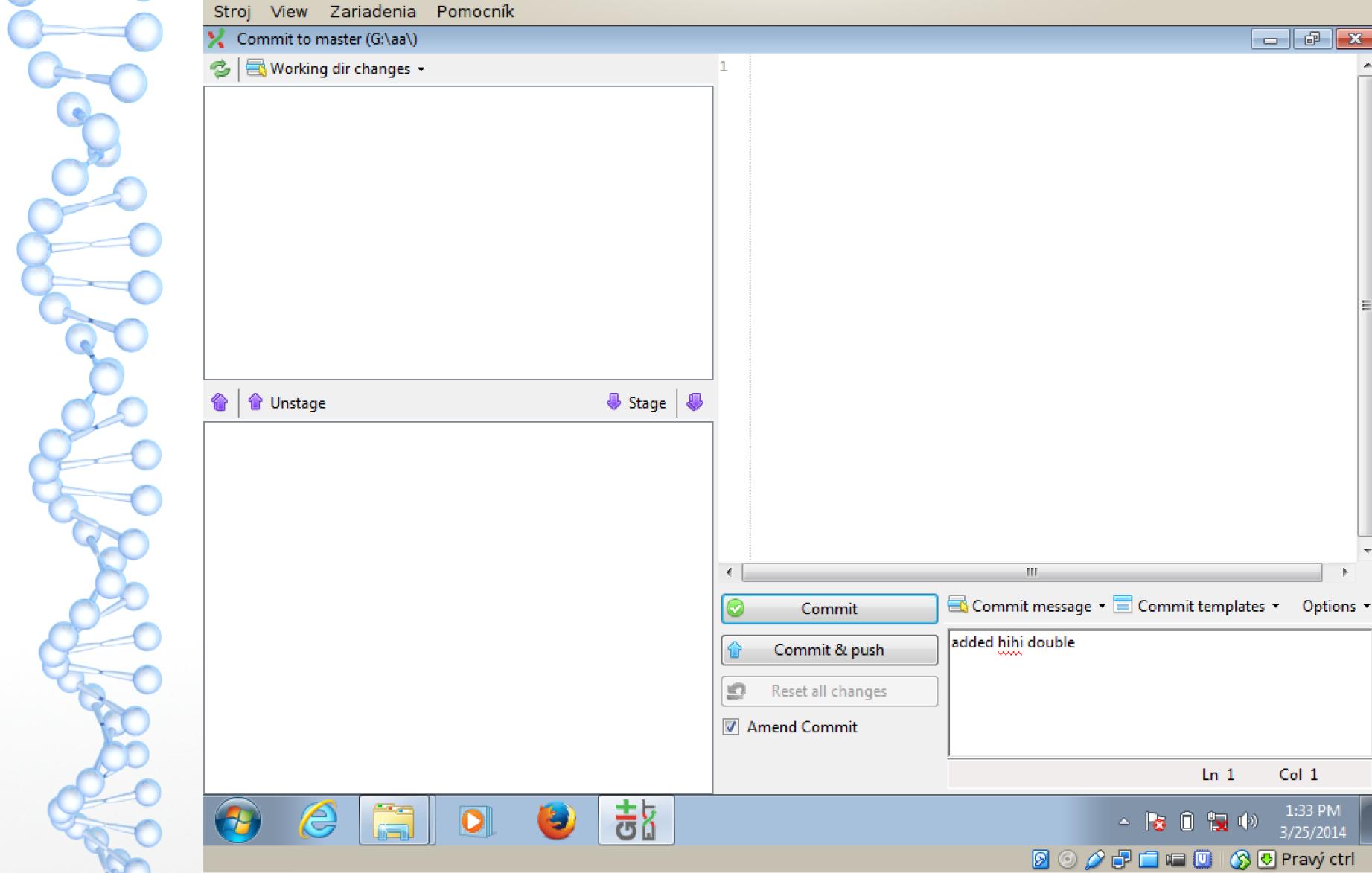


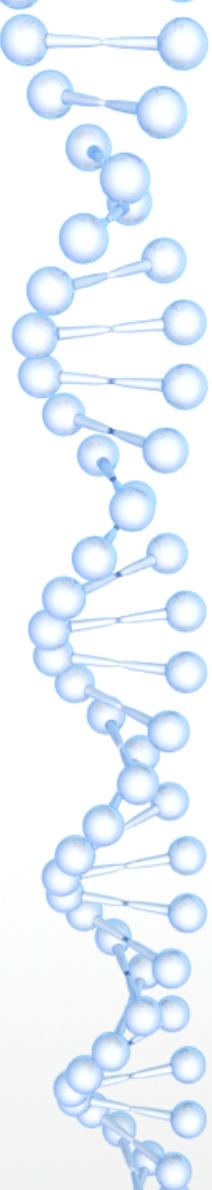
4.j) Čo ked' spravím chybu? (opravy bežných chýb)

- V commite som urobil pravopisnú chybu a zabudol súbor:

```
$ git add {zabudnutý súbor}
```

```
$ git commit –amend # pozmeni posledny commit
```





4.j) Čo ked' spravím chybu? (opravy bežných chýb)

- Posledný commit je CELÝ zle a chcem ho zrušiť (napr. aj pri zlom merge -chcem zrušiť merge)

```
$ git reset --hard HEAD^
```

```
$ git reset --soft HEAD^
```

git MINGW32:/g/aa

Welcome to Git (version 1.8.3-preview20130601)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

m@A8 /g/aa <master>

\$

Commit

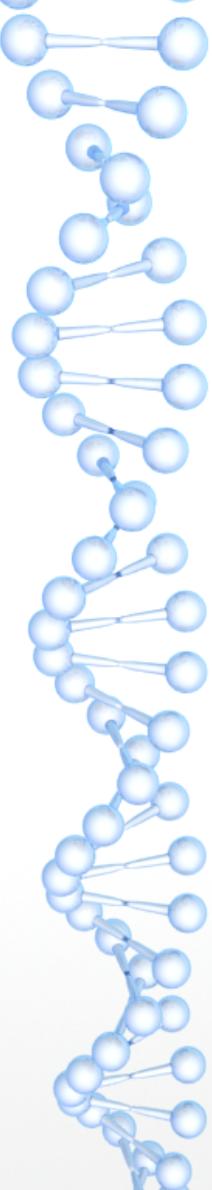
Parent(s): 1f9c15519b

added hihi double

Contained in branches: [master](#)

Contained in no tag

The screenshot shows the Git Extensions application running on a Windows system. The main window displays a commit history for the 'master' branch. The first commit, which has been selected, was made by a user with a black silhouette icon and has a commit message 'added hihi double'. It also lists its parent commit hash as '1f9c15519b'. Below the commit history, it indicates that the commit is contained in the 'master' branch. The bottom of the window features a terminal window showing the Git command line interface. The taskbar at the bottom of the screen includes icons for the Start button, Internet Explorer, File Explorer, a media player, Firefox, and the Git Extensions application itself. The system tray shows the date and time as '1:34 PM 3/25/2014'.



4.j) Čo ked' spravím chybu? (opravy bežných chýb)

- Ak už je push-nutý nejaký commit a chcem nad ním urobiť „undo“

\$ git revert {oznacenie zleho commitu -napr. Hash}

- NIKDY nedeletujeme commity, ktoré sú už push-nuté! Niekoľko iných ich mohol pull-núť a zlý commit je v histórií jeho commitov. Vzdialený repozitár toto zmazanie preto zvyčajne nedovolí.



master | added hihi double

aaaa | hihi added second time

hihi added

hahaha

babbbbrak_poradny 5 minutes ago

babbbbrak_poradny 9 minutes ago

L2 minutes ago

20 minutes ago

Revert commit

Revert this commit:

1f9c15519b

hihi added

Author: babbbbrak_poradny Commit date: 3/25/2014 1:23:02 PM

Tag(s): n/a

Branch(es): n/a

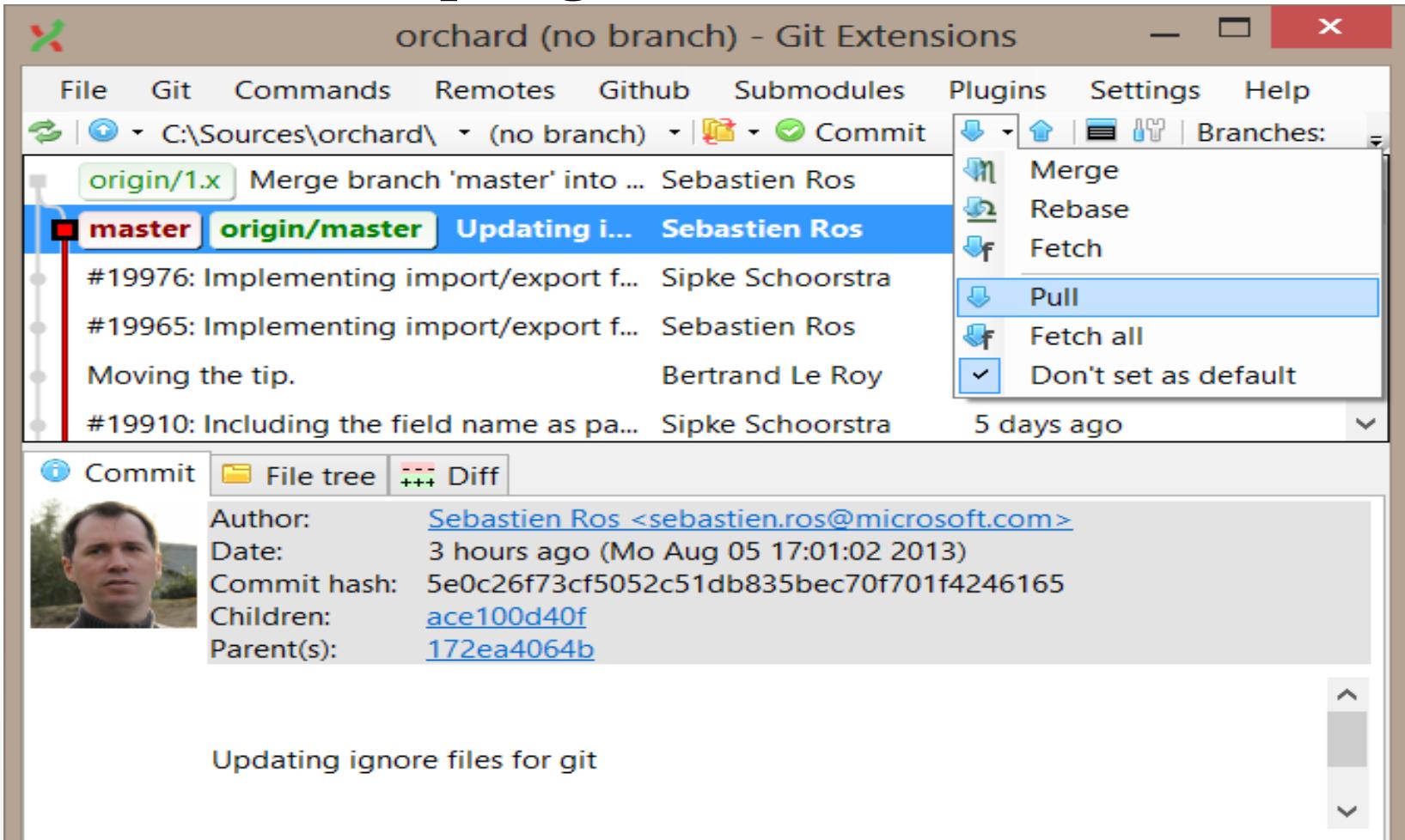
Commit File tree ++

Author:
Date:
Commit hash:
Children:
Parent(s):

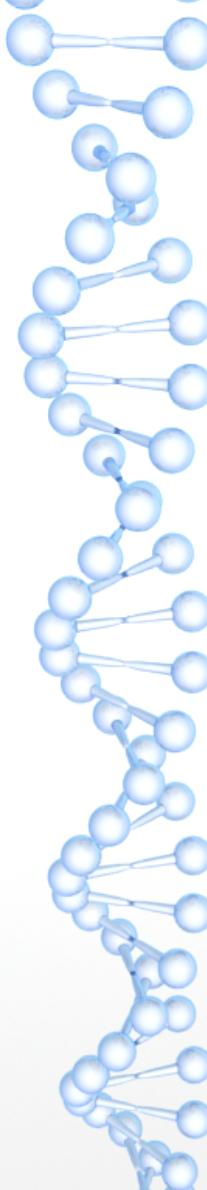
hihi added

Contained in branches: [master](#)
Contained in no tag Automatically create a commit 

5. Rôzne GUI pre git -Windows -Git Extensions



5. rôzne GUI pre git -Windows -GitHub



local
repositories

github
niik
reactiveui

+ create refresh tools

Filter Repositories

- [reactiveui/ReactiveUI](#)
- [reactiveui/ReactiveUI.Samples](#)
- [reactiveui/RxUIProjectTemplate](#)
- [reactiveui/RxUI_QCon](#)

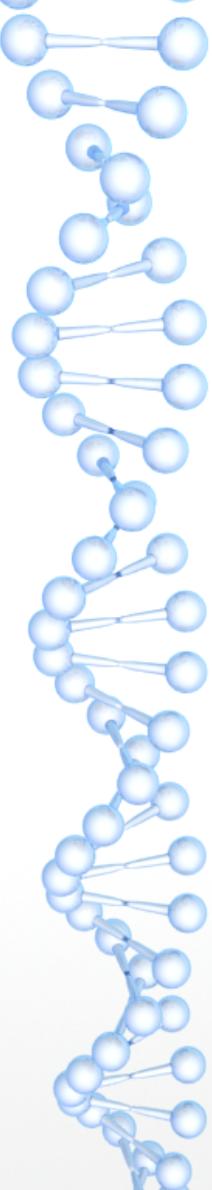
ReactiveUI

Use the Reactive Extensions for .NET along with Silverlight, WPF, or Windows Phone to create elegant, testable User Interfaces.

This library is organized into several high-level assemblies:

- **ReactiveUI** - Core library that doesn't rely on any particular UI framework. `ReactiveObject`, the base `ViewModel` object, as well as `ReactiveCollection`, a more awesome `ObservableCollection`, is in here.
- **ReactiveUI.Xaml** - Classes that require references to a Xaml'y framework, like WPF or WinRT. `ReactiveCommand`, an implementation of `ICommand`, as well as the `UserError` classes are in this assembly.
- **ReactiveUI.Blend** - This class has several Blend Behaviors and Triggers that make attaching `ViewModel` changes to Visual State Manager states.
- **ReactiveUI.Routing** - A screens and navigation framework as well as `ViewModel` locator. This framework helps you to write applications using IoC containers to locate views, as well as navigating back and forwards between views.

5. Rôzne GUI pre git -MacOS -GitHub



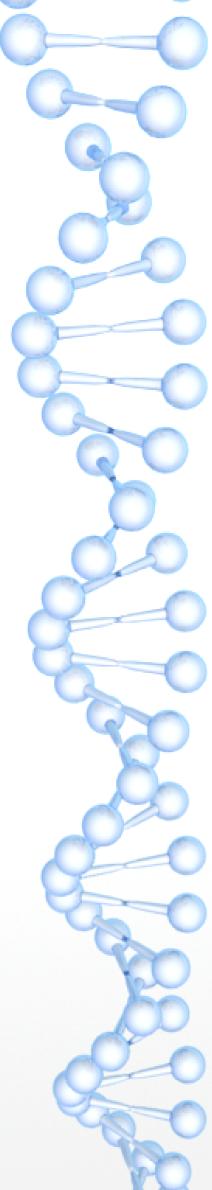
A screenshot of the GitHub desktop application interface on macOS. The window title is "Repositories > ReactiveCocoa >". The sidebar on the left has icons for History, Changes, Branches, and Settings, with "History" currently selected. The main pane shows a list of commits in the "master" branch of the "ReactiveCocoa" repository. The commits are as follows:

- 3515 commits in master (SHA: bba8fa0dee0e88ece51ed1cdc30b3dc5799af331)
- jshaber 5 hours ago: Merge pull request #864 from ReactiveCocoa/b35e733
- jspahrsummers 5 hours ago: Merge pull request #859 from ReactiveCocoa/1c39fc6
- jspahrsummers 6 hours ago: Test for +createSignal: disposing of inner subscriptions (bba8fa0) (selected commit)
- jspahrsummers 6 hours ago: Test that RACSerialDisposable.disposable... (0549fcd)
- keithduncan 12 hours ago: Use `CFBridgingRetain` in place of a cas... (c175133)
- jspahrsummers 2 days ago: Fix RACSerialDisposable leaking its set... (6de8f44)
- jspahrsummers 2 days ago: Dispose of RACSubscriber.disposable re... (d390f76)

The selected commit (bba8fa0) is expanded, showing the commit message "Test for +createSignal: disposing of inner subscriptions" and the author "Authored by jspahrsummers on Oct 14, 2013 at 10:18 AM. Verified to fail on master." Below the commit message, it says "Showing 1 changed file with 22 additions." The diff view shows the following code changes:

```
@@ -223,6 +223,28 @@ describe(@"subscribing", ^{
223   223     });
224   224     expect(currentScheduler).willNot.beNil();
225   225   });
226 +
227 + it(@"should automatically dispose of other subscriptions from
228 + +createSignal:", ^{
229 +     __block BOOL innerDisposed = NO;
230 +
231 +     RACSignal *innerSignal = [RACSignal
232 + createSignal:^(id<RACSubscriber> subscriber) {
233 +         return [RACDisposable disposableWithBlock:^{
234 +             innerDisposed = YES;
235 +         }];
236 +
237 +         RACSignal *outerSignal = [RACSignal createSignal:^{
238 +             RACDisposable * (id<RACSubscriber> subscriber) {
239 +                 [innerSignal subscribe:subscriber];
240 +                 return nil;
241 +             }];
242 +         }];
243 +     });
244 + }
```

5. Rôzne GUI pre git -Linux -qgit

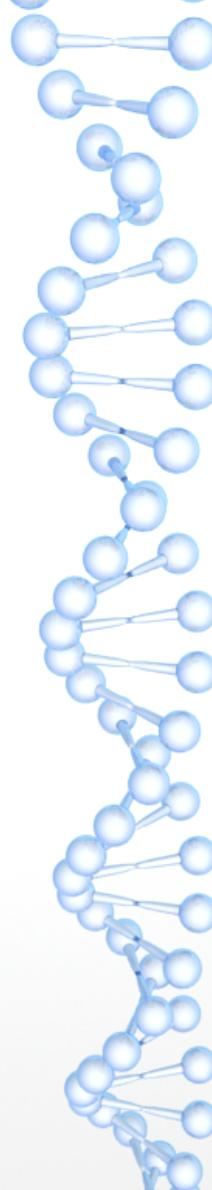


The screenshot shows the qgit graphical user interface for managing Git repositories on Linux. The window has a menu bar with File, Edit, View, Actions, and Help. A toolbar above the main area includes icons for file operations like Open, Save, and Commit, as well as a magnifying glass for search and a gear for settings.

The main interface consists of several panes:

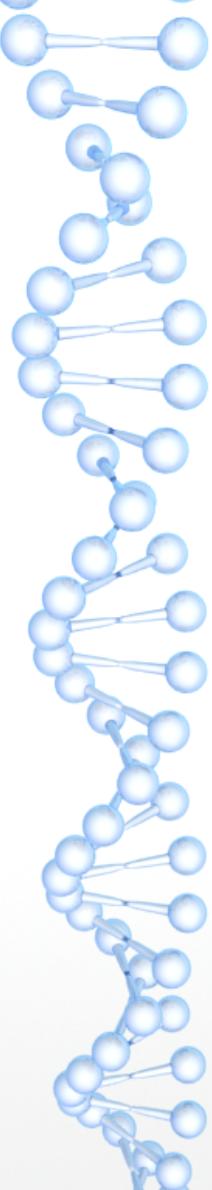
- Git tree:** On the left, a tree view of the repository structure showing files and directories like deaf, core, dbDumps, deaf, logs, static, .gitignore, countries.json, data_class_m..., howToStart.txt, langs.json, lessons-data..., LICENSE, manage.py, nepocujuci_ci..., nepocujuci_ci..., nepocujuci.xmi, README.md, and requirements...
- Rev list:** This pane contains a list of commits. It includes tabs for Graph and Short Log. The Short Log tab is active, showing a list of commits with their authors, dates, and commit messages. One commit, "v0.122 Feature #2209", is highlighted with a red background.
- Code editor:** Below the rev list, there is a large code editor window displaying the Python source code for a file named core/admin.py. The code is shown in a diff format, with changes highlighted in green and red. The code relates to system text push notifications and command-line interface registration.
- Log pane:** On the right, there is a smaller pane showing the log of changes made to core/admin.py, with the commit message "SECURING the external https call (FB API, iTunes); fixed DictionarySe...".

At the bottom of the interface, a status bar displays the tag "Tag: v0.122 [Feature #2209]".



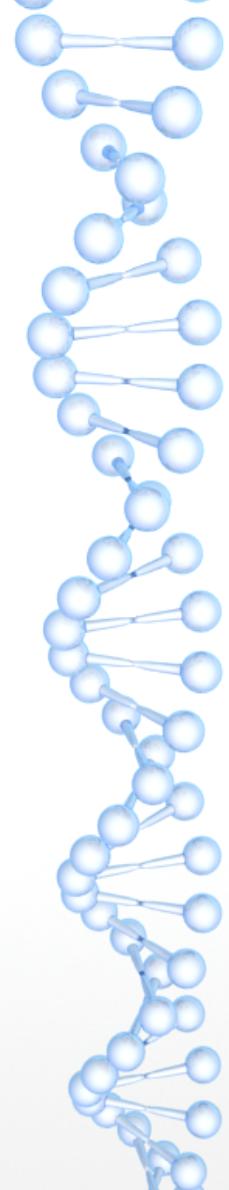
6. Sumár (alebo „Čo už vieme“)

1. V čom je git dobrý
2. Ako vnútorne funguje GIT
3. Ako vytvoriť a naklonovať GIT repozitár
4. Ako komunikovať so vzdialeným repozitárom
5. Ako commitnúť zmeny, pridať súbor, merge-ovať vetvy
6. Ako napraviť najzvyčajnejšie chyby
7. Ktorý GUI nástroj je fajn

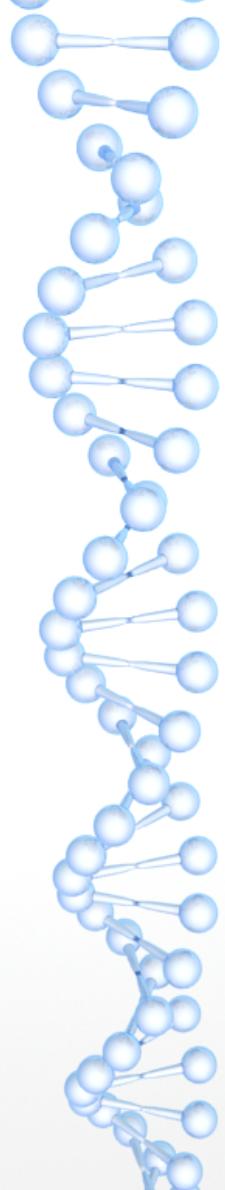


7. Záverom

- Najvhodnejšia príručka ku GIT je *Pro GIT* od Scotta Chacona: je v češtine, používa veľa obrázkových príkladov, ľahko čitateľná, obsahuje takmer celú user dokumentáciu GIT-u
- Git dokáže robiť len *automatizovateľnú* prácu -nemožno od neho čakať „telepatiu“ (vid' Merge-ovanie)
- GUI nástroje sú fajn ;)



Otázky?



Ďakujem za pozornosť!