



Σεμινάριο Git & GitHub

Ημέρα 2

Θέμης Παπαμελετίου

Διονύσης Ζήνδρος

ΕΜΠ 2015

Βοηθοί σήμερα

- Δημήτρης Λαμπρινός
 - Πληροφορική ΑΠΘ, βοηθός
- Νικόλας Κορασίδης
 - ΗΜΜΥ ΕΜΠ, βιντεοσκόπηση & ηχοληψία

Τι θα μάθουμε

- Προχωρημένες τεχνικές git
 - Undo (revert, reset)
 - Blame
 - Tag
 - Cherry pick
 - Rebase (interactive)
- Workflows και συνεργατικές τεχνικές
 - Pull Requests
 - Issues
 - master/develop

Pull requests

- Θέλουμε να κάνουμε push σε ένα repo που δεν είναι δικό μας
 - Έχουμε αναπτύξει κάποια νέα δυνατότητα
 - Έχουμε διορθώσει κάποιο bug
- Έχουμε ήδη κάνει:
 - fork αυτό το repo στο GitHub
 - clone το fork στο σύστημα μας
 - branch για αυτό το feature
 - αλλαγές και commits

Pull requests (PRs)

- Είναι μια αίτηση στον ιδιοκτήτη του repo για pull
- Ζητάμε να κάνει pull (fetch & merge) τις αλλαγές που έχουμε ανεβάσει στο fork μας
- Από ένα δικό μας branch σε ένα δικό του
- Συνήθως στο master του

Βήματα για Pull request

1. Δημοσιεύουμε τις αλλαγές μας στο fork μας
 - `git push origin feature`
 - αυτό δημιουργεί το branch feature από το clone στο fork
2. Δημιουργούμε ένα Pull Request στο GitHub



This repository Search

Pull requests Issues Gist



Oblynx / git-class

Watch 2

A 4h class to teach git and GitHub

42 commits

2 branches

0 releases

3 contributors

Your recently pushed branches:

themicp:feature (5 minutes ago)

Compare & pull request



Branch: master

git-class / +



This branch is 42 commits ahead, 1 commit behind dionyziz:master.

Pull request Compare

Merge pull request #3 from themicp/gh-pages

Oblynx authored 5 hours ago

latest commit ba3cb3c182

images Add NTUA logo, remove shadows from logos 6 days ago

javascripsts Create gh-pages branch via GitHub 12 days ago

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base fork: **Oblynx/git-class** base: **master** ... head fork: **themicp/git-class** compare: **feature**

✓ **Able to merge.** These branches can be automatically merged.



Added a link to the slides of Day 1

Write Preview Markdown supported

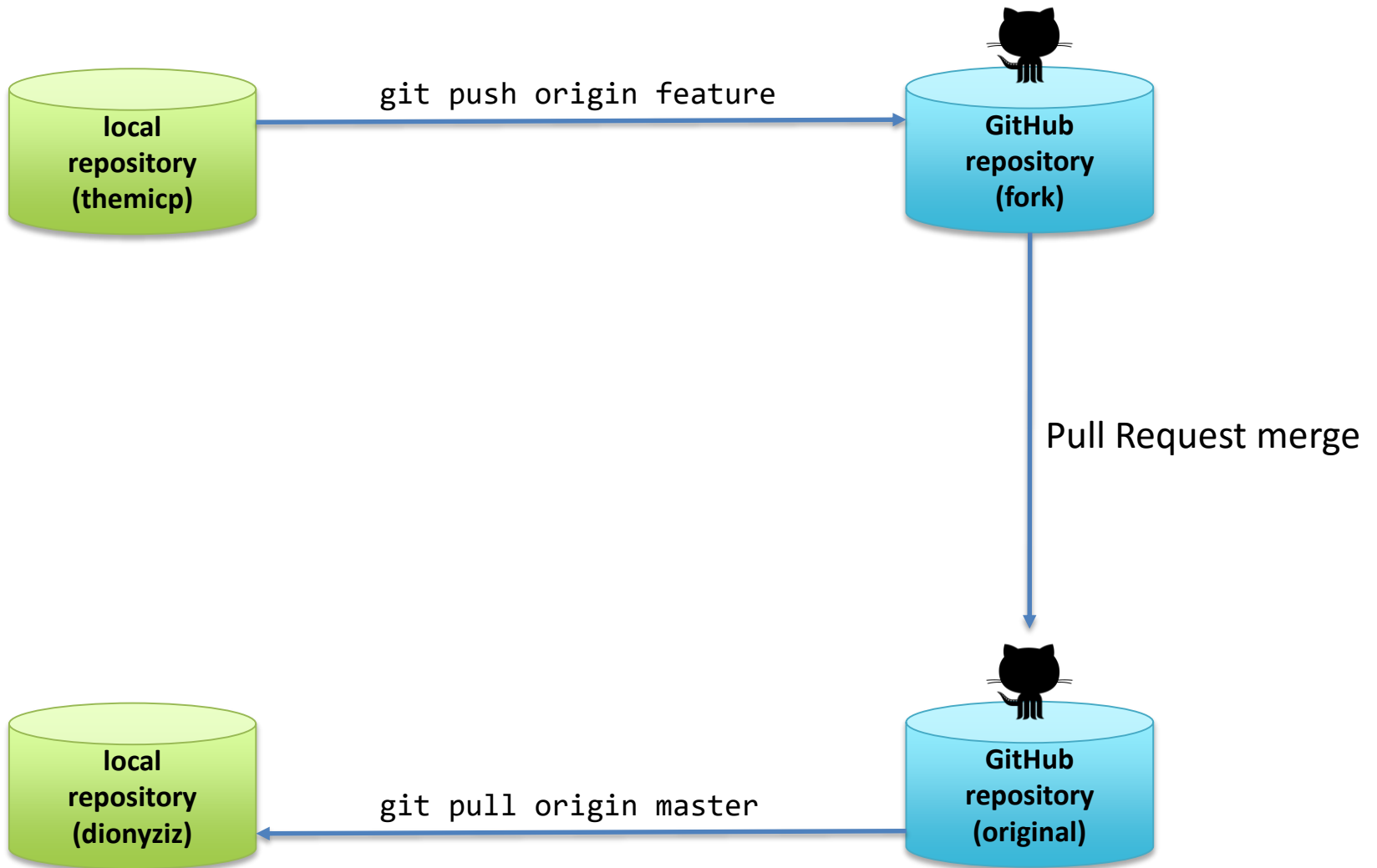
Uploaded the pptx file and added a link in the HTML page.

Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.

Create pull request

Βήματα για Pull request

1. Δημοσιεύουμε τις αλλαγές μας στο fork μας
 - `git push origin feature`
 - αυτό δημιουργεί το branch feature από το clone στο fork
2. Δημιουργούμε ένα Pull Request στο GitHub
3. Ο ιδιοκτήτης του repository κάνει merge το Pull Request μας



Reviews

- Το pull request ελέγχεται για ποιότητα κώδικα πριν γίνει merge
- Αν υπάρχουν προβλήματα, γράφονται σχόλια που αφορούν την ποιότητα του κώδικα
- Ο δημιουργός του pull request διορθώνει τα προβλήματα
 - Κάνοντας push νέα commits στο υπάρχον feature branch του fork του

```
53 +     if (username == '') {
54 +         winston.info('empty');
```



dionyziz added a note 25 days ago

Owner

Use more descriptive error messages.

Add a line note



dionyziz commented on an outdated diff 25 days ago

Hide outdated diff

realtime/tests/server.js



```
@@ -37,8 +37,39 @@ describe('Node Server', function() {
```

```
37     done();
38     });
39     });
40 - });
41 +     it('rejects an empty username', function(done) {
42 +         socket.emit('login', '');
43 +
44 +         socket.on('login-response', function(resp) {
45 +             expect(resp.success).toBe(false);
46 +             expect(resp.error).toBe('empty');
47 +             done();
48 +         });
49 +     });
50 +
51 +
```



dionyziz added a note 25 days ago

Owner

Avoid double blank lines

Add a line note

Ομαδικό workflow

- Έχουμε ένα «κεντρικό» repo
- Όλοι οι άλλοι έχουν forks
- Ο «τελευταίος» κώδικας είναι το master branch του κεντρικού repo
- Κανείς δεν κάνει απευθείας push σε **κανένα** master
- Ούτε ο ιδιοκτήτης του κεντρικού repo!

Ομαδικό workflow

- Για να γίνει μία αλλαγή, ο δημιουργός της αλλαγής κάνει push το feature branch στο fork του
- Κάνει pull request προς το master του κεντρικού repo
- Το pull request γίνεται review από την ομάδα
- Το pull request γίνεται merge
 - Αυτός είναι ο **μόνος** τρόπος που αλλάζει το master

‘upstream’ remote

- Κάθε μέλος της ομάδας προσθέτει το κεντρικό repository ως επιπλέον remote με το όνομα «upstream»
- Άρα κάθε μέλος έχει 2 remotes:
 - “origin”, το δικό τους fork
 - “upstream”, το κεντρικό fork

Ομαδικό workflow

- Συστηματικά, όλα τα μέλη κάνουν pull από το upstream master στο τοπικό τους master
 - Στη συνέχεια κάνουν push στο origin master τους
- Αυτό συγχρονίζει όλα τα master branches (τοπικά και remote)
- Για να συνεργαζόμαστε αποδοτικά, όλα τα μέλη της ομάδας προστίθενται ως “contributors” στο κεντρικό repo
 - Αυτό τους επιτρέπει να κάνουν merge PRs

Το εργαλείο hub

- Επεκτείνει την εντολή 'git' με κάποιες εντολές που αφορούν συγκεκριμένα το GitHub
 - Μπορούμε να κάνουμε pull requests από το CLI
 - Όλα όσα κάνει μπορούμε να τα κάνουμε από το UI του GitHub
- Κατεβάστε το από εδώ:
<https://github.com/github/hub>

master workflow

```
git checkout master
```

```
git pull upstream master
```

```
git push origin master
```

```
git checkout -b feature
```

```
vim && git add && git commit
```

```
git push origin feature
```

```
hub pull-request -b upstream:master
```

Άσκηση

- Κάντε fork στο repo “git-class” του dionyziz στο GitHub
- <https://github.com/dionyziz/git-class>
- Κάντε clone το fork σας
- Φτιάξτε ένα feature branch “birthday”
- Δημιουργήστε ένα αρχείο μέσα στο φάκελο birthdays με τίτλο onoma.eponymo.txt
- Μέσα στο αρχείο γράψτε την ημερομηνία γέννησής σας

Άσκηση

```
git add && git commit
```

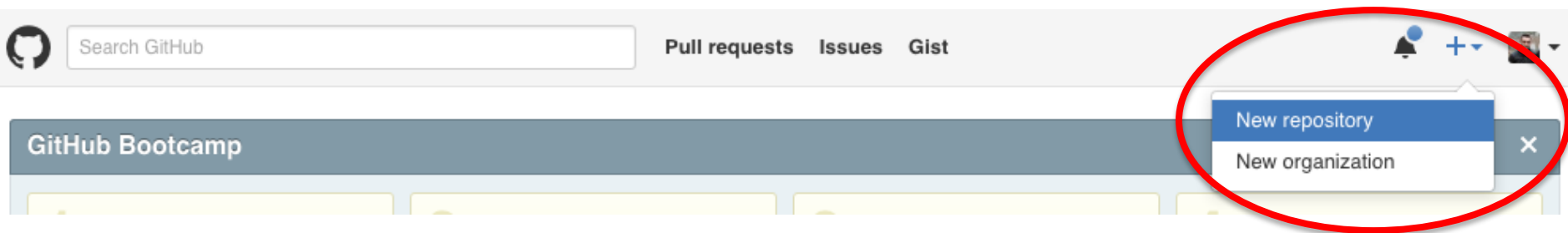
```
hub pull-request -b dionyziz:master
```

(ή μέσω GitHub)

- Βρείτε ένα συνεργάτη
- Ζητήστε του να σας κάνει review
- Κάντε review κάποιον άλλον
- Γράψτε κάποιο σχόλιο στο pull request

Φτιάχνοντας το δικό μας repo


- Μπορούμε εκτός από forks, να φτιάξουμε και δικά μας repos μέσω του GitHub



- Απλώς κάνουμε clone το δικό μας repo για να το έχουμε τοπικά

Owner


Repository name

 themicp ▾ / equanimous-octo-fibula ✓

Great repository names are short and memorable. Need inspiration? How about **equanimous-octo-fibula**.

Description (optional)

 **Public**
Anyone can see this repository. You choose who can commit.

 **Private**
You choose who can see and commit to this repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **None** ▾



Create repository

Πλοήγηση στον γράφο του git

- Υπάρχουν και άλλοι τρόποι να αναφερθούμε σε commits
- HEAD = το commit του τρέχοντος branch
- Αναφορές προγόνων:
Μπορούν να αναφερθούν σε άλλες αναφορές (αναδρομικά)

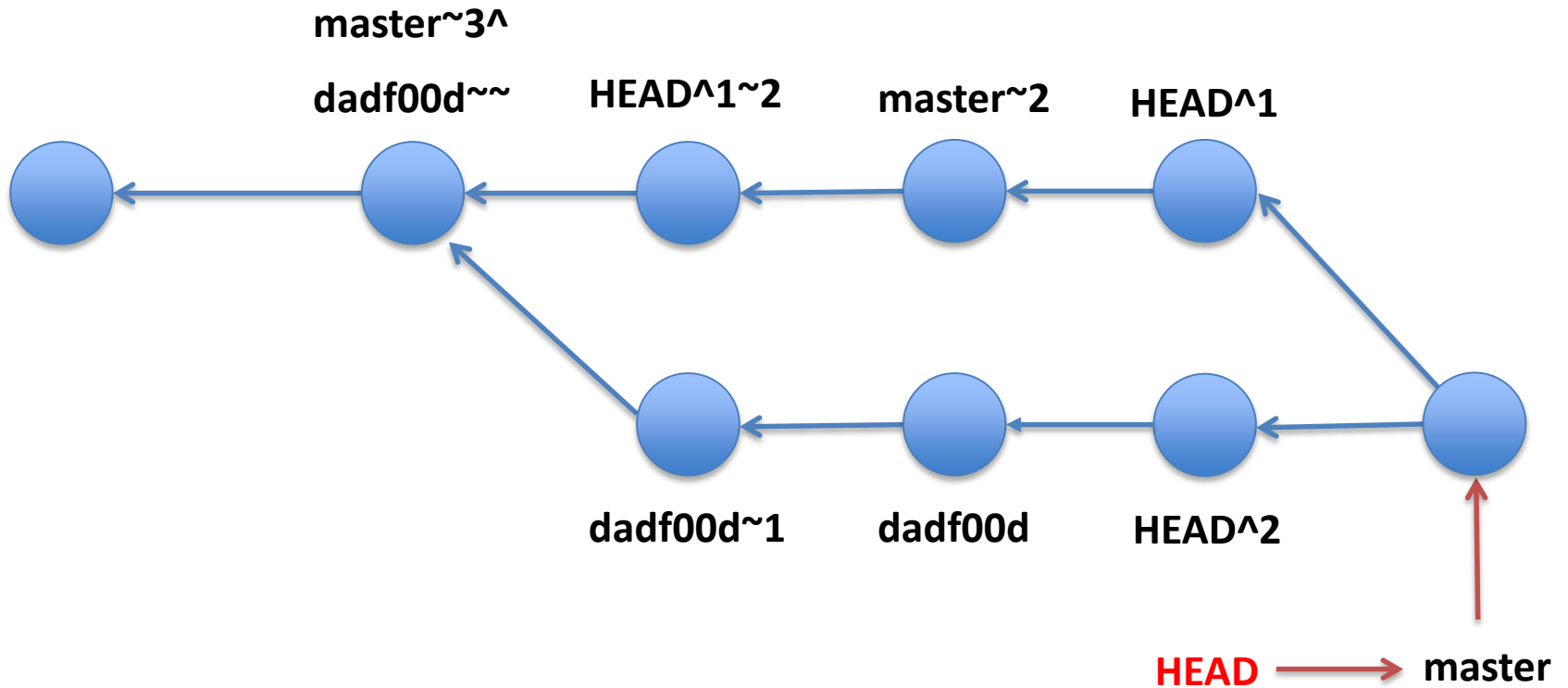
~	Ένα commit πίσω
^	Ο πρώτος από τους γονιούς του commit
~2	Ο παππούς του commit (Δύο commits πίσω, ακολουθώντας τον πρώτο γονέα αν κάποιο είναι merge commit)
^2	Ο δεύτερος από τους γονιούς του commit (που είναι merge commit)

Πλοήγηση στον γράφο του git

- HEAD~2: Δύο commits πίσω
- HEAD^2: Ο δεύτερος γονιός του τωρινού commit (που είναι merge commit)
- 192a812~2: Δύο commits πίσω από το commit 192a812

Quiz: Πού είναι;

HEAD^1, HEAD^2, dadf00d~1, dadf00d~~, master~2, HEAD^1~2, master~3^



GitHub Issues

- Bug tracking σύστημα
- Κάθε issue
 - Είναι ένα bug ή ένα task
 - Είναι ανοιχτό (open) ή κλειστό (closed-resolved)
 - Έχει ένα description
 - Έχει comments
- Issues μπορούν να ανοίξουν όλοι (και μη ιδιοκτήτες του repo)

- Issues
- Pull requests
- Labels
- Milestones

Filters is:issue is:open

New Issue

96 Open ✓ 66 Closed		Author	Labels	Milestones	Assignee	Sort
!	Stop migration procedure in case that .history has wrong permissions					0
	#385 opened on Aug 12 by pkakelas					
!	Write custom error on permissions error in migration UI.					0
	#384 opened on Aug 12 by pkakelas					
!	[!] Memory limit reached. bug					1
	#383 opened on May 17 by nikatlas					
!	Unauthorized after Register. String userID bug					5
	#381 opened on May 17 by nikatlas					
!	Coverage report					0
	#378 opened on Aug 27, 2014 by dionyziz					
!	Attribute generation in HTML helper DRY					0
	#376 opened on Aug 16, 2014 by dionyziz					
!	Beautify forgot password page					0
	#370 opened on Aug 12, 2014 by dionyziz					
!	Remember me checkbox must be clickable through its label on login					0
	#369 opened on Aug 12, 2014 by pkakelas					
!	Username field must be focused by default on login page					0
	#368 opened on Aug 12, 2014 by pkakelas					
!	Make boturl work without http					0
	#367 opened on Aug 12, 2014 by pkakelas					

- <>
- !
- 🔗
- 📄
- 🔊
- 📊

[!] Memory limit reached. #383

New Issue

Open nikatlas opened this issue on May 17 · 1 comment



nikatlas commented on May 17

I created 2 default nodejs bots and ran gamescript.sh.

First of all the script timeout after a minute.

Secondly and most important, from now on there is this error on my home page:

Fatal error: Allowed memory size of 134217728 bytes exhausted (tried to allocate 65552 bytes) in /var/www/html/endofcodes/models/db.php on line 39

Debugging showed that this memory allocations is because it tries to load the last game.

The last game had 131671 rounds (even after timeout)

On Game Model constructor if id is passed the whole game is loading on memory including data for every round and this caused me the memory limit reached exception.

Actually the model load exactly 4439 rounds before crashing and that's only with two bots.

Maybe the game object shouldnt load all the states on memory but only current.

```

endofcodes/models/game.php - Sublime Text (UNREGISTERED)
protected static $attributes = [ 'width', 'height', 'created' ];

public static function getLastGame() {
    if ( $game = dbSelect( 'games', [ 'id' ], [], 'created DESC', 1 ) ) {
        throw new ModelNotFoundException();
    }
    return new Game( $game[ 0 ][ 'id' ] );
}

public function __construct( $id = false, $roundid = false ) {
    require_once 'models/round.php';
    if ( $id != false ) {
        $this->exists = true;
        try {
            $game_info = dbSelectOne( 'games', [ 'created', 'width', 'height' ], compact( 'id' );
        }
        catch ( DBException $e ) {
            throw new ModelNotFoundException();
        }
    }
}

```

Labels

bug

Milestone

No milestone

Assignee

VitSalis

Notifications

Subscribe

You're not receiving notifications from this thread.

3 participants



Αναγνωριστικά GitHub Issues

- Μοναδικό αναγνωριστικό για κάθε issue μέσα στο repo
- Μπορούμε να αναφερθούμε σε αυτό
 - από το commit message:
 - `git commit -m "Refactor code for issue #765"`
 - `git commit -m "Closes #765"` (θα κλείσει το issue 765)
 - από PR descriptions παρομοίως
- Το GitHub μετατρέπει τις αναφορές σε links



bugos commented on May 23, 2014

Collaborator


Why don't we change the controller without reloading the page? We could call `dispatch()` after catching the exception.



dionyziz commented on May 25, 2014

Owner

You mean in regular HTTP pages?

 **bugos** referenced this issue on Jun 13, 2014

Error controller to handle HTTP errors. #351

 Merged



dionyziz closed this in [#351](#) on Jun 21, 2014



Ένα Pull Request που περιέχει αναφορά σε αυτό το issue

Error controller to handle HTTP errors. #351



dionyziz merged 17 commits into `dionyziz:master` from `bugos:error_controllers` on Jun 21, 2014

Conversation 17

Commits 17

Files changed 8



bugos commented on Jun 13, 2014

Collaborator

Fixes [#310](#) and [#257](#).

In this pull request:

- I defined a new `ErrorRedirectException` that can be used instead of `RedirectException` to change the controller without reloading the page with an HTTP redirect.
- I changed the `HTTPErrorException` to extend the new `ErrorRedirectException`. When thrown it changes the controller to the new `HTTPErrorController` as [@VitSalis](#) suggested in [#257](#).
- I refactored the `ControllerBase`.



bugos added some commits on Jun 13, 2014

- Create httperror controller ✓ ffaa5d8
- Minor fix in html helper b5fb004
- Create controllers/httperror.php e85e46d
- Refactor ControllerBase and handle method errors. ✓ d59471c
- Declare ControllerBase variables as protected instead of public. ✗ de1c835
- Create `ErrorRedirectException` and use it to call the error controllers. ✗ 601a27e

Lab

Non

Mile

No r

Ass

No c

Noti

You'
notifi
three

2 pa



git checkout <commit>

- Επαναφέρει το working copy μας στην κατάσταση που ήταν το commit που επιλέξαμε
- Αλλάζει το HEAD ώστε να δείχνει σε αυτό το commit
 - Το HEAD πλέον δεν δείχνει σε κάποιο branch αλλά σε κάποιο commit απευθείας
 - Οπότε είμαστε σε state detached HEAD

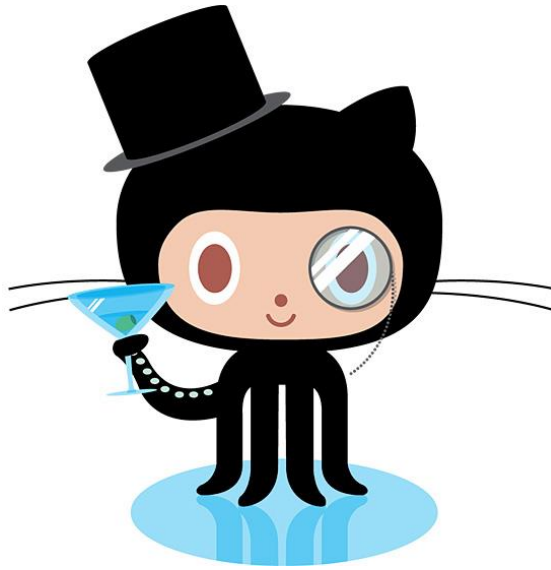
git checkout <commit>

- Δεν μπορούμε να κάνουμε commits σε αυτή τη κατάσταση
- Πρέπει να κάνουμε πάλι checkout σε κάποιο branch
- Χρησιμοποιείται κυρίως για να δούμε πώς ήταν ο κώδικας κάποια στιγμή

git checkout <commit> <file>

- Αλλάζει τα περιεχόμενα του αρχείου file
- Τα φέρνει στην κατάσταση που ήταν στο commit
- Δεν δημιουργεί κάποιο καινούργιο commit
- Δεν κάνει stage

Διάλλειμα



Βρίσκοντας ένα bug

- Εντοπίζουμε ένα bug στον κώδικα μας
- Δεν είμαστε σίγουροι σε ποιο commit δημιουργήθηκε
- Κάνουμε checkout σε προηγούμενα commits για να δούμε πότε εμφανίστηκε αυτό το bug
 - Πάμε σε ένα πολύ παλιό commit στο οποίο ξέρουμε ότι δεν υπήρχε το bug
 - Γνωρίζουμε ότι το bug είναι στο HEAD αυτή τη στιγμή
 - Δοκιμάζουμε όλα τα commits ανάμεσα στο HEAD και στο commit αυτό που γνωρίζουμε ότι δουλεύει

git bisect

- Ένας πιο αποδοτικός τρόπος
- Αφήνουμε το git να το κάνει για εμάς
 1. `git bisect start`
 2. `git bisect bad`
 - Η τωρινή έκδοση (HEAD) είναι buggy
 3. `git bisect good <commit>`
 - Το commit που γνωρίζουμε ότι δεν είναι buggy

git bisect

- Το git θα μας κάνει μια σειρά από ερωτήσεις
 - Θα κάνει checkout κάποιο commit της επιλογής του
 - Θα μας ρωτήσει αν υπάρχει το bug σε αυτό το commit
 - Απαντάμε με
 - git bisect good
 - git bisect bad
- Βρίσκει τελικά το commit που δημιούργησε το bug
- Μπορούμε να ακυρώσουμε τη διαδικασία οποιαδήποτε στιγμή και να επιστρέψουμε στο παλιό HEAD
 - git bisect reset
 - Πρέπει να το τρέξουμε στο τέλος του bisect

git bisect demo

git blame <file>

- Μας δείχνει ποιος ήταν υπεύθυνος για την τελευταία κατάσταση της κάθε γραμμής του αρχείου
- Μας επιτρέπει να βρούμε ποιος δημιούργησε ένα bug

2be7ef36	server.js	(pkakelas	2015-06-22	12:06:08	+0300	1)	var io = require('socket.io');
2be7ef36	server.js	(pkakelas	2015-06-22	12:06:08	+0300	2)	var req = require('request');
568f2e2a	server.js	(Vitalis Salis	2015-06-26	12:32:30	+0300	3)	var fs = require('fs');
78f152a6	server.js	(Dimitris Lamprinos	2015-08-18	18:42:21	+0300	4)	var winston = require('winston');
78f152a6	server.js	(Dimitris Lamprinos	2015-08-18	18:42:21	+0300	5)	
78f152a6	server.js	(Dimitris Lamprinos	2015-08-18	18:42:21	+0300	6)	winston.add(winston.transports.File,
d01536e0	realtime/server.js	(Vitalis Salis	2015-09-03	13:59:22	+0300	7)	winston.level = 'debug';
568f2e2a	server.js	(Vitalis Salis	2015-06-26	12:32:30	+0300	8)	
c365f3d5	realtime/server.js	(Dionysis Zindros	2015-08-29	19:28:02	+0300	9)	var config = JSON.parse(fs.readFileSy
568f2e2a	server.js	(Vitalis Salis	2015-06-26	12:32:30	+0300	10)	
c7ea3af5	realtime/server.js	(Kostis Karantias	2015-08-29	13:57:57	+0300	11)	if (fs.existsSync('../config/local.js
c365f3d5	realtime/server.js	(Dionysis Zindros	2015-08-29	19:28:02	+0300	12)	config = JSON.parse(fs.readFileSy
568f2e2a	server.js	(Vitalis Salis	2015-06-26	12:32:30	+0300	13)	}
568f2e2a	server.js	(Vitalis Salis	2015-06-26	12:32:30	+0300	14)	
568f2e2a	server.js	(Vitalis Salis	2015-06-26	12:32:30	+0300	15)	URL = 'http://' + config.node.hostnam
568f2e2a	server.js	(Vitalis Salis	2015-06-26	12:32:30	+0300	16)	PORT = config.node.port;
2be7ef36	server.js	(pkakelas	2015-06-22	12:06:08	+0300	17)	
2be7ef36	server.js	(pkakelas	2015-06-22	12:06:08	+0300	18)	var socket = io.listen(PORT);
2be7ef36	server.js	(pkakelas	2015-06-22	12:06:08	+0300	19)	
2be7ef36	server.js	(pkakelas	2015-06-22	12:06:08	+0300	20)	var people = {};
78081fe2	server.js	(pkakelas	2015-07-08	06:11:46	+0300	21)	var usernames = {};
4e1257a5	realtime/server.js	(Vitalis Salis	2015-07-16	19:49:15	+0300	22)	var messages_typing = {};
4e1257a5	realtime/server.js	(Vitalis Salis	2015-07-16	19:49:15	+0300	23)	
4e1257a5	realtime/server.js	(Vitalis Salis	2015-07-16	19:49:15	+0300	24)	function getOptions(form, target, met
4e1257a5	realtime/server.js	(Vitalis Salis	2015-07-16	19:49:15	+0300	25)	var headers = {
4e1257a5	realtime/server.js	(Vitalis Salis	2015-07-16	19:49:15	+0300	26)	'User-Agent': 'node-tin
4e1257a5	realtime/server.js	(Vitalis Salis	2015-07-16	19:49:15	+0300	27)	'Content-Type': 'applicat
4e1257a5	realtime/server.js	(Vitalis Salis	2015-07-16	19:49:15	+0300	28)	}
4e1257a5	realtime/server.js	(Vitalis Salis	2015-07-16	19:49:15	+0300	29)	
4e1257a5	realtime/server.js	(Vitalis Salis	2015-07-16	19:49:15	+0300	30)	return {
4e1257a5	realtime/server.js	(Vitalis Salis	2015-07-16	19:49:15	+0300	31)	url: URL + '/api/messages/' +
4e1257a5	realtime/server.js	(Vitalis Salis	2015-07-16	19:49:15	+0300	32)	method: method,
4e1257a5	realtime/server.js	(Vitalis Salis	2015-07-16	19:49:15	+0300	33)	headers: headers,
4e1257a5	realtime/server.js	(Vitalis Salis	2015-07-16	19:49:15	+0300	34)	form: form
4e1257a5	realtime/server.js	(Vitalis Salis	2015-07-16	19:49:15	+0300	35)	}
4e1257a5	realtime/server.js	(Vitalis Salis	2015-07-16	19:49:15	+0300	36)	}
2be7ef36	server.js	(pkakelas	2015-06-22	12:06:08	+0300	37)	

GitHub Blame

- Μας δείχνει τα ίδια πράγματα με το `git blame` απλά με πιο ευανάγνωστο τρόπο και με links στα διάφορα commits

Branch: master ▾

ting / realtime / server.js



 **mbalamat** 15 days ago Change regex to accept greek punctuated words

4 contributors 

198 lines (165 sloc) | 5.99 KB

Raw **Blame** History   

```
1 var io = require('socket.io');
2 var req = require('request');
3 var fs = require('fs');
4 var winston = require('winston');
5
6 winston.add(winston.transports.File, { filename: 'server.log' });
7 winston.level = 'debug';
```


GitHub show

- Μας δείχνει τα ίδια πράγματα με το `git show` αλλά με πιο ευανάγνωστο τρόπο

Fix watchify by enabling poll option

[Browse files](#)

substack/watchify#216 (comment)


🔗 master (#120)

 gtklocker authored 28 days ago

1 parent [2fdb8e1](#) commit [b726c4d468aed786b534b47741c9a8aca0b812b7](#)

📄 Showing 1 changed file with 3 additions and 1 deletion.

Unified Split

4  client/gulpfile.js

View

⚙ @@ -20,7 +20,9 @@ function act/watch) {

20 fullPaths: true

21 });

22 if (watch) {

23 - b = watchify(b);

24 }

25 b.transform(function(file) {

26 return reactify(file, {

20 fullPaths: true

21 });

22 if (watch) {

23 + b = watchify(b, {

24 + poll: true

25 + });

26 }

27 b.transform(function(file) {

28 return reactify(file, {

Conflicts

- Όταν κάνουμε merge, μπορεί να μην μπορεί να γίνει αυτόματα, π.χ.
 - ένας προγραμματιστής άλλαξε ένα αρχείο, ενώ ένας άλλος το διέγραψε
 - δύο άτομα άλλαξαν το ίδιο σημείο ενός αρχείου (διόρθωσαν το ίδιο bug με διαφορετικό τρόπο)
- Το git δεν ξέρει ποια είναι η σωστή αλλαγή
- Τότε μας λέει ότι υπάρχουν “conflicts” (διενέξεις)

```
$ vim realtime/server.js
themicp@Berners ~/www/ting | feature •
$ git add realtime/server.js
themicp@Berners ~/www/ting | feature +
$ git commit -m 'Rename server.log to realtime.log'
[feature c4246d4] Rename server.log to realtime.log
1 file changed, 1 insertion(+), 1 deletion(-)
themicp@Berners ~/www/ting | feature
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)
themicp@Berners ~/www/ting | master
$ git merge feature
Auto-merging realtime/server.js
CONFLICT (content): Merge conflict in realtime/server.js
Automatic merge failed; fix conflicts and then commit the result.
✘ themicp@Berners ~/www/ting | master •+ >M<
$
```


Επίλυση conflicts

- Το git έχει αλλάξει το αρχείο μας
 - Ενοποιεί όλες τις αλλαγές που μπορεί αυτόματα
 - Τις υπόλοιπες τις αφήνει σ' εμάς
- Ανοίγουμε το αρχείο που έχει conflicts με τον editor μας
- Ψάχνουμε για <<<<< ===== >>>>>

```
1 var io = require('socket.io');
2 var req = require('request');
3 var fs = require('fs');
4 var Winston = require('Winston');
5 Στο master άλλαξε το winston σε κεφαλαία
6 <<<<<<< HEAD
7 Winston.add(Winston.transports.File, { filename: 'server.log' });
8 Winston.level = 'debug';
9 =====
10 winston.add(winston.transports.File, { filename: 'realtime.log' });
11 winston.level = 'debug';
12 >>>>>>> feature Στο feature άλλαξε το όνομα αρχείου σε realtime.log
13
```

```
1 var io = require('socket.io');
2 var req = require('request');
3 var fs = require('fs');
4 var Winston = require('Winston');
```

Η έκδοση στο master πριν το merge

```
5
6 <<<<<< HEAD
```

```
7 Winston.add(Winston.transports.File, { filename: 'server.log' });
8 Winston.level = 'debug';
```

```
9 =====
```

```
10 winston.add(winston.transports.File, { filename: 'realtime.log' });
```

```
11 winston.level = 'debug';
```

```
12 >>>>>> feature
```

```
13
```

Η έκδοση στο feature πριν το merge

Επίλυση conflicts

- Αντικαθιστούμε το τμήμα με το σωστό κώδικα
 - σε όλα τα τμήματα και αρχεία που έχουν conflicts
- Κάνουμε git add κάθε αρχείο
- **Προσοχή:** Δεν κάνουμε άλλες αλλαγές στα αρχεία πέρα από την επίλυση conflicts
- Κάνουμε git commit
 - Έτσι δημιουργείται το merge commit
 - Ο τελικός γράφος είναι ο ίδιος σαν να μην είχαμε conflicts

```
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
You have unmerged paths.
  (fix conflicts and run "git commit")

Unmerged paths:
  (use "git add <file>..." to mark resolution)

        both modified:   realtime/server.js

no changes added to commit (use "git add" and/or "git commit -a")
themicp@Berners ~/www/ting ⌕ master •+ >M<
$ git add realtime/server.js
themicp@Berners ~/www/ting ⌕ master + >M<
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:

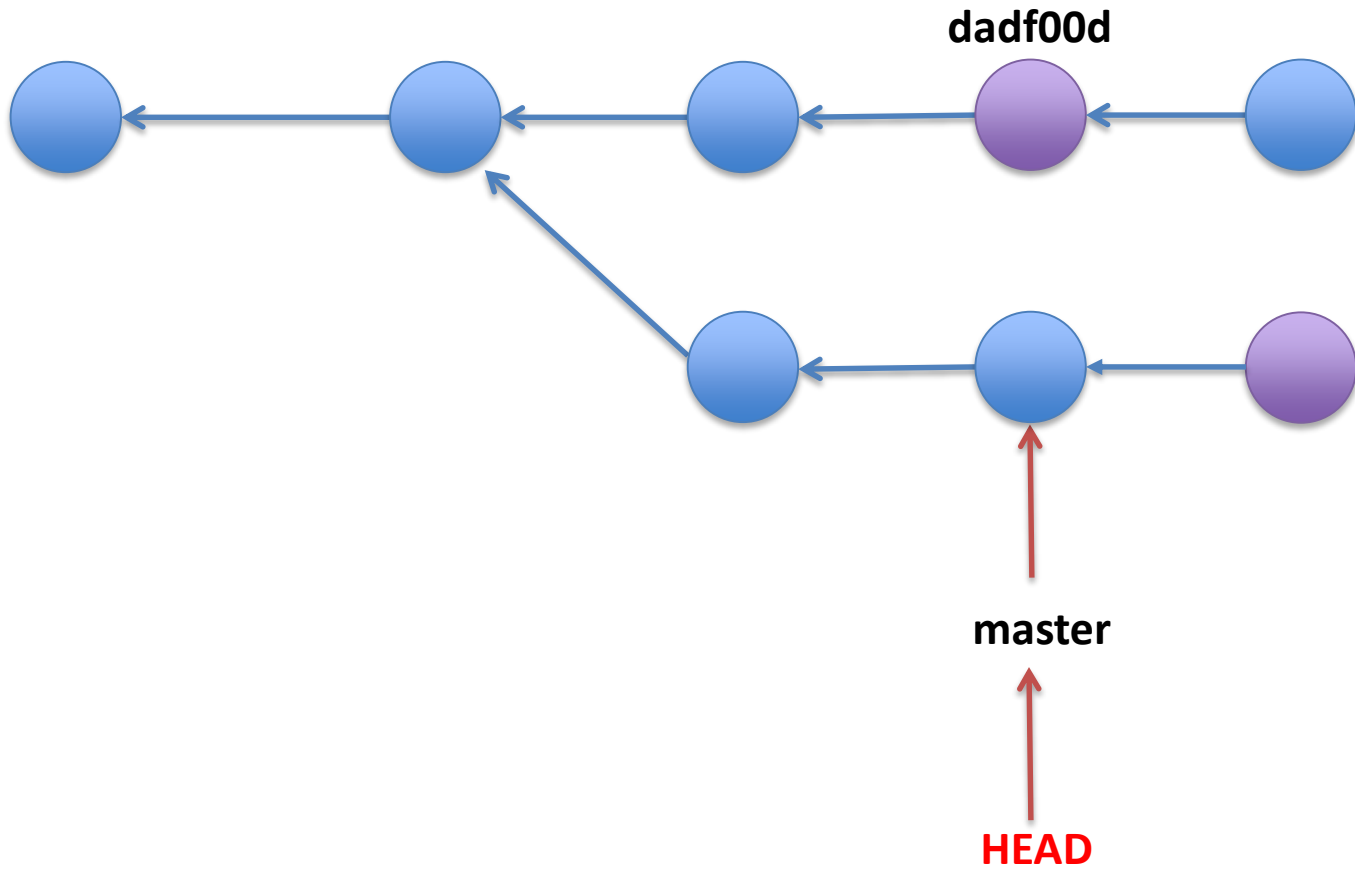
        modified:   realtime/server.js

themicp@Berners ~/www/ting ⌕ master + >M<
$ git commit
[master 680b5f2] Merge branch 'feature'
themicp@Berners ~/www/ting ⌕ master
$
```

git cherry-pick <commit>

- Αντιγράφει ένα commit από κάποιο σημείο του γράφου
- Το τοποθετεί πάνω στο ενεργό branch
- Δηλαδή δημιουργεί ένα νέο commit, αντίγραφο του επιλεγμένου
- Το νέο commit που δημιουργείται έχει:
 - Καινούργιο αναγνωριστικό
 - Τις ίδιες ακριβώς αλλαγές στα αρχεία
 - Την ίδια περιγραφή

(master) git cherry-pick
dadf00d



git revert <commit>

- Παρόμοιο με το cherry-pick
- Μόνο που εφαρμόζει το 'αντίστροφο' ενός commit
- Αντιγράφει τις αλλαγές που έκανε ένα <commit> σε ένα νέο commit
- Το νέο commit περιέχει τις **αντίστροφες** αλλαγές
 - add αντί για delete
 - delete αντί για add
 - προσθήκη γραμμών αντί για αφαίρεση
 - κλπ.
- Το χρησιμοποιούμε για να ακυρώσουμε ένα commit που έχουμε δημοσιεύσει ήδη


```
commit fd2c7b639ac530e015953052eec048971209a478
```

```
Author: Themis Papameletiou <themicp@gmail.com>
```

```
Date: Wed Oct 7 15:39:10 2015 +0300
```

```
Revert "Changed README file"
```

```
This reverts commit cefc5d3973b3c17eb8dd2ab46af119f3b2a7b1ef.
```

```
commit cefc5d3973b3c17eb8dd2ab46af119f3b2a7b1ef
```

```
Author: Themis Papameletiou <themicp@gmail.com>
```

```
Date: Mon Oct 5 17:24:29 2015 +0300
```

```
Changed README file
```

Αρχικό commit

```
commit cefc5d3973b3c17eb8dd2ab46af119f3b2a7b1ef
Author: Themis Papameletiou <themicp@gmail.com>
Date: Mon Oct 5 17:24:29 2015 +0300
```

Changed README file

```
diff --git a/README b/README
index af5626b..6892cea 100644
--- a/README
+++ b/README
@@ -1,1 @@
-Hello, world!
+Hello, GitHub!
(END)
```

Αντίστροφο commit (revert)

```
commit fd2c7b639ac530e015953052eec04897120
Author: Themis Papameletiou <themicp@gmail.com>
Date: Wed Oct 7 15:39:10 2015 +0300
```

Revert "Changed README file"

This reverts commit cefc5d3973b3c17eb8

```
diff --git a/README b/README
index 6892cea..af5626b 100644
--- a/README
+++ b/README
@@ -1,1 @@
-Hello, GitHub!
+Hello, world!
(END)
```

git tag

- Θέλουμε να ξεχωρίζουμε κάποιες σημαντικές καταστάσεις στο ιστορικό
 - Release 1.0
 - Release 2.0
- `git tag -a <tag>`
 - Προσθέτει ένα tag στο τελευταίο commit του branch που βρισκόμαστε
- `git tag -a <tag> <commit>`
 - Προσθέτει ένα tag στο commit που επιλέξαμε
- `git tag`
 - Δείχνει τη λίστα με τα tags στο repo
- `git tag -d <tag>`
 - Διαγράφει ένα tag

Δημοσιεύοντας tags

- `git push <remote> <tag>`
 - Δημοσιεύει το `<tag>` στο `<remote>`
- `git fetch --tags <remote>`
 - Κατεβάζει όλα τα tags από το `<remote>`

git add -p

- Μας επιτρέπει να προσθέσουμε μέρος ενός αρχείου στο staging area
- Είναι διαδραστικό: Μας ρωτάει για κάθε αλλαγή σε κάθε αρχείο
- Παίρνει παράμετρο αρχείο
 - ή χωρίς παραμέτρους μας ρωτάει για όλα τα αρχεία που έχουν αλλάξει
- Μας επιτρέπει να κάνουμε καθαρότερα commits

```
$ vim README.md
themicp@Berners ~/www/ting master
$ vim client/ting.jsx
themicp@Berners ~/www/ting master
$ git add -p
diff --git a/client/ting.jsx b/client/ting.jsx
index 9121839..0b4e1b9 100644
--- a/client/ting.jsx
+++ b/client/ting.jsx
@@ -16,7 +16,7 @@ const Ting = React.createClass({
    this.refs.messageForm.onLogin(username, people);
    this.refs.userList.onLogin(username, people);

-    $.getJSON('/api/messages/' + this.state.channel, (messages) => {
+    $.getJSON('/api/history/' + this.state.channel, (messages) => {
        const history = _.indexBy(messages, 'id');

        this.refs.history.onHistoricalMessagesAvailable(history);
Stage this hunk [y,n,q,a,d,/,e,?]? █
```

git reflog

- Μία λίστα στην οποία αποθηκεύονται όλα όσα κάνουμε με τη σειρά
 - Πού ήταν το HEAD κάθε στιγμή
- Κάθε commit που γίνεται μπαίνει εκεί σειριακά
- Την εμφανίζουμε με `git reflog`

```
6656470 HEAD@{0}: commit (amend): Add basic queue structure
3b19f12 HEAD@{1}: checkout: moving from 3b19f120f2c73df3ceba5d4c84f0bc2980296281 to typing_queue
3b19f12 HEAD@{2}: checkout: moving from typing_queue to 3b19f120f2c73df3ceba5d4c84f0bc2980296281
3b19f12 HEAD@{3}: commit: Add basic queue structure
6537bae HEAD@{4}: checkout: moving from master to typing_queue
6537bae HEAD@{5}: pull origin master: Fast-forward
22e8d43 HEAD@{6}: checkout: moving from http_endpoint_arch to master
676bb05 HEAD@{7}: checkout: moving from more_client_tests to http_endpoint_arch
29965f7 HEAD@{8}: commit: Add tests for typing and self messages
22e8d43 HEAD@{9}: checkout: moving from typing_animation to more_client_tests
40b65ac HEAD@{10}: commit (amend): Add typing bubble animation
788a43e HEAD@{11}: checkout: moving from more_client_tests to typing_animation
22e8d43 HEAD@{12}: checkout: moving from master to more_client_tests
22e8d43 HEAD@{13}: checkout: moving from more_client_tests to master
22e8d43 HEAD@{14}: checkout: moving from master to more_client_tests
22e8d43 HEAD@{15}: checkout: moving from typing_animation to master
788a43e HEAD@{16}: commit (amend): Add typing bubble animation
7ef693f HEAD@{17}: commit: Add typing bubble animation
22e8d43 HEAD@{18}: checkout: moving from master to typing_animation
```


git reflog <branch>

- Δείχνει από πού πέρασε αυτό το branch
- `git reflog --date=relative`
 - Μας δείχνει σχετικό χρόνο για την κάθε αλλαγή

6537bae	master@{6 days ago}: pull origin master: Fast-forward
22e8d43	master@{13 days ago}: pull origin master: Fast-forward
7717a96	master@{2 weeks ago}: pull origin master: Fast-forward
2fdb8e1	master@{4 weeks ago}: pull origin master: Fast-forward
cc2b753	master@{4 weeks ago}: pull origin master: Fast-forward
867f3f0	master@{4 weeks ago}: pull: Fast-forward
fb62906	master@{4 weeks ago}: pull origin master: Fast-forward
a032c3b	master@{5 weeks ago}: pull origin master: Fast-forward
19df1be	master@{5 weeks ago}: pull origin master: Fast-forward
52fee3e	master@{5 weeks ago}: pull origin master: Fast-forward
2705708	master@{5 weeks ago}: pull origin master: Fast-forward
018cbc9	master@{5 weeks ago}: pull origin master: Fast-forward
7a95969	master@{6 weeks ago}: pull origin master: Fast-forward

Ξαναγράφοντας το ιστορικό

- Το git μας επιτρέπει να ξαναγράψουμε την ιστορία
- Μπορούμε να αλλάξουμε commits
 - Την σειρά με την οποία εφαρμόστηκαν
 - Το τι αλλαγή έκανε το καθένα
 - Τις περιγραφές τους

Ξαναγράφοντας το ιστορικό

- Αυτό πρέπει να γίνεται πολύ προσεκτικά
- Κανόνας της πιάτσας:

Μην ξαναγράφεις το ιστορικό σε commits που μπορεί να έχουν κάνει pull άλλοι

git commit --amend

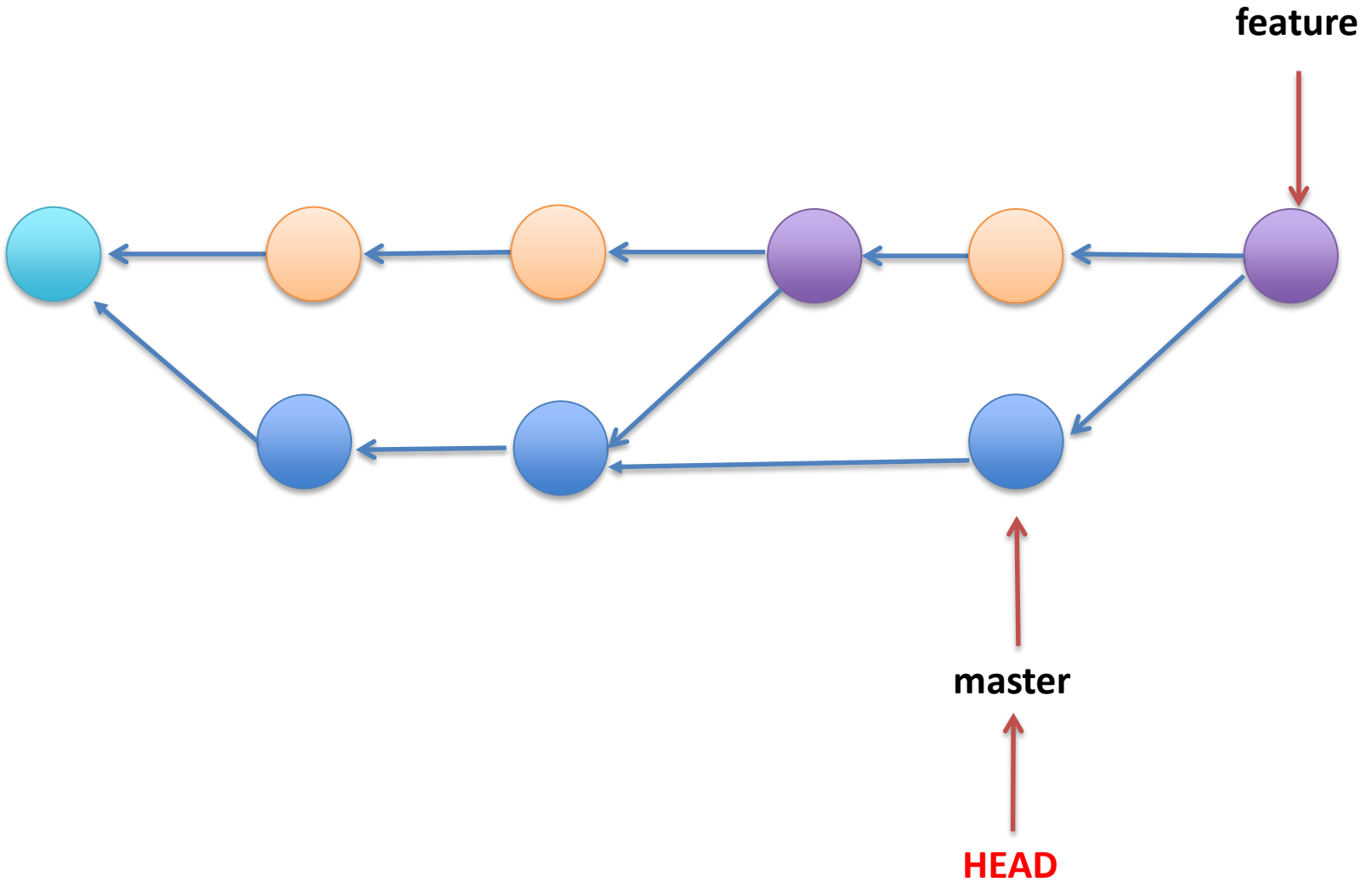
- Αλλάζει το πιο πρόσφατο commit
- Προσθέτει τα πράγματα που έχουμε ετοιμάσει για commit (staging)
- Μας ρωτάει για αλλαγή στο commit message
 - εκτός αν περάσουμε --no-edit

Back-merging

- Όσο δουλεύουμε σε ένα feature branch μπορεί να έχουν γίνει αλλαγές στο master
- Θέλουμε να δουλεύουμε στη νεότερη έκδοση του κώδικα
- Γι' αυτό μπορούμε στο feature να τρέξουμε:
 - `git merge master`
- Επιλύουμε conflicts
- Συνεχίζουμε να δουλεύουμε στον νεότερο κώδικα
- Κάνουμε κι άλλα commits στο feature

Merge workflow

```
git checkout master  
git pull upstream master  
git push origin master  
git checkout feature  
git merge master  
git push origin feature  
hub pull-request
```



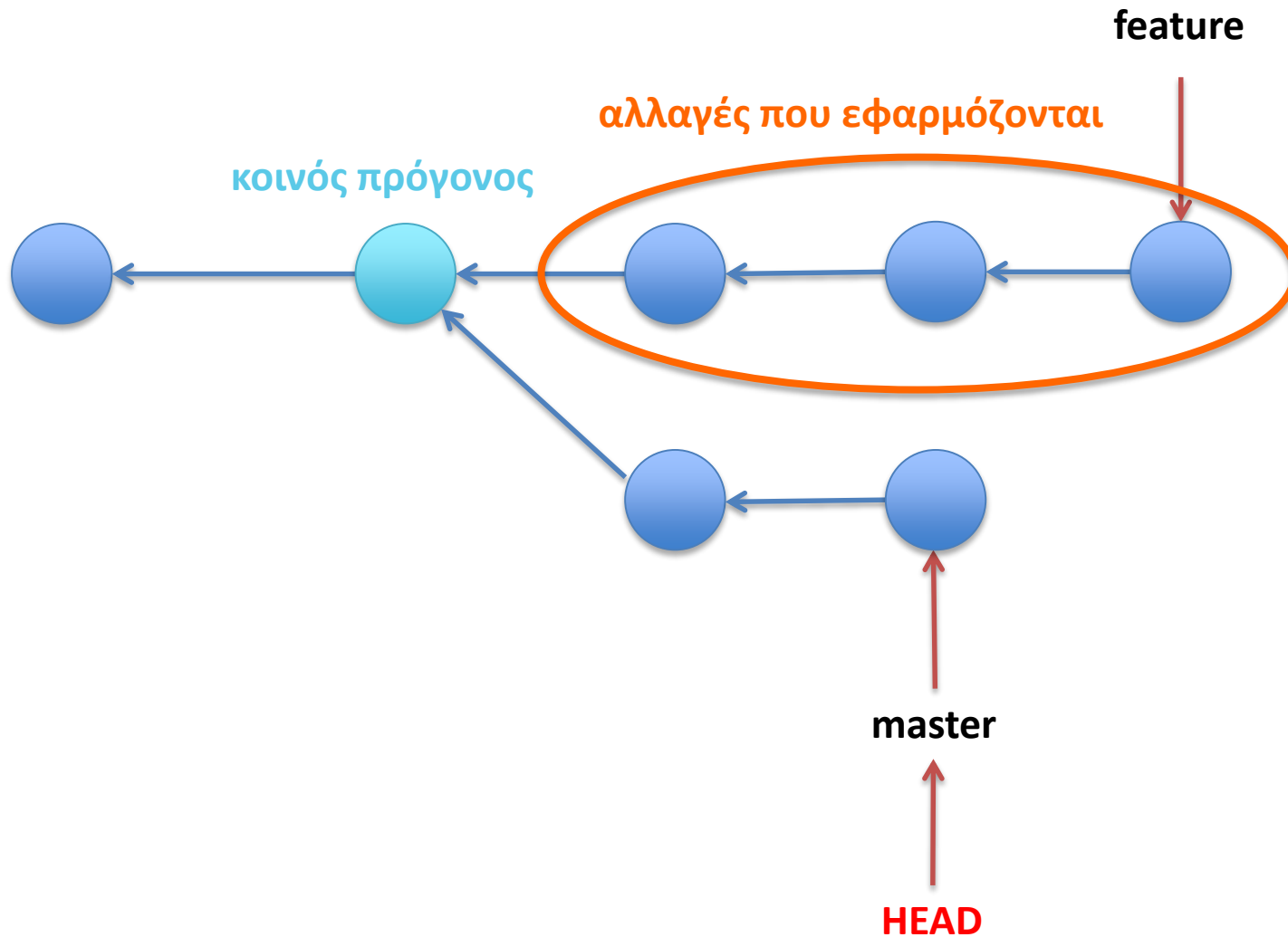
git rebase

- Παρόμοιο με το git merge
- Εφαρμόζεται ανάμεσα σε 2 branches
 - Το branch στο οποίο βρισκόμαστε (π.χ. feature)
 - Το branch πάνω στο οποίο κάνουμε rebase (π.χ. master)

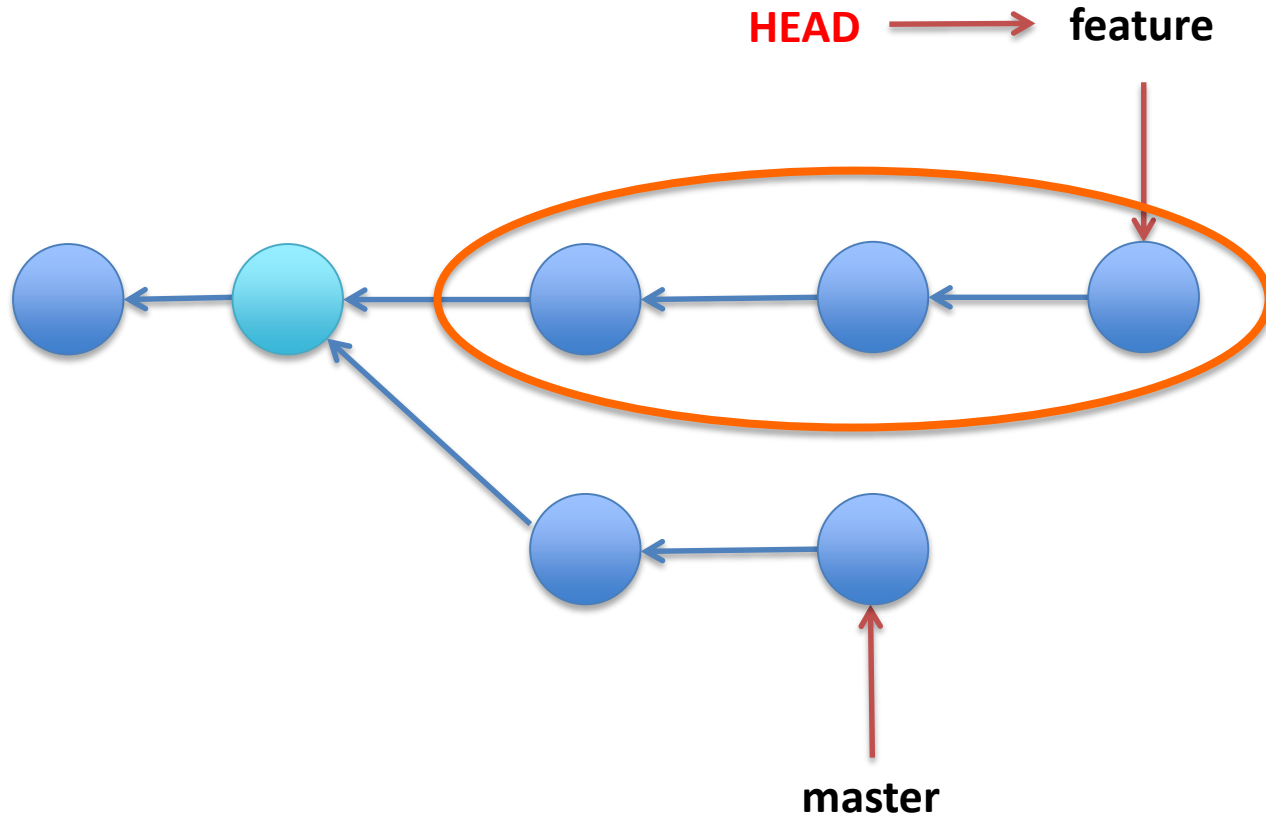
Αλγόριθμος rebase

- Παρομοίως με το merge
 - Βρίσκει τον πιο πρόσφατο κοινό πρόγονο
 - Βρίσκει το diff ανάμεσα:
 - Στον πιο πρόσφατο κοινό πρόγονο
 - Στο branch στο οποίο βρισκόμαστε (feature)
- Εφαρμόζει όλα τα commits που βρήκε πάνω στο branch στο οποίο γίνεται rebase
 - Το αποτέλεσμα αποθηκεύεται στο branch που βρισκόμαστε (feature)
- Το branch πάνω στο οποίο γίνεται rebase (master) δε μεταφέρεται

(feature) git rebase master



(feature) git rebase master



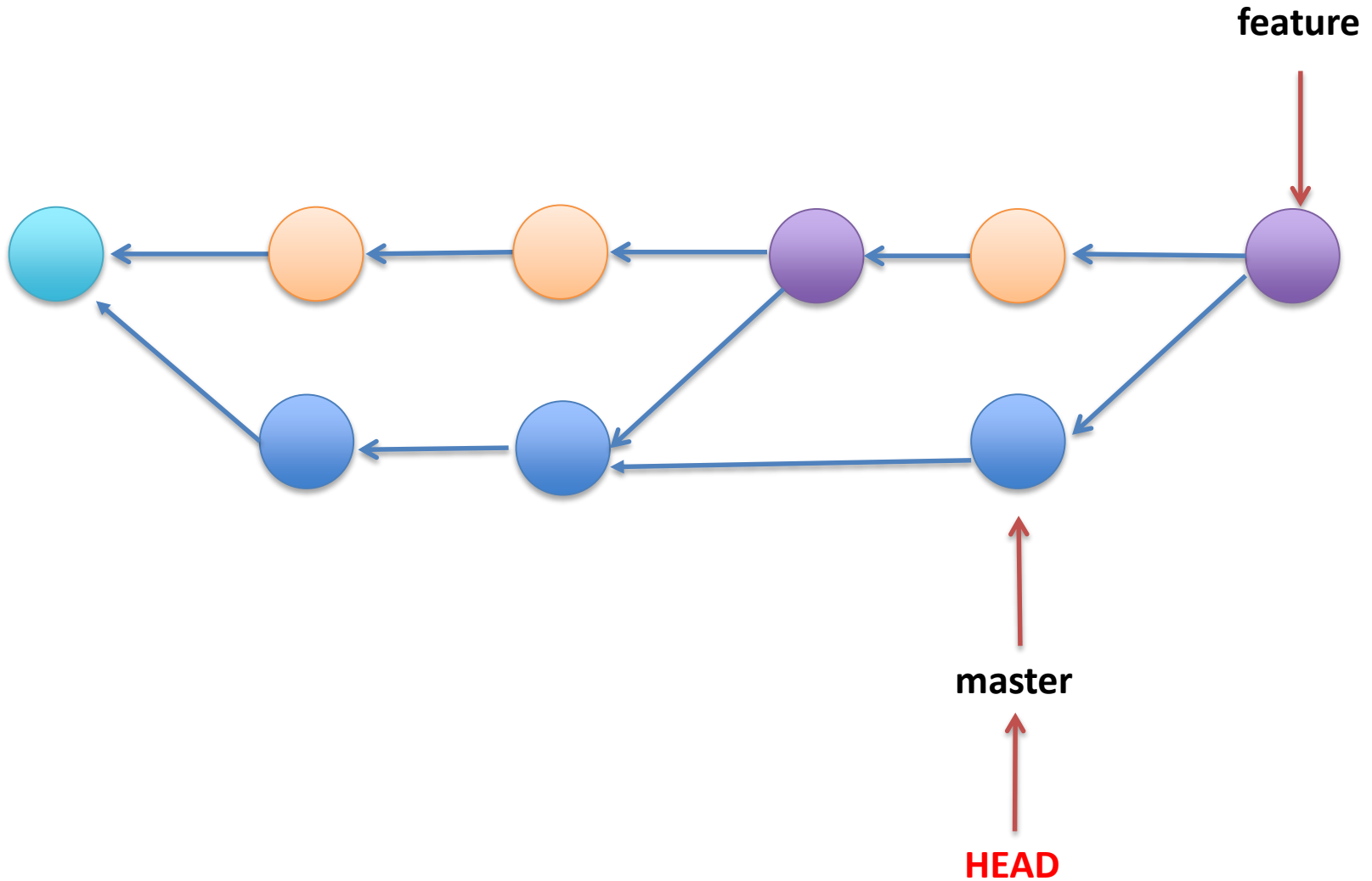
Χρησιμότητα rebase

- «Αλλάζει τη βάση» στην οποία έχει φτιαχτεί ένα branch
- Επαναγράφει το ιστορικό
- Χρησιμεύει στο να έχουμε καθαρότερο ιστορικό
- Μετά από rebase μπορούμε να κάνουμε καθαρό pull request

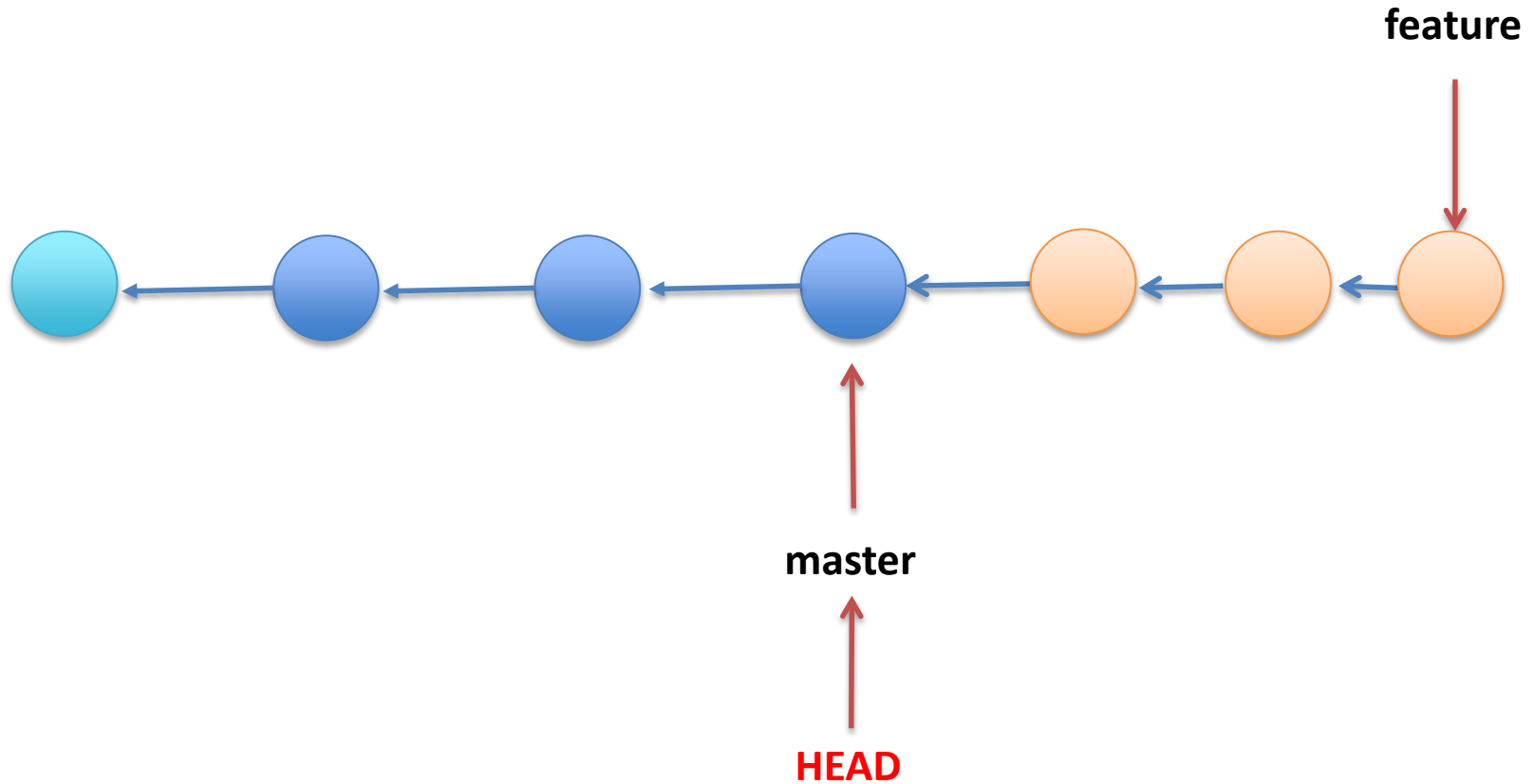
Merge VS Rebase

- Σας προτείνουμε να κάνετε rebase αντί για merge
- merge να χρησιμοποιείτε μόνο για το τελικό pull request όταν ενοποιείται ο κώδικας του feature branch

Γράφος με merge



Γράφος με rebase



Conflicts σε rebase

- Κατά τη διάρκεια του rebase μπορεί να προκύψουν conflicts
- Αυτό σημαίνει ότι:
 - Το branch στο οποίο βρισκόμαστε (feature) έχει commits που δεν μπορούν να εφαρμοστούν στο branch στο οποίο γίνεται rebase (master)
 - Διότι το master εντωμεταξύ έχει αλλάξει
- Επιλύουμε τα conflicts κατά τα γνωστά
 - Μετά την επίλυση:
 - `git add && git rebase --continue`

Rebase workflow

```
git checkout master  
git pull upstream master  
git push origin master  
git checkout feature  
git rebase master  
git push origin feature  
hub pull-request
```

git rebase -i

- Το rebase είναι το πιο δυνατό git command
- Φημίζεται ότι όλες οι εντολές του git μπορούν να γραφτούν σαν rebase με κατάλληλες παραμέτρους ;)
- Με την παράμετρο -i επιλέγουμε να κάνουμε rebase δυναμικά

Δυναμικό rebase

- Επιτρέπεται να:
 - Αλλάξουμε τη σειρά που θα εφαρμοστούν τα commits
 - Αλλάξουμε τα μηνύματα των commits
 - Ενώσουμε πολλαπλά commits σε ένα (squash/fixup)
 - Σπάσουμε ένα commit σε πολλά

rebase -i demo

```
1 pick 00f8037 Remove trailing whitespace
2 pick be98838 Add client-side part of typing
3 fixup 2857e78 Make minUpdate a constant and remove debugging messages
4 pick 4bc382c Add node part of typing
5 fixup abc9436 Add typing to the POST request on server.js
6 reword 481cfcb Add typing and rename datetime to datetime_start, refactor the tests accordingly
7 fixup 8ae2a35 Remove useless migrations and add a new one
8 reword f8ed578 Add django-side support for typing
9 fixup 8c23dbd Make further checks about typing being false
10 reword 3639743 Add django-side message deletion support
11 pick e3c5822 Make typing messages less opaque
12 edit 049731b Add node-side support for typing
13 pick e9b6cff Use cleaned_data and objects.create()
14 pick 6ff5140 Remove useless migrations
15 pick 3df83f8 Make typing not required and make forms return the message on save()
16 pick 1965e56 Use valid constant names, add some tests
17 pick 6cd5915 Add clean_datetime_start function
18 pick d17c944 Make to_dict on Message model cooler
19 pick 5b2462c Remove uselss ast
20 pick 5302275 Make the view return the datetime_sent too on GET
21 pick b11c3e5 Remove yoda code
22 pick b62ff1b Use each instead of for
23 pick 7e99d44 Do not allow empty messages
24 pick 8f0ab99 Fix typo
25 pick a8619e0 Add datetime_sent migration
26 pick f8269bd Rename the datetime_sent migration
27 pick 5d5df60 Add comment on testing
28 pick b99fe59 Remove useless if statement
29 pick 8899ade Create custom client class
30 pick d6f02e3 Delete old migrations and create a squashed migration
31 pick a4ef69b Merge conflicting migrations
32 pick 50cc5d0 Fix some bugs after rebase
33 pick 23e7800 Make sure that persistent messages can not be patched
34
35 # Rebase 4d178aa..23e7800 onto 4d178aa
36 #
37 # Commands:
38 # p, pick = use commit
39 # r, reword = use commit, but edit the commit message
40 # e, edit = use commit, but stop for amending
41 # s, squash = use commit, but meld into previous commit
42 # f, fixup = like "squash", but discard this commit's log message
43 # x, exec = run command (the rest of the line) using shell
44 #
45 # These lines can be re-ordered; they are executed from top to bottom.
46 #
47 # If you remove a line here THAT COMMIT WILL BE LOST.
48 #
49 # However, if you remove everything, the rebase will be aborted.
50 #
51 # Note that empty commits are commented out
```

git reset <commit>

- Γυρνάει το τρέχον branch στο <commit>
- Ακυρώνει όλες τις αλλαγές εντωμεταξύ
- Συνήθης χρήση:
 - Ακύρωση του πιο προσφατου commit
 - `git reset HEAD~`
- Μην το κάνετε για δημοσιευμένα commits!
 - Φτιάξτε ένα νέο με `git revert`

git push --force-with-lease

- Σε περίπτωση που αλλάξουμε το ιστορικό, το git θα αρνηθεί να κάνει push
- Γι' αυτό κάνουμε
git push --force-with-lease
- ή αν δεν είναι διαθέσιμο
– git push --force

**PUSH REJECTED, REBASE OR
MERGE**

GIT PUSH --FORCE

master/develop workflow

- Σε **μεγάλα** open source projects, προτείνουμε τον τρόπο εργασίας master/develop
 - Αρκετά γραφειοκρατικός
 - Δεν τον προτείνουμε για μικρά projects
- Παρόμοιος με το master workflow
- Στο main repo έχουμε 2 branches: master & develop
- master = stable έκδοση, για το ευρύ κοινό
- develop = unstable έκδοση

master/develop workflow

- Όλη η ομάδα έχει forks
- **Κανείς** δεν κάνει push στα master & develop
- Κάθε αλλαγή γίνεται **με pull request** μέσω του αντίστοιχου fork
- Κάθε αλλαγή γίνεται σε feature branch που έχει προκύψει από το develop
- Τα pull requests γίνονται στο develop branch του main repo

master/develop workflow

git checkout develop

git pull upstream develop

git push origin develop

git checkout -b feature

vim && git add && git commit

git push origin feature

hub pull-request -b upstream:develop

master/develop workflow

- Τα PRs στο develop γίνονται review
- Διορθώσεις για τα PRs με βάση τα reviews γίνονται κανονικά push στο αντίστοιχο feature branch
- Τα PRs εν τέλει γίνονται merge στο develop
- Όλοι οι developers κάνουν pull στο clone τους από το develop του upstream
- Διατηρούν το develop branch του origin τους συγχρονισμένο με push

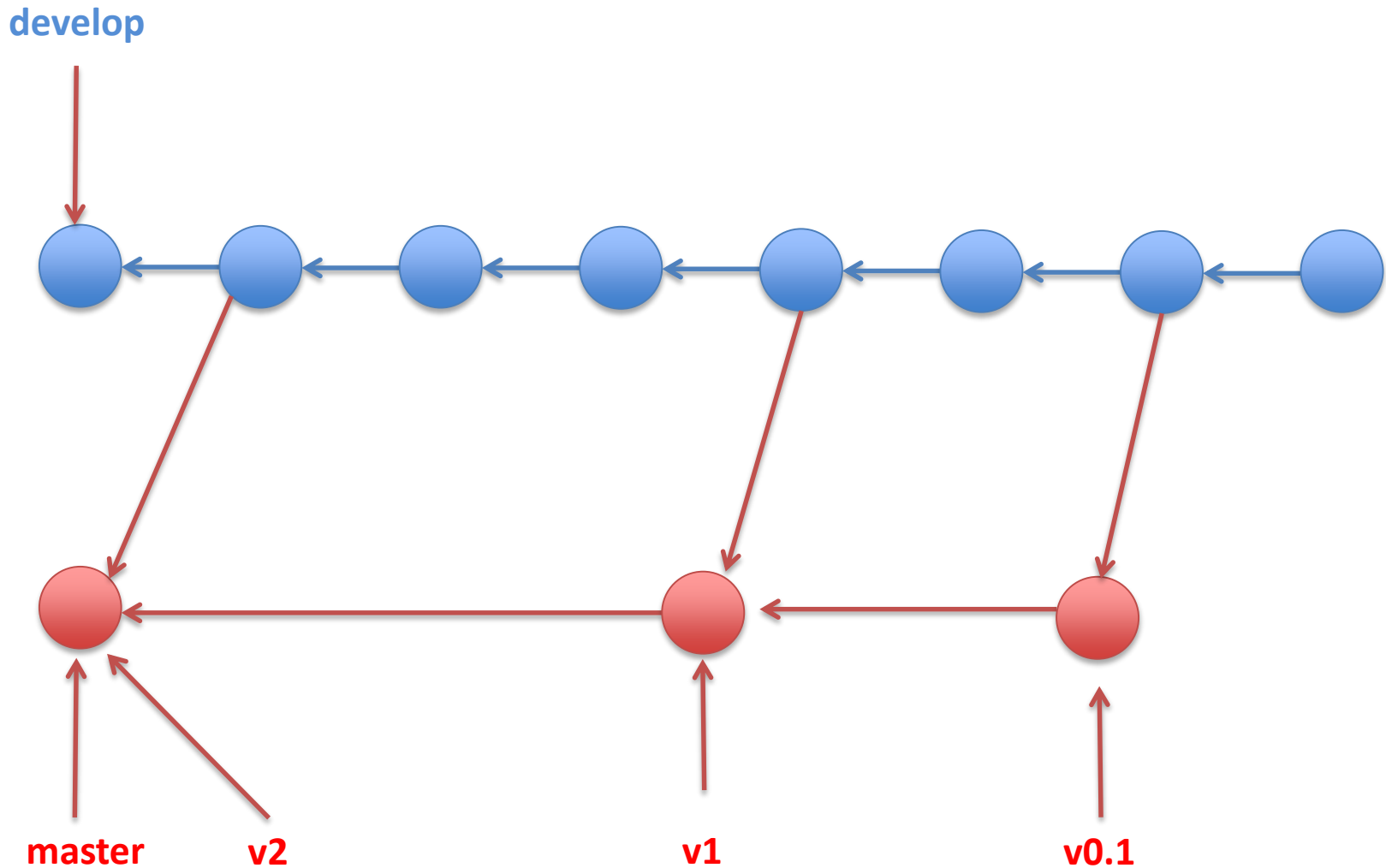
master/develop workflow

- Το master branch περιέχει την πιο πρόσφατη “stable” έκδοση
- Απευθύνεται στους **τελικούς χρήστες** του software
- Κάθε φορά που θέλουμε να γίνει release κάνουμε merge από το develop στο master
- Είναι χρήσιμο ώστε οι τελικοί χρήστες να μην τρέχουν παλιές εκδόσεις

Release workflow

```
git checkout develop
git pull upstream develop
git push origin develop
git checkout master
git pull upstream master
git merge develop
git tag -a v2.0
git push origin master
git push origin v2.0
git push upstream master
git push upstream v2.0
```

master/develop workflow





Ευχαριστούμε!