

Intro to Git and Github

John Duff



Sunday, May 10, 2009

Software developer working for Overlay.tv, a local Ottawa Company, I've been there about three years now.

Going to talk to you a little bit about Git and Github...who here knows what Git is? ...Github?

Git

- SCM (Source Control Management)
- Fully Distributed
- Open Source
- Linux Kernel, Rails

Sunday, May 10, 2009

SCM - think SVN, CVS, Perforce, whatnot

Distributed - all branches are everywhere, your checkout, mine, the servers - means its fast

Open Source - so ya, we love it

Github

- Hosted Git Repositories
- A "social network" for programmers
- Free (for public Open Source Projects)

Sunday, May 10, 2009

Hosted - think Sourceforge, google code

social network - more than just a repository, lets you easily build a community around your projects

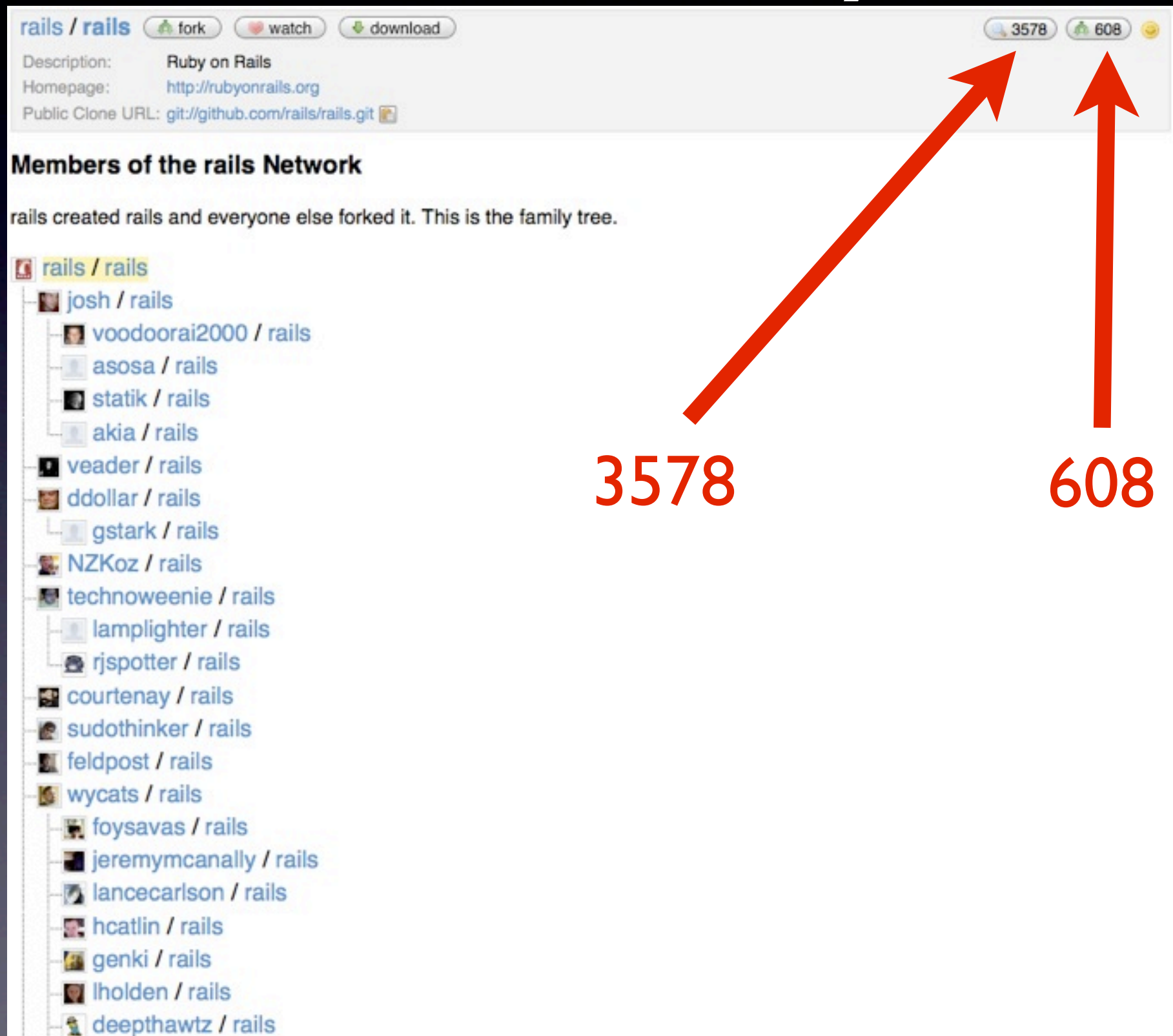
free - so ya, we love it

The rest of the talk is sort of split into 2 sections, the first being why you want to use git/github, the second half showing you the basics of getting started with them...so, on to why...

Why Github kicks Sourceforge Ass



Community



Sunday, May 10, 2009


-huge number of really smart people

Community

**why**
[unfollow](#) [message](#)

Name: **why the lucky stiff**
Email: why@whytheluckystiff.net
Followers: **970**
Public Repos: **25**
Member Since: **Feb 27, 2008**

8cpj
the octocat's pajamas
Updated April 23, 2009


all commits commits by owner 52 week participation


blimlimb
blimLimb, the travelling IRC bot troupe
Updated May 16, 2008



all commits commits by owner 52 week participation

bloopsaphone
arcade sounds and chiptunes for ruby and c
Updated March 23, 2009


all commits commits by owner 52 week participation

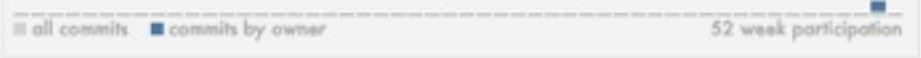
camping
the 4k pocket full-of-gags web microframework
Updated about 3 hours ago


all commits commits by owner 52 week participation


**drnic**
[unfollow](#) [message](#)

Name: **Dr Nic Williams**
Email: drnicwilliams@gmail.com
Website/Blog: <http://drnicwilliams.com>
Company: **Mocra** <http://mocra.com>
Location: **Brisbane, Australia**
Followers: **443**
Public Repos: **86**
Member Since: **Jan 30, 2008**

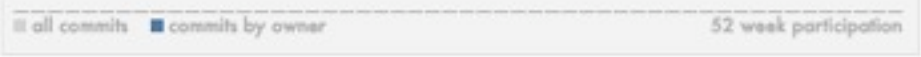
8cpj
the octocat's pajamas
Forked from [why/8cpj](#) April 23, 2009


all commits commits by owner 52 week participation

blinksale-greasemonkey
Greasemonkey/GreaseKit extensions for Blinksale
Updated October 11, 2008


all commits commits by owner 52 week participation


choctop
Build and deploy tools for Cocoa apps using Sparkle for distributions and upgrades; it's like Hoe...
Updated March 01, 2009


all commits commits by owner 52 week participation

Sunday, May 10, 2009

REAL people
okay, so they still use nicknames, but i know who these people are and can easily see what they've done

Community

delayed_job / Commit History 

2009-03-12

refactor job object



dmag (author)

March 12, 2009

commit 266fc15c12953a94a6d052281ce10059ddd5ebc1

tree bd3ee1b4866271d91d7eb2a2a95521d6021e6b2d

parent ad27c3e36fd951abb67545e7b711e5ed9521ec79

expand readme to include more hints



dmag (author)

March 12, 2009

commit ad27c3e36fd951abb67545e7b711e5ed9521ec79

tree f252a4cd0d24bc54ad02f9f07a853cef112e9dbd

parent dbc5e37c4d313cb2b85701999adff490df0c272b

2009-02-20

Added Module#handle_asynchronously that allows you to mark methods for async processing (Michael Koziarski)



tobi (author)

February 20, 2009

commit dbc5e37c4d313cb2b85701999adff490df0c272b

tree e2d2e71b7b83d53503cce98f8ee7e6e81bbb9a59

parent cf4ad9f6235a9dfeef7ff4f1ae92525945464350

2009-01-03

args.second changed to args[1] (pre-2.2 compatible)



olegshaldybin (author)


January 03, 2009

commit cf4ad9f6235a9dfeef7ff4f1ae92525945464350

tree 1948e081b52a35106d9742a11219ef3cbd679af2

parent 9935082b7e6b4eb900ee6830042d0a80543f3a7d

2008-12-30

Use sort_by{rand} instead of sort{rand} for randomization as sort{rand} is of consistent outcome. Thanks Alex Ostleitner. 



tobi (author)

December 30, 2008

commit 9935082b7e6b4eb900ee6830042d0a80543f3a7d

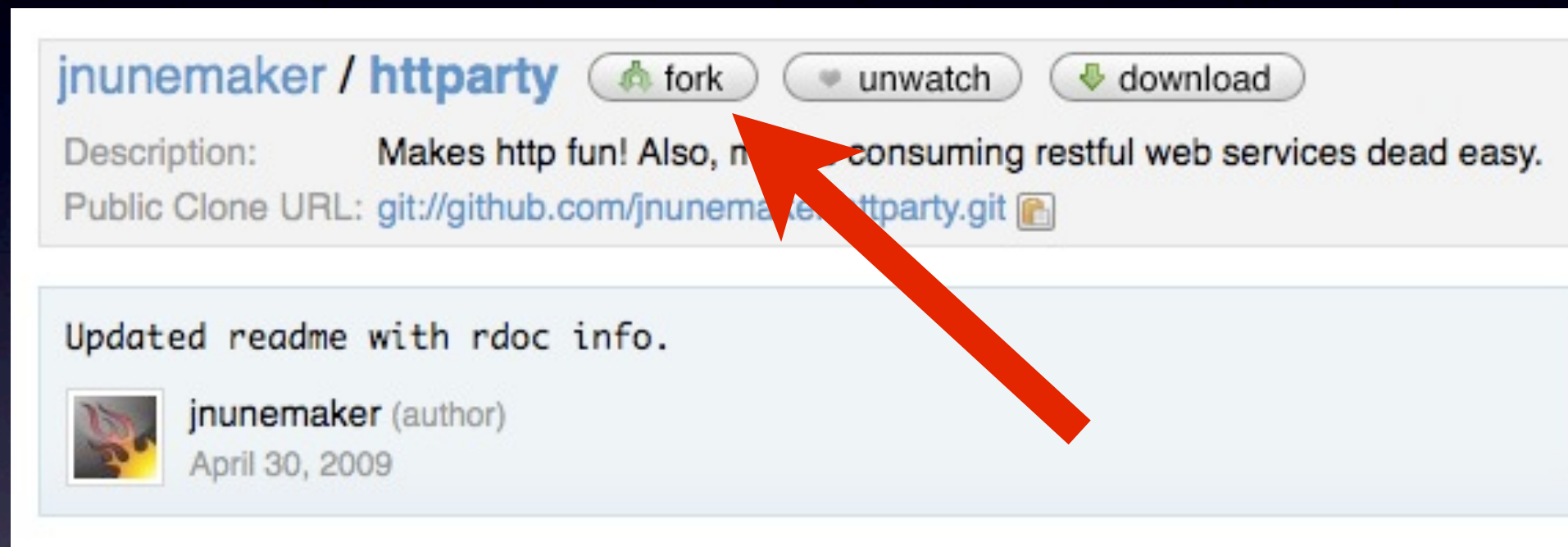
tree e4b8281d2a511aceb9e72d7bb0014f081b6482e2

parent a99302bc23f112b95120d0e85b21fa3d3c8ba699


Sunday, May 10, 2009

I can see what has been committed, by whom

Low Barriers to Entry



Low Barriers to Entry



The screenshot shows the GitHub interface for the repository 'jduff / httparty'. At the top, there are buttons for 'edit', 'pull request', 'unwatch', and 'download'. Below these, it says 'Fork of jnunemaker/httparty'. The description is 'Makes http fun! Also, makes consuming restful web services dead easy.' with an 'edit' link. The homepage is 'Click to edit' with an 'edit' link. The public clone URL is 'git://github.com/jduff/httparty.git' and the 'Your Clone URL' is 'git@github.com:jduff/httparty.git'. A large red arrow points from the bottom right towards the 'Your Clone URL'.

jduff / httparty edit pull request unwatch download

Fork of jnunemaker/httparty

Description: Makes http fun! Also, makes consuming restful web services dead easy. [edit](#)

Homepage: Click to edit [edit](#)

Public Clone URL: <git://github.com/jduff/httparty.git>

Your Clone URL: <git@github.com:jduff/httparty.git>

Sunday, May 10, 2009

now i have my very own copy of the code that i can clone and play around with

Low Barriers to Entry

Terminal — bash — 128x33

```
john$ git clone git@github.com:jduff/httparty.git
```

git clone

Low Barriers to Entry

```
:workspace_rails john$ cd httparty/  
:httparty john$ ls -alp
```

```
 612  6 May 21:19 ./
 612  6 May 21:19 ../
 408  6 May 21:19 .git/
   36  6 May 21:19 .gitignore
4834  6 May 21:19 History
1057  6 May 21:19 MIT-LICENSE
1235  6 May 21:19 Manifest
1078  6 May 21:19 README
1378  6 May 21:19 Rakefile
  102  6 May 21:19 bin/
   18  6 May 21:19 cucumber.yml
 306  6 May 21:19 examples/
 272  6 May 21:19 features/
3147  6 May 21:19 httparty.gemspec
  136  6 May 21:19 lib/
36162 6 May 21:19 setup.rb
  306  6 May 21:19 spec/
  136  6 May 21:19 website/
:~httparty john$
```

Sunday, May 10, 2009

and boom, i've got my own copy of the code checked out that i can do whatever i want with.

AND ITS UNDER VERSION CONTROL. You can't do this with a sorceforge or google code project as far as i know. sure, you can stumble around and find out how to check out the code, but you can't do a whole lot with it, it's not version controlled for you (anonymous checkout) and it's a pain in the ass to submit and manage patches.

Contributions

lawrencepit wants you to pull from lawrencepit/pending at 7e943130500b7947ee421b9dab27b4c0855daf08.



lawrencepit said 2 months ago:



Added test that shows that failure of a mock expectation isn't suppressed by a pending block. Not sure how to fix that.

Also added nicer description of pending cases at end of run (instead of seeing `_bind_1213214342342342` stuff).

Sunday, May 10, 2009

Github also makes it super easy to integrate and manage contributions. They can send you 'pull requests' and you can take a look at the code they committed and pull it into your fork of the project if you'd like

Contributions

Your **chronic** Fork Queue [?](#)

Will likely apply cleanly Will likely not apply cleanly

This data is out of date. We are showing you a cached version. Please refresh in a minute for more up to date data.

jdl / chronic

Select: All, None Actions

<input type="checkbox"/>	7fbbf0	jdl	Fixed a bug with DST and RepeaterDayName.	March 12, 2009
<input type="checkbox"/>	0bd0f4	jdl	Bumped the gemspec version number.	March 12, 2009

abhay / chronic

Select: All, None Actions

<input type="checkbox"/>	021d0c	abhay	moved time and object to a core extension folde...	February 18, 2008
<input type="checkbox"/>	a2419f	abhay	specs against core extensions	February 18, 2008
<input type="checkbox"/>	821438	abhay	added .gitignore	February 18, 2008
<input type="checkbox"/>	97f0bc	abhay	added helper line to all specs	February 18, 2008
<input type="checkbox"/>	b20b10	abhay	numerizer spec	February 18, 2008
<input type="checkbox"/>	908195	abhay	chronic spec	February 18, 2008
<input type="checkbox"/>	afc1da	abhay	Revert "chronic spec"	February 18, 2008
<input type="checkbox"/>	f2f967	abhay	added chronic spec	February 18, 2008
<input type="checkbox"/>	a2485c	abhay	added a small check to prevent console spam	February 18, 2008

evaryont / chronic

Select: All, None Actions

<input type="checkbox"/>	e409e0	evaryont	Added Chronic.strip_tokens	April 21, 2009
<input type="checkbox"/>	f51c99	evaryont	Made the regular expressions case insensitive	April 24, 2009

Sunday, May 10, 2009

The Fork Queue lets you see all the commits from various forks of the project, and lets you cherry-pick and apply them directly from the web interface.

More Github

- Issue Tracking
- API
- Pages
- Web based code viewer

Git

- index is kept in the root directory
- Small memory footprint
- working offline
- Pain free context switching

Sunday, May 10, 2009

git - no .svn dirs everywhere

small footprint

working offline

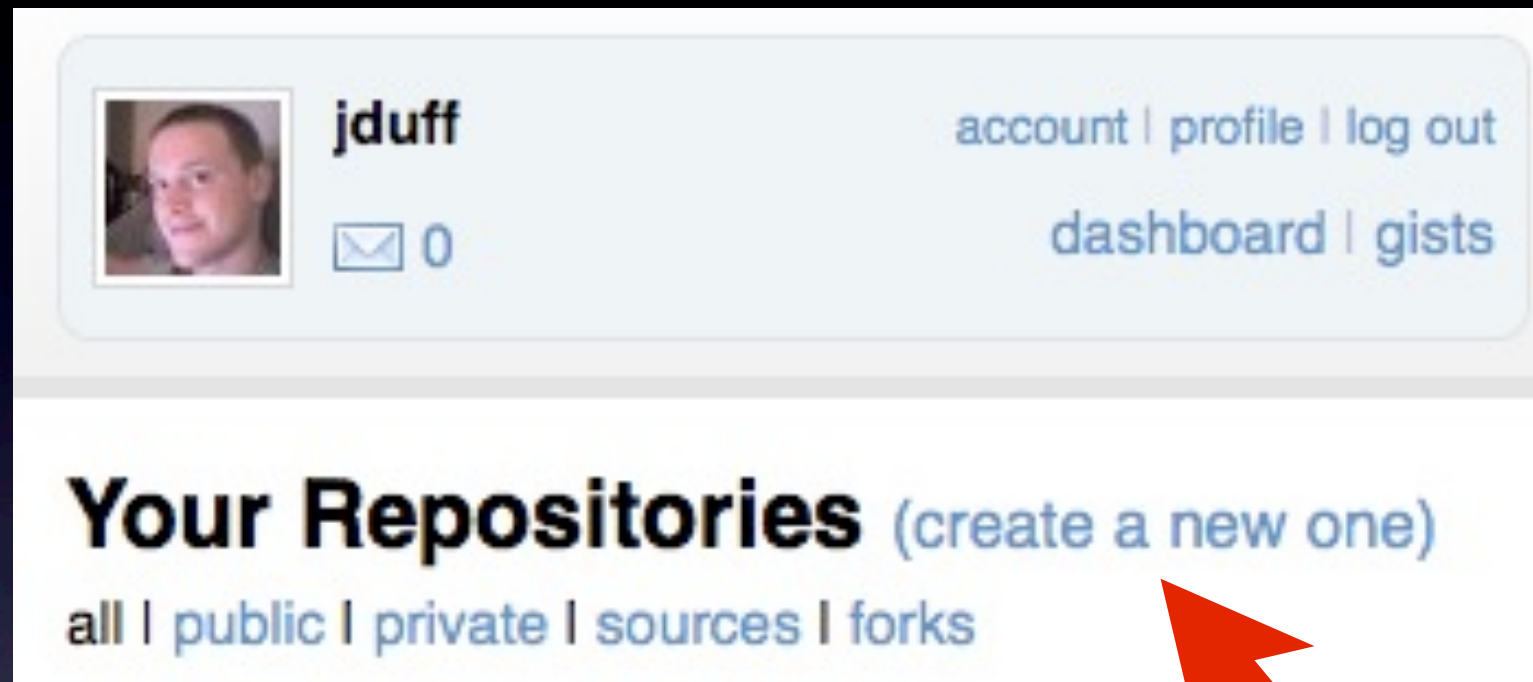
pain free context switching (clone prod, create issue branch, switch back to prod for hotfix, go back to issue and keep working)

Using Git

Sunday, May 10, 2009

Step by step walkthrough of setting up a project on github and how to use it.

Creating a Project



Creating a Project

Create a New Repository

Create a new empty repository into which you can push your local git repo.

NOTE: If you intend to push a copy of a repository that is already hosted on GitHub, then you should [fork](#) it instead.

Project Name

Description

Homepage URL

Who has access to this repository? (You can change this later)



Anyone ([learn more about public repos](#))



[Upgrade your plan to create more private repositories!](#)

Create Repository

Sunday, May 10, 2009

fill in the details, you can make private repositories with paid accounts

Creating a Project

Global setup:

```
Download and install Git  
git config --global user.name "John Duff"  
git config --global user.email duff.john@gmail.com
```

Next steps:

```
mkdir git-presentation  
cd git-presentation  
git init  
touch README  
git add README  
git commit -m 'first commit'  
git remote add origin git@github.com:jduff/git-presentation.git  
git push origin master
```

Existing Git Repo?

```
cd existing_git_repo  
git remote add origin git@github.com:jduff/git-presentation.git  
git push origin master
```

Importing a SVN Repo?

[Click here](#)

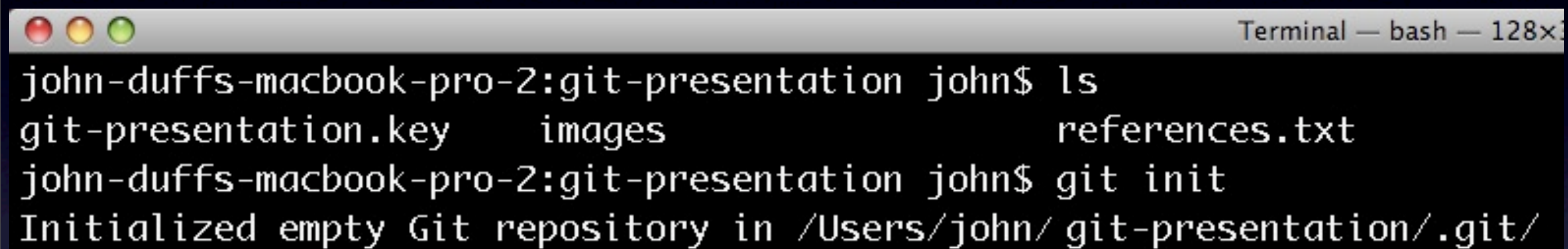
When you're done:

[Continue](#)

Sunday, May 10, 2009

once the project is created github tells you what to do next, if you've never used git or github start with installing git etc. were going to move on to creating the local repository and pushing it up to github

Pushing to Github

A screenshot of a macOS Terminal window. The title bar shows three colored window control buttons (red, yellow, green) on the left and the text "Terminal — bash — 128x32" on the right. The terminal content shows a user at a prompt "john-duffs-macbook-pro-2:git-presentation john\$". They run the command "ls", and the output is "git-presentation.key", "images", and "references.txt". Then they run "git init", and the output is "Initialized empty Git repository in /Users/john/git-presentation/.git/".

```
john-duffs-macbook-pro-2:git-presentation john$ ls
git-presentation.key  images  references.txt
john-duffs-macbook-pro-2:git-presentation john$ git init
Initialized empty Git repository in /Users/john/git-presentation/.git/
```

git init

Sunday, May 10, 2009

I already have the dir with the presentation in it going, just call git init and it will create the .git dir IN THE ROOT with all the info git needs (no .git dirs all over the place like svn)

Pushing to Github

```
git-presentation john$ git add .
git-presentation john$ git status
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file:   .DS_Store
#       new file:   git-presentation.key
#       new file:   images/.DS_Store
#       new file:   images/71px-Git-logo.svg.png
#       new file:   images/chronic_fork_queue.png
```

git add

Sunday, May 10, 2009

now we stage all the files to commit with git add. this lets you pick and choose what will go into the commit.

Pushing to Github

```
git-presentation john$ git commit -m 'first commit'
[master (root-commit)]: created 423f11e: "first commit"
18 files changed, 19 insertions(+), 0 deletions(-)
create mode 100644 git-presentation.key
create mode 100644 images/71px-Git-logo.svg.png
create mode 100644 images/chronic_fork_queue.png
create mode 100644 images/commit_history.png
create mode 100644 images/drnic_github_profile.png
create mode 100644 images/github-logo.png
create mode 100644 images/httpart_clone.png
create mode 100644 images/httparty.png
create mode 100644 references.txt
git-presentation john$
```

git commit

Sunday, May 10, 2009

and now we can commit the changes. it's import to not that everything we've done so far is local, we haven't hit the network at all, even when committing these files.

Pushing to Github

```
git-presentation john$ git remote add origin git@github.com:jduff/git-presentation.git
git-presentation john$ git push origin master
Counting objects: 21, done.
Compressing objects: 100% (21/21), done.
Writing objects: 100% (21/21), 1.01 MiB, done.
Total 21 (delta 10), reused 0 (delta 0)
To git@github.com:jduff/git-presentation.git
 * [new branch]      master -> master
john-duffs-macbook-pro-2:git-presentation john$
```

git remote add origin
git push origin master

Sunday, May 10, 2009

next we can let our local repository know about the remote on github and then push it up to github. You might be wondering what these 'origin' and 'master' things are. origin is basically just a label for the remote repository, you can add any number of remotes with different labels. master is the label of our main branch.

Viewing it on Github

jduff / git-presentation [edit](#) [pull request](#) [unwatch](#) [download](#) 1 0

Description: My Git/Github presentation for Summercamp2009 [edit](#)

Homepage: Click to edit [edit](#)

Public Clone URL: [git://github.com/jduff/git-presentation.git](https://github.com/jduff/git-presentation.git)

Your Clone URL: [git@github.com:jduff/git-presentation.git](https://github.com/jduff/git-presentation.git)

first commit

jduff (author)
about a minute ago

commit [423f11e86368f1676c83e82e366024815d597a9c](#)

tree [729a75917a641a9e4477b438dff1803b31c8240f](#)

git-presentation /

name	age	message	history
.DS_Store	about a minute ago	first commit [jduff]	
git-presentation.key	about a minute ago	first commit [jduff]	
images/	about a minute ago	first commit [jduff]	
references.txt	about a minute ago	first commit [jduff]	

Sunday, May 10, 2009

now if we go back and refresh the project page on github we can see our first commit and the files we put there.

Viewing it on Github

first commit



jduff (author)

about a minute ago

commit 423f11e86368f1676c83e82e366024815d597a9c

tree 729a75917a641a9e4477b438dff1803b31c8240f

git-presentation / references.txt

100644

19 lines (10 sloc)

0.371 kb

[edit](#)

[raw](#)

[blame](#)

[history](#)

```
1  rockem_sockem.jpg
2  http://farm3.static.flickr.com/2075/1697408176_6b7f192ba1.jpg?v=0
3
4  71px-Git-logo.svg.png
5  http://en.wikipedia.org/wiki/Git_(software)
6
7  github-logo.png
8  http://gihub.com
9
10
11
12 http://github.com/schacon/git-presentations/tree/master
13
14 http://www.readwriteweb.com/archives/github_a_social_network_for_programmers.php
15
16
17 Resources
18
19 http://git.or.cz/course/svn.html
```

Sunday, May 10, 2009

we can even take a look at the contents of the files, run blame commands or look at the history of the file, all from the web interface.

Working with Others

The git-presentation network graph

All branches in the network using **jduff/git-presentation** as the reference point.

[Show Help](#)



Sunday, May 10, 2009

Next lets see what it's like to work with someone else on a project. My friend Josh noticed my presentation here on github and was thinking about doing one so he forked mine and made a couple changes. On the network graph we can see all the forks visually to get an idea of what's going on with the repository. this is also useful when a project gets popular and some branches might be moving at different speeds, with some more features, you can choose which one you'd like to use.

Working with Others


Your **git-presentation** Fork Queue

Will likely apply cleanly

Will likely **not** apply cleanly

jreynolds / git-presentation

Select: [All](#), [None](#)

Actions 



7afab2



jreynolds

Added new reference

about 11 hours ago



6d49ee



jreynolds






Added 3 new resources

about 11 hours ago

Sunday, May 10, 2009


Now we saw that josh made some changes, so we can go over to the 'Fork Queue' and take a look at what he's done, in case I might want it for my presentation.

Working with Others


jreynolds / git-presentation  watch  download  1  1 

Fork of jduff/git-presentation

Description: My Git/Github presentation for Summercamp2009

Public Clone URL: [git://github.com/jreynolds/git-presentation.git](https://github.com/jreynolds/git-presentation.git) 

Added new reference

 **jreynolds** (author)
about 11 hours ago

commit


7afab2a2bd0aa6a75d78687a839c9e80b79fded6

tree

56615c0d70aaf2463c5e8909fc38174faee42d3a

parent

62b751ec8e187e95a255d44bb56be2a9ed54b258

 **references.txt**


references.txt

...	...	@@ -13,6 +13,7 @@ http://github.com/schacon/git-presentations/tree/master
13	13	
14	14	http://www.readwriteweb.com/archives/github_a_social_network_for_programmers.php
15	15	
	16	+ http://git.or.cz/gitwiki/GitSvnComparsion
16	17	
17	18	Resources
18	19	

Sunday, May 10, 2009


taking a peak at one of his commits we see he's added another reference link. We'd like that as well so lets apply his change.



Working with Others

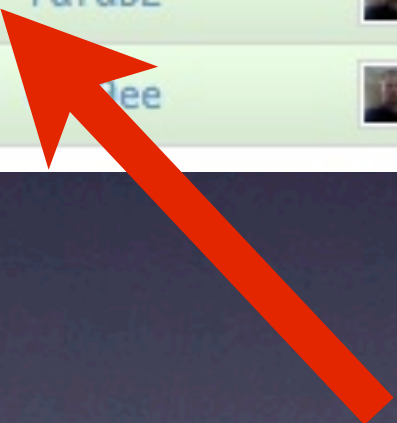
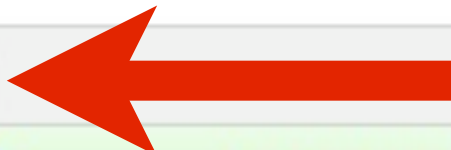
Your **git-presentation** Fork Queue 

Will likely apply cleanly Will likely **not** apply cleanly

jreynolds / git-presentation

Select: All, None Actions 

<input type="checkbox"/>	7afab2	 jreynolds	Added new reference	about 11 hours ago
<input type="checkbox"/>	lee	 jreynolds	Added 3 new resources	about 11 hours ago



Sunday, May 10, 2009

just select the change and then apply from the actions menu to apply the change

Working with Others

```
john-duffs-macbook-pro-2:git-presentation john$ git pull origin master
From git@github.com:jduff/git-presentation
 * branch                master      -> FETCH_HEAD
Updating 784b210..44215da
Fast forward
 references.txt |      1 +
 1 files changed, 1 insertions(+), 0 deletions(-)
john-duffs-macbook-pro-2:git-presentation john$ █
```

git pull

Sunday, May 10, 2009

now back on my machine i can pull in the changes that were made from the web interface

Working with Others

```
* references.txt
1 rockem_sockem.jpg
2 http://farm3.static.flickr.com/2075/1697408176\_6b7f192ba1.jpg?v=0
3
4 71px-Git-logo.svg.png
5 http://en.wikipedia.org/wiki/Git\_\(software\)
6
7 github-logo.png
8 http://github.com
9
10
11
12 http://github.com/schacon/git-presentations/tree/master
13
14 http://www.readwriteweb.com/archives/github\_a\_social\_network\_for\_programmers.php
15
16 http://git.or.cz/gitwiki/GitSvnComparsion
17
18 Resources
19
20 http://git.or.cz/course/svn.html
```

Sunday, May 10, 2009

and we can see the new reference josh added in my local file

Working with Others

- `git remote add jreynolds git://github.com/jreynolds/git-presentation.git`
- `git checkout jreynolds/master`
- `git pull jreynolds master`

Sunday, May 10, 2009

`git remote add ...` lets you add another persons fork so that you can use it

`git checkout ...` will checkout jreynolds master branch, this is useful if you want to take a look at what's in another persons branch without merging it with your own

`git pull jreynolds` will pull the other branch into your own and merge them

Resources

Git Cheat Sheet

<http://git.or.cz/>

Remember: `git command --help`

Global Git configuration is stored in `$HOME/.gitconfig` (`git config --help`)

Create

From existing data

```
cd ~/projects/myproject
git init
git add .
```

From existing repo

```
git clone ~/existing/repo ~/new/repo
git clone git://host.org/project.git
git clone ssh://you@host.org/proj.git
```

Show

Files changed in working directory

```
git status
```

Changes to tracked files

```
git diff
```

What changed between \$ID1 and \$ID2

```
git diff $id1 $id2
```

History of changes

```
git log
```

History of changes for file with diffs

```
git log -p $file $dir/ec/tory/
```

Who changed what and when in a file

```
git blame $file
```

A commit identified by \$ID

```
git show $id
```

A specific file from a specific \$ID

```
git show $id:$file
```

All local branches

```
git branch
```

(star "*" marks the current branch)

Concepts

Git Basics

master : default development branch
origin : default upstream repository
HEAD : current branch
HEAD~ : parent of HEAD
HEAD~4 : the great-great-grandparent of HEAD

Revert

Return to the last committed state

```
git reset --hard
```

⚠ you cannot undo a hard reset

Revert the last commit

```
git revert HEAD
```

Creates a new commit

Revert specific commit

```
git revert $id
```

Creates a new commit

Fix the last commit

```
git commit -a --amend
```

(after editing the broken files)

Checkout the \$id version of a file

```
git checkout $id $file
```

Branch

Switch to the \$id branch

```
git checkout $id
```

Merge branch1 into branch2

```
git checkout $branch2
git merge branch1
```

Create branch named \$branch based on the HEAD

```
git branch $branch
```

Create branch \$new_branch based on branch \$other and switch to it

```
git checkout -b $new_branch $other
```

Delete branch \$branch

```
git branch -d $branch
```

Commands Sequence

the curves indicate that the command on the right is usually executed after the command on the left. This gives an idea of the flow of commands someone usually does with Git.

Update

Fetch latest changes from origin

```
git fetch
```

(but this does not merge them)

Pull latest changes from origin

```
git pull
```

(does a fetch followed by a merge)

Apply a patch that some sent you

```
git am -3 patch.mbox
```

(in case of a conflict, resolve and use `git am --resolved`)

Publish

Commit all your local changes

```
git commit -a
```

Prepare a patch for other developers

```
git format-patch origin
```

Push changes to origin

```
git push
```

Mark a version / milestone

```
git tag v1.0
```

Useful Commands

Finding regressions

```
git bisect start
```

(to start)

```
git bisect good $id
```

(\$id is the last working version)

```
git bisect bad $id
```

(\$id is a broken version)

```
git bisect bad/good
```

(to mark it as bad or good)

```
git bisect visualize
```

(to launch gitk and mark it)

```
git bisect reset
```

(once you're done)

Check for errors and cleanup repository

```
git fsck
git gc --prune
```

Search working directory for foo()

```
git grep "foo()"
```

Resolve Merge Conflicts

To view the merge conflicts

```
git diff
```

(complete conflict diff)

```
git diff --base $file
```

(against base file)

```
git diff --ours $file
```

(against your changes)

```
git diff --theirs $file
```

(against other changes)

To discard conflicting patch

```
git reset --hard
git rebase --skip
```

After resolving conflicts, merge with

```
git add $conflicting_file
```

(do for all resolved files)

```
git rebase --continue
```

Git Cheat Sheet

<http://www.cheat-sheets.org/saved-copy/git-cheat-sheet-large.png>

Sunday, May 10, 2009

So, that's it for me. this presentation didn't go into all the details of what you can do with git and Github, but hopefully it's enough for you to get started

Resources

- <http://github.com/schacon/git-presentations/tree/master>
- http://www.readwriteweb.com/archives/github_a_social_network_for_programmers.php
- <http://git.or.cz/gitwiki/GitSvnComparision>
- <http://git.or.cz/course/svn.html>

Resources (win)

- <http://github.com/guides/using-git-and-github-for-the-windows-for-newbies>
- <http://kylecordes.com/2008/03/22/git-windows-works/>
- <http://www.gitcasts.com/posts/git-on-windows>

Image References

- rockem_sockem.jpg - http://farm3.static.flickr.com/2075/1697408176_6b7f192ba1.jpg?v=0
- 71px-Git-logo.svg.png - [http://en.wikipedia.org/wiki/Git_\(software\)](http://en.wikipedia.org/wiki/Git_(software))
- github-logo.png - <http://github.com>

Contact

- github.com/jduff
- twitter.com/johnduff
- duff.john@gmail.com
- github.com/jduff/git-presentation