Intermediate' GIT

@looselytyped

Recap

Commits

Commit

References a tree
0..n "parent" commits
Author info and timestamp

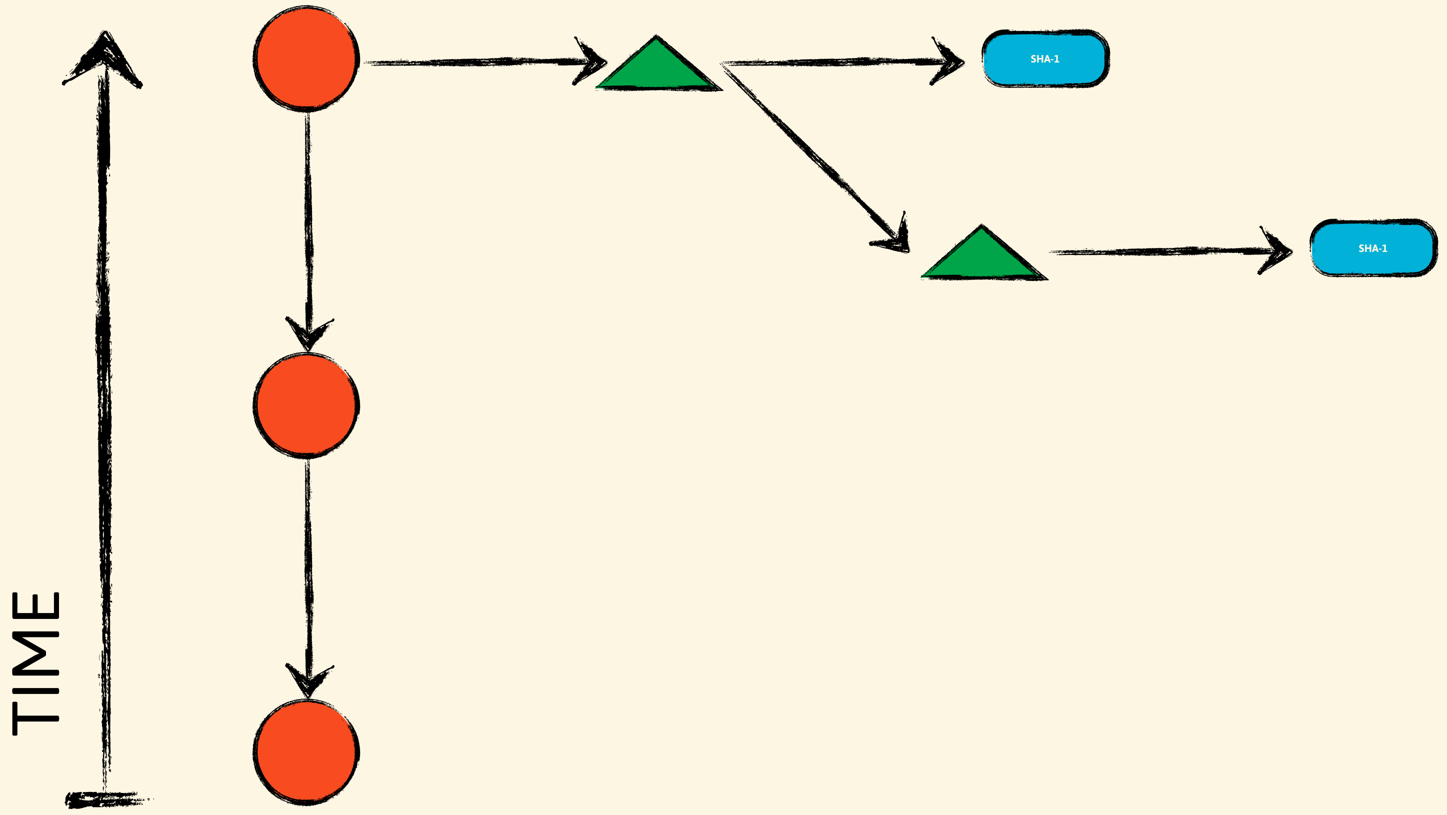
Commit message body

Commit

```
tree 19bb38db5027a6c0f1e2dd38e364dcab388ba236
parent 5d672786b5581e5f2816784a64a61950eddbf676
author Raju Gandhi <raju.gandhi+github@gmail.com> 1369509976 -0400
committer Raju Gandhi <raju.gandhi+github@gmail.com> 1369509976 -0400

Added lib folder with dependency
```

Commit



Reflog

Know where you have been

Reflog

List of branch tips

Updated every time HEAD moves

Reflog

Your Safety Net

Reflog

```
(master) > git reflog
30dc0d8 HEAD@{87}: checkout: moving from someBranch to master
ab48e0e HEAD@{88}: commit: S1
30dc0d8 HEAD@{89}: checkout: moving from master to someBranch
80f7604 HEAD@{90}: commit (initial): chore: empty initial commit
```

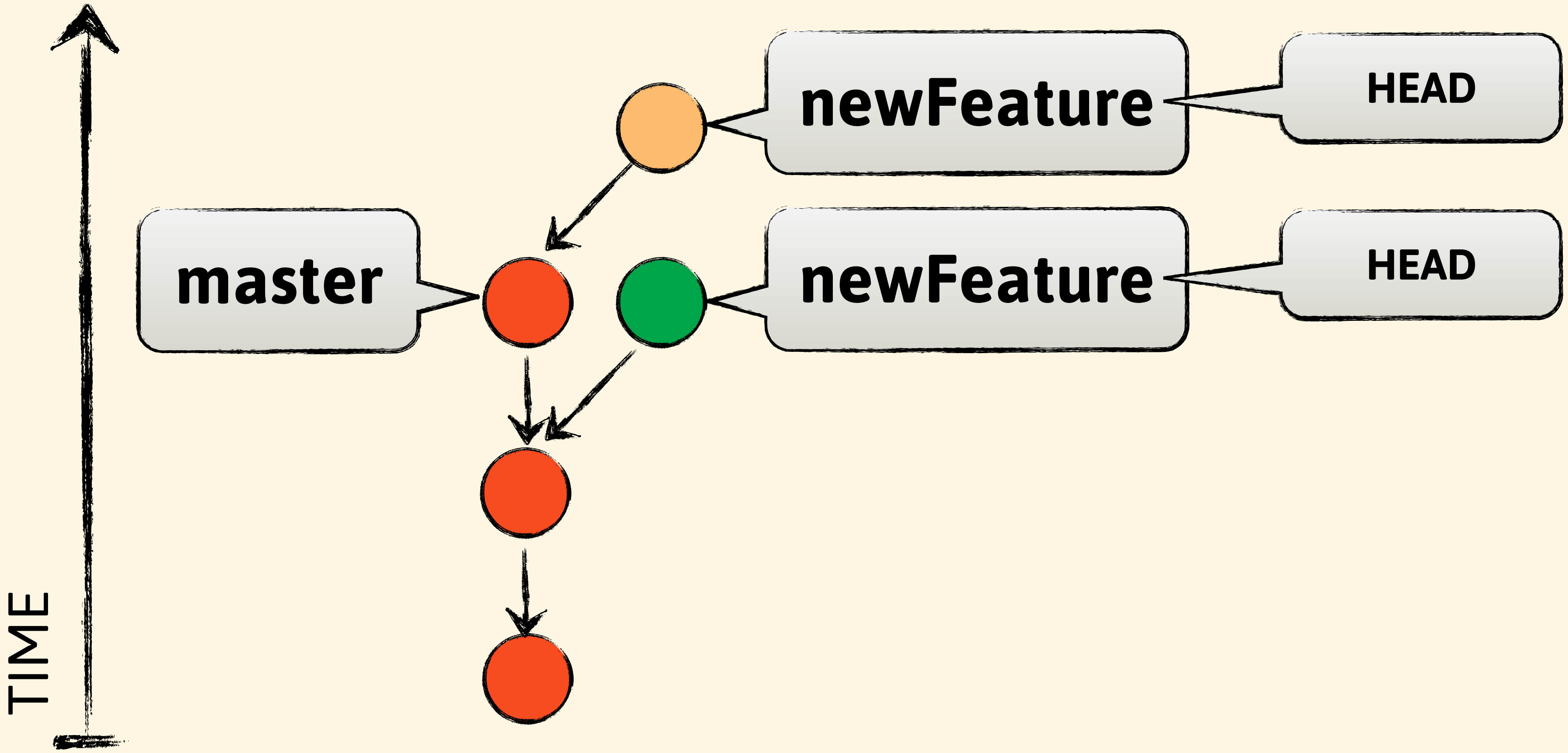
KEEP CALM USE
AND REFLOG

http://www.keepcalm-o-matic.co.uk/p/keep-calm-and-use-git-reflog/

Rebase

Rewriting history

Rebase



```
(newFeature) > git rebase master
```

Rebase interactively

Allows you to rewrite commits

Rebase interactively

```
(someBranch) > git rebase –i <someCommit>
```

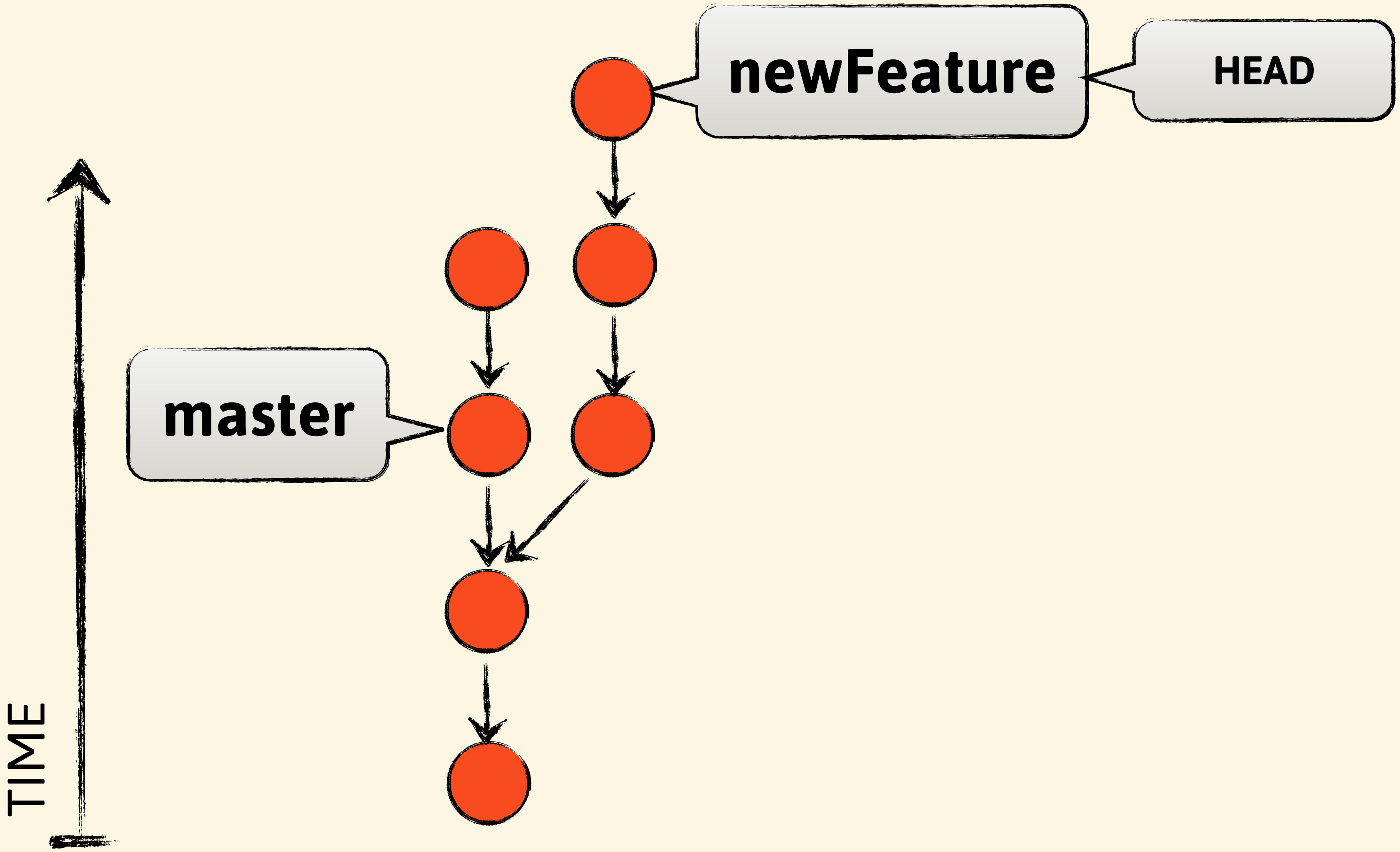
Rebase interactively

```
(someBranch) > git rebase -i <someCommit>
```

Rebase interactively

```
# Commands:
#  p, pick = use commit
#  r, reword = use commit, but edit the commit message
#  e, edit = use commit, but stop for amending
#  s, squash = use commit, but meld into previous commit
#  f, fixup = like "squash", but discard this commit's log message
#  x, exec = run command (the rest of the line) using shell
```

Rebase interactively (Tip)



```
(newFeature) > git rebase -i HEAD~3
```

Stash

Pseudo-Commits

```
# hack hack hack
```

```
# Oh No! Production Bug
```

```
git status
```

```
# On branch getCommits
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#	new file:   src/com/looselytyped/galore/bla.clj
#
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#	modified:   .gitignore
#	modified:   src/com/looselytyped/galore/commit.clj
#	modified:   src/com/looselytyped/galore/fs.clj
#
```

Gah!!!

KEEP CALM USE

AND STASH

http://www.keepcalm-o-matic.co.uk/p/keep-calm-and-use-git-reflog/

```
git status
```

```
# On branch getCommits
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#	new file:   src/com/looselytyped/galore/bla.clj
#
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#	modified:   .gitignore
#	modified:   src/com/looselytyped/galore/commit.clj
#	modified:   src/com/looselytyped/galore/fs.clj
#
```

```
git stash
```

```
git status
```

```
# On branch master
nothing to commit, working directory clean
```

Yay!!!

Stash

Stores the index and working tree state

Resets to the current commit!

Stash

Is just like a commit

Does not participate in the log

Stash

```
git stash

git stash save <message>
```

Stash

```
git stash list

git stash pop

git stash apply
```

Stash (Tip)

```
# at regular intervals

git stash && git stash apply

# :)
```

Workflow

Tips & Tricks

Initial Empty Commit

Tips & Tricks - Empty Commit

```
() > git init
Initialized empty Git repository in ~/newRepo/.git/
(master) >
```

Tips & Tricks - Empty Commit

```
(master) > git log
fatal: bad default revision 'HEAD'
```

Tips & Tricks - Empty Commit

```
(master) $ git branch someBranch
fatal: Not a valid object name: 'master'.
```

Instead …

Tips & Tricks - Empty Commit

```
() > git init
Initialized empty Git repository in ~/newRepo/.git/
(master) >
```

Tips & Tricks - Empty Commit

```
(master) $ git commit \
               --allow-empty \
               -m "chore: empty initial commit"
[master (root-commit) 1c6a739] chore: empty initial
commit
```

Tips & Tricks - Empty Commit

```
(master) $ git commit \
               --allow-empty \
               -m "chore: empty initial commit"
[master (root-commit) 1c6a739] chore: empty initial
commit
```

Tips & Tricks - Empty Commit

```
(master) $ git log
commit 1c6a739441caf4f52f3b3cd13faee6bd2766afb0
Author: Raju Gandhi <raju.gandhi+github@gmail.com>
Date:   Fri Jan 24 15:18:29 2014 -0500

    chore: empty initial commit
```

Tips & Tricks - Empty Commit

```
(master) $ git branch someBranch
(master) $ git branch
* master
  someBranch
```

Alternatively

Tips & Tricks - Empty Commit

```
git config --global \
       alias.start \
         '!git init \
          && git commit \
               --allow-empty \
               -m "chore: empty initial commit"'
```

Git Hooks

Tips & Tricks - Git Hooks

```
.git/hooks/
├── applypatch-msg.sample
├── commit-msg.sample
├── post-update.sample
├── pre-applypatch.sample
├── pre-commit.sample
├── pre-push.sample
├── pre-rebase.sample
├── prepare-commit-msg.sample
└── update.sample
```

Tips & Tricks - Git Hooks

```
.git/hooks/
├── applypatch-msg.sample
├── commit-msg.sample
├── post-update.sample
├── pre-applypatch.sample
├── pre-commit.sample
├── pre-push.sample
├── pre-rebase.sample
├── prepare-commit-msg.sample
└── update.sample
```

Tips & Tricks - Git (Pre) Hooks

- The sample supplied with your Git repo is a good start
  - Checks for trailing white-space

Tips & Tricks - Git (Pre) Hooks

- Use these for
  - Spell check in the documentation
  - Ensure that code compiles
  - Lint-ing
  - Run tests (???)
  - …

Tips & Tricks - Git Hooks

```
.git/hooks/
├── applypatch-msg.sample
├── commit-msg.sample
├── post-update.sample
├── pre-applypatch.sample
├── pre-commit.sample
├── pre-push.sample
├── pre-rebase.sample
├── prepare-commit-msg.sample
└── update.sample
```

Tips & Tricks - Git (Commit) Hooks

- Use these for
  - Spell checking commit message
  - Ensure that commit follows a certain format
    - See Angular.JS' hook for inspiration

Git Log

SMALL COMMITS

FREQUENT COMMITS

Tips & Tricks - Git Log

- Each commit should be atomic
  - Commit fast, commit often
  - Use `rebase` (interactive) to fix commits prior to "publishing"

Tips & Tricks - Git Log

- Be consistent in commit messages
  - Use a `git-commit-hook`

A **Commit Message** has

Tips & Tricks - Git Log

Short title

Tips & Tricks - Git Log

Short title **< 50c**

Tips & Tricks - Git Log

Short title **< 50c**

Longer description

Tips & Tricks - Git Log

Short title	**< 50c**
Longer description	wrap at 72 chars

Tips & Tricks - Git Log

```
<type>: <subject>
_BLANK LINE_
<body>
_BLANK LINE_
<footer>
```

Tips & Tricks - Git Log

```
<type>: <subject>

#feat|fix|docs|style|refactor|test|chore|revert
```

Tips & Tricks - Git Log

```
<type>: <subject>

feat: commit header should have a valid 'type'
```

Tips & Tricks - Git Log

```
<body>

This is a much more detailed summary of the commit. Notice how there
is a new line at the end of 72 characters so that we don't overflow
the line.

Paras are delimited by blank lines.
```

Tips & Tricks - Git Log

```
<footer>

JRA-090 #close
Change-Id: Ic8aaa0728a43936cd4c6e1ed590e01ba8f0fbf5b
```

Tips & Tricks - Git Log

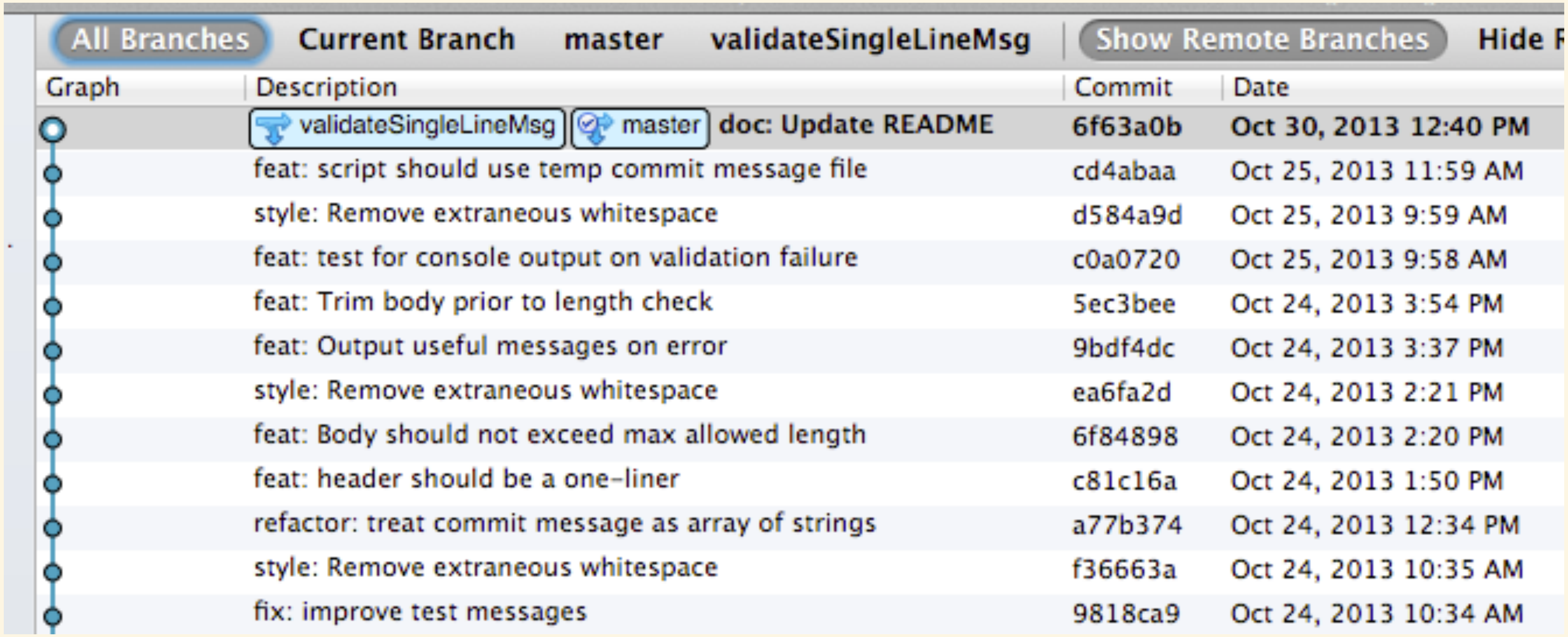
```
doc: update README to include examples

This is a much more detailed summary of the commit. Notice how there
is a new line at the end of 72 characters so that we don't overflow
the line.

Paras are delimited by blank lines.

JRA-090 #close
Change-Id: Ic8aaa0728a43936cd4c6e1ed590e01ba8f0fbf5b
```

Tips & Tricks - Git Log



Graph	Description	Commit	Date
	validateSingleLineMsg master **doc: Update README**	**6f63a0b**	**Oct 30, 2013 12:40 PM**
	feat: script should use temp commit message file	cd4abaa	Oct 25, 2013 11:59 AM
	style: Remove extraneous whitespace	d584a9d	Oct 25, 2013 9:59 AM
	feat: test for console output on validation failure	c0a0720	Oct 25, 2013 9:58 AM
	feat: Trim body prior to length check	5ec3bee	Oct 24, 2013 3:54 PM
	feat: Output useful messages on error	9bdf4dc	Oct 24, 2013 3:37 PM
	style: Remove extraneous whitespace	ea6fa2d	Oct 24, 2013 2:21 PM
	feat: Body should not exceed max allowed length	6f84898	Oct 24, 2013 2:20 PM
	feat: header should be a one-liner	c81c16a	Oct 24, 2013 1:50 PM
	refactor: treat commit message as array of strings	a77b374	Oct 24, 2013 12:34 PM
	style: Remove extraneous whitespace	f36663a	Oct 24, 2013 10:35 AM
	fix: improve test messages	9818ca9	Oct 24, 2013 10:34 AM

THANKS

Credits

- Asap font
- Theme blatantly stolen from Git's homepage

Resources

- Stashes
- A Note About Git Commit Messages