Git hooks

About me

Rajesh Kumar

DevOps Architect

@RajeshKumarIN | www.RajeshKumar.xyz

Why ?

Problem: Some guys forgot to log their working time. Therefore, for teamleaders, they can not know how this ticket going.

Solution: After you git commit, open browser to log your working time.

How ?

Git has one function named as hooks. (http://git-scm.com/book)

In my words, Hooks like event trigger

After you did git operation, it will trigger some events.

Examples

Git checkout ---------> you want to check whether all gems have been installed.

Git push ---------> you want to trigger jenkins server to run rspec to check you code.

Place

- Git hooks is in .git/hooks

- You can use shell script or ruby or python

- But remember to use chmod +x filename

Types

There has two types:
 Client side (In your local PC)
 Server side (In remote server like Stash)

- applypatch-msg
- post-update
- pre-rebase
- commit-msg
- pre-applypatch
- update
- post-commit
- pre-commit
- post-receive

Git Commit

When you use git commit :

pre-commit

↓

prepare-commit-msg

↓

commit-msg

↓

post-commit

No Ticket, No Commit

In the future, when you look back to the code,
you want to know why this guy changed this code ?

Such as: "Fixed rspec error"

"Refactored the code"

---- >   "**GI-12345**: Fixed rspec error"

"**GI-54321**: Refactored the code"

Pre-commit -1

Before commit, this hook can check something for you.

 For example, in my pc, I will check where the code has "binding.pry" or not.

Place: .git/hooks/pre-commit

Code:

```ruby
#!/usr/bin/env ruby

require 'colorize'

# check for rails debugger
rails_result = `grep -rlsn "binding.pry" --exclude-dir=".git" .`
if rails_result != ""
  puts "[Error] you put binding.pry in your files\n".red
  puts rails_result
```

Pre-commit -2

```
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:    app/controllers/checkout_controller.rb
#
rakuten@malldev:~/mount/gep-ui-mall$ git commit -m "add binding.pry"
[Error] you put binding.pry in your files
./app/controllers/checkout_controller.rb
rakuten@malldev:~/mount/gep-ui-mall$ ▇
```

Prepare-commit-msg -1

- Means giving commit-msg a template

  For example:
  Now you are working at pr-GI-12345-2.5-id

  Every time, when you commit, you want to add ticket number to your commit message.

  commit message here
    ↓
- **Ticket #GI-12345** commit message here.

Prepare-commit-msg -2

```ruby
#!/usr/bin/env ruby


TICKET_REGEXP = /[A-Z]+-\d+/ # WEB-12345


TICKET_REF_FORMAT = 'Ticket #%s # ' # JIRA Ticket reference syntax.


branch_name = `git branch --no-color 2> /dev/null`[/^\* (.+)/, 1]
msg_filename = ARGV[0]


if ticket_in_branch = branch_name.match(TICKET_REGEXP)
  ticket_ref = TICKET_REF_FORMAT % ticket_in_branch.to_s
  old_msg = File.read msg_filename


  unless old_msg.match ticket_ref
    File.open msg_filename, 'w' do  | new_msg |
      new_msg.write ticket_ref + old_msg
    end
  end
end
```

Prepare-commit-msg -3


```
hebinbin@ubuntu:~/project/shop$ git br
   master
 * pr-GI-12345-2.5-id
hebinbin@ubuntu:~/project/shop$
```

```
# On branch pr-GI-12345-2.5-id
# 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:   app/controllers/application_controller.rb
#
no changes added to commit (use "git add" and/or "git commit -a")
hebinbin@ubuntu:~/project/shop$ git add app/
hebinbin@ubuntu:~/project/shop$ git st
# On branch pr-GI-12345-2.5-id
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   app/controllers/application_controller.rb
#
hebinbin@ubuntu:~/project/shop$ git commit -m "fixed rspec error"
```

```
commit 97d7f91a5c005742da00d253bbfcf5a74cb1d366
Author: hebinbin <hebinbin0905@gmail.com>
Date:   Thu May 9 20:40:53 2013 +0900

    Ticket #GI-12345 # fixed rspec error
```

Commit-msg -1

- We can this hook to check whether your commit message has ticket number or not.

- #!/usr/bin/env ruby
-

- require 'colorize'
-

- TICKET_REGEXP = /[A-Z]+-\d+/ # WEB-12345
-

- commit_message = File.open ARGV[0], 'r' { |f| f.read }
-

- unless commit_message.match(TICKET_REGEXP)
-   puts "[ERROR] commit-message-format: JIRA ticket number is not existed.".red
-   exit(1)
- end

Commit-msg -2

Post-commit -1

If everything goes well, then we can open firefox to log your time.

```ruby
#!/usr/bin/env ruby

require 'colorize'

TICKET_REGEXP = /[A-Z]+-\d+/ # WEB-12345

LINK = "https://*****.atlassian.net/browse/"

puts "[Warning] log time: Please do not forget to log your time".red

commit_message = %x{git log -1 --abbrev-commit}.split("\n").last.strip

jira_ticket = commit_message.match(TICKET_REGEXP)
puts "please visit #{LINK}#{jira_ticket.to_s} to log time".green

exec("firefox #{LINK}#{jira_ticket.to_s}")
```

Post-commit -2

Server side (git push)

- pre-receive
- update
- post-receive

Thanks

https://github.com/hebinbin/git-hooks-jira