17 tháng 7 2018

Home »pull request »Git: Pull Request là gì ? Vì sao cần Pull Request ?

Git: Pull Request là gì ? Vì sao cần Pull Request ?

Lúc mới dùng git, ví dụ github, bạn hay thấy có cái nút Pull Request. Lúc đó bạn không hiểu tính năng của nó để làm gì. Thật ức chế. Hãy cùng mình thông não cái này.

Định nghĩa

Pull Request (viết tắt là PR) là chức năng cho phép bạn nói với người khác về các thay đổi bạn đã đẩy lên kho Github (Github repository) của người sở hữu code đó (chủ repository). Một khi Pull Request được gửi, người nào quan tâm có thể Review (xem xét) lại các thay đổi, hoặc thảo luận các sửa đổi tiềm năng, và nếu PR đó được chấp thuận có thể tiếp theo đó đẩy tiếp các commit của họ lên kho code nếu cần thiết.

PR thường được sử dụng trong các team hoặc tổ chức mà các thành viên làm việc hợp tác sử dụng mô hình kho chia sẻ (như github), ở đó mỗi người chia sẻ các kho riêng lẻ của họ và các nhánh chủ đề (topic branch) để phát triển các tính năng và để tách biệt các thay đổi khỏi nhánh chính (master, kho chính thức mà code ổn định nhất). Nhiều dự án trên Github sử dụng pull request để quản lý các thay đổi từ các người đóng góp, việc áp dụng pull request giúp cho người đóng góp thông báo cho người bảo trì dự án về các thay đổi khi họ tạo Pull Request và bắt đầu việc code review đồng thời thảo luận về các thay đổi trước khi chúng được merge (trộn) vào nhánh chính.

Vậy thì làm sao để tạo Pull Request.

Có 2 work flow chính khi làm việc với pull request:

Một là Pull Request từ forked repository (Một bản sao của code gốc)

Hai là Pull Request từ nhánh bên trong repository.

Ở đây mình sẽ tập chung vào cách 2.

Đầu tiên là phải tạo Topical Branch (một nhánh chính)

Đầu tiên ta sẽ phải tạo một nhánh từ commit (thay đổi) mới nhất trên master. Để đảm bảo repository (kho code) được cập nhật mới nhất trước tiên.

```
git pull origin master
```

`git pull` sẽ thực hiện `git fetch` tiếp theo là `git merge` để cập nhật local repo (bản trên máy tính) giống như là remote repo (bản code trên server).

Để tạo branch sử dụng `git checkout -b [] `, ở đó `[base-branch-name]` là tuỳ chọn và mặc định là `master` . Ta sẽ thử tạo branch (nhánh) mới có tên là `pull-request-demo` từ branch `master` và đẩy nó lên github.

```
git checkout -b pull-request-demo
git push origin pull-request-demo
```

Tạo Pull Request

Để tạo pull request, bạn phải thay đổi committed (các xác nhận đã sửa đổi) tới new branch (nhánh mới).
Hãy vào mục **repository page** trên github. Và click vào nút “Pull Request” trong repo header (tiêu đề repo).

Chọn branch bạn muốn merged sử dụng “Head branch” dropdown. Bạn nên để trường còn lại như vậy, trừ khi bạn làm việc từ remote branch. Trong trường hợp đó, chỉ cần chắc chắn rằng base repo và base branch được đặt đúng.

Nhập các tiêu đề và mô tả cho việc pull request.

Cuối cùng click vào “Send pull request” để hoàn tất quá trình tạo pull request. Cuối cùng bạn có thể thấy open pull request.

Sử dụng Pull Request

Có thể viết comment liên quan tới pull request
Xem các commit trong pull request.
Hoặc là xem tất cả các file đã thay đổi từ pull request giữa các commit trong phần “File Changed”.
Bạn thậm chí còn có thể để lại các comment ở dòng cụ thể trong code change bằng cách rê chuột vào bên trái của dòng và click vào biểu tượng màu xanh nước biển.

Trộn Pull Request

Một khi bạn và các người hợp tác đã ok với các thay đổi, bạn cần phải trộn nó trở lại master. Có vài cách để thực hiện điều này.

Đầu tiên bạn có thể dùng nút “Merge pull request” bên dưới pull request để trộn các thay đổi. Điều này chỉ có thể thực hiện khi không code merge conflict với base branch. Nếu tất cả OK, bạn chỉ cần thêm commit message và click vào Confirm Merge” để merge các thay đổi.

Merging Locally

Nếu pull request không thể được trộn online do merge conflict, hoặc bận muốn kiểm tra lại các thứ locally trước khi gửi merge trở lại repo trên github, bạn có thể thực hiện merger locally thay thế.

Bạn có thể tìm thấy hướng dẫn bằng cách bấm vào nút `(i)` trên merge bar. Tuy nhiên cách thay thế này có thể tốt hơn cho các branch dài.

Squash, Rebase, and Cherry Pick

Trong các long standing branch, việc trộn có thể thường gây ra nhiều vấn đề khi cập nhật nó nếu thay đổi ở branch cho trước conflict với thay đổi gần đây được merged vào master branch. Nếu có nhiều commit vào cùng file, `git merge` có thể ép buộc bạn fix cùng merge conflict lặp đi lặp lại, gây ra rất nhiều sự đau đầu. Trong khi đó có nhiều cách để giảm nhẹ vấn đề này, như là bật thể độ `git rerere` để tái sử dụng các recorded resolution của conflict merge, squashing một loạt các thay đổi liên quan vào 1 commit và cherry-picking nó vào master là một giải pháp tuyệt vời, đặc biệt là đối với topic branch và các tính năng được cô lập hoá (isolated).

Có một vài ưu điểm khi thực hiện trộn theo cách này. Đầu tiên bạn chỉ phải xử lý với merge conflict một lần, khi tất cả commit được nén thành 1. Thứ hai, từng commit diễn tả toàn bộ các thay đổi yêu cầu cho tính năng hoặc công việc, điều đó khiến cho nó dễ dàng để pin point bug và các vấn đề khác khi chúng này sinh và để xoá bỏ các change set khi nó không còn cần thiết.

Cũng có nhược điểm khi dùng squashing commits. Đầu tiên, bạn sẽ mất các chi tiết và thông tin về từng thay đổi, như khi tất cả thay đổi được squashed (nén, ép) là được nén cùng nhau. Vì thế các ảnh hưởng là như nhau. Thứ hai, nó còn thể nguy hiểm và khó giải quyết (problematic) nếu được sử dụng sai như là squashing các commit mà đã được push lên remove server và các cái khác phụ thuộc vào công việc của họ. Bởi vì squashing là thay đổi git history, bạn có thể gây nhiều conflict theo cách này. Tuy nhiên, nếu bạn sử dụng nó locally hoặc bạn chỉ là 1 người làm việc trên branch của mình, và bạn biết chính xác những gì bạn đang làm.

Để thực hiện điều đó bạn sử dụng

```
git rebase -i HEAD~10
```

`i` có nghĩa là bạn ở chế độ tương tác và `HEAD~10` có nghĩa là kiểm tra 10 commit mới nhất.
Nếu bạn thấy lỗi `fatal: Needed a single revision` , nó thường có nghĩa rằng không còn nhiều commit. Hãy thử với con số thấp hơn.

Nó sẽ mở ra một editor với các commit message.
Có nhiều tuỳ chọn có sẵn ở giai đoạn này, có thể đọc các help của github. Ở đây, tôi chỉ đơn giản squash tất cả thay đổi trong pull request thành 1. Lưu và đóng editor lại.

Màn hình tiếp theo sẽ pop up và hỏi bạn sửa lại commit message. Bạn có thể chọn để edit chúng hoặc đơn giản là tiếp tục. Lưu và đóng editor lại.

Một khi bạn squash thành công, bạn có thể push nó lên remote repo. Trong trường hợp này, các squashed commit đã được đẩy lên server. Tuy nhiên, tôi chỉ là một user của branch này, và có thể force push một cách an toàn commit để update git repo.

```
git push origin pull-request-demo -f
```

Để merged the commit, chúng ta sẽ sử dụng `git cherry-pick`

Bạn đã xong, Github sẽ phát hiện thay đổi và update pull request. Bạn có thể đánh dấu pull request là đã được merged và có thể tuỳ chọn xoá bỏ branch đi.

Closing Pull Request

Bạn có thể đơn giản click “Close” button trên pull request để close nó. Một cách tuỳ chọn, bạn có thể xoá các branch một cách trực tiếp hoặc sử dụng nút “Delete this branch”

Không có nhận xét nào:

Đăng nhận xét