

applypatch-msg

“git am” komutu ile ilişkilendirilir. commit mesajı dosyasını düzenleyebilir ve genellikle bir yapılan eklemelerin iletisini bir projenin standart standartlarına göre doğrulamak veya etkin biçimde biçimlendirmek için kullanılır. Sıfır olmayan bir çıkış durumu commiti iptal eder.

1 parametre alır: Önerilen commit mesajını içeren dosyanın adı

pre-applypatch

“git am” komutu ile ilişkilendirilmiştir. Sistemde eksik görülen bir kısma yapılan eklemeler uygulandıktan sonra, değişiklikler yapılmadan önce çağrılır. Sıfır olmayan bir durumdan çıkmak değişiklikleri kabul edilmeyen bir durumda bırakacaktır. Değişiklikleri yapmadan önce ağacın durumunu kontrol etmek için kullanılabilir.

Parametre almaz.

post-applypatch

“git am” komutu ile ilişkilendirilmiştir. Sistemde eksik görülen bir kısma yapılan eklemeler uygulandıktan ve commit edildikten sonra çalıştırılır. Commit edildikten sonra çalıştırıldığı için işlemi iptal edemez ve çoğunlukla bildirimler oluşturmak için kullanılır.

Parametre almaz.

pre-commit

“git commit” komutu ile ilişkilendirilmiştir. Bu hook, önerilen commit mesajını almadan önce çağrılır. Mesaj yerine commitin kendisini kontrol etmek için kullanılır.

Parametre almaz.

prepare-commit-msg

“git commit” komutu ile ilişkilendirilmiştir. Sadece geçerli mesaj editörünü başlatmadan önce, varsayılan commit mesajını aldıktan sonra çağrılır. Sıfır olmayan bir çıkış commiti iptal eder. Bu, mesajı engellenemeyecek şekilde düzenlemek için kullanılır.

1 ila 3 arasında parametre alır: 1 mesajın bulunduğu geçici alan, 2 commit in türü (git commit ten sonra -m, -t girildiyse yada merge işlemi yapıldıysa bu bilgi olur) 3- SHA1 git commit -c olarak çağırıldıysa commitin özet değeri

commit-msg

“git commit” komutu ile ilişkilendirilmiştir. Bir standarda uygunluğu sağlamak veya herhangi bir kritere göre reddetmek için, düzenlendikten sonra mesajı ayarlamak için kullanılabilir. Sıfır olmayan bir değerle çıkarsa commiti iptal edebilir.

1 parametre alır: Önerilen mesajı tutan dosya.

post-commit

“git commit” komutu ile ilişkilendirilmiştir. Asıl commit yapıldıktan sonra çağrılır. Bu nedenle commiti bozamaz. Özellikle bildirimlere izin vermek için kullanılır.

Parametre almaz.

pre-rebase

“git rebase” komutu ile ilişkilendirilmiştir. Bir dal(branch) rebase edilirken çağrılır. Temel olarak rebase durdurmak için kullanılır.

2 parametre alır: İlk parametre, dizinin çatallandığı ters yönlendirir. İkinci parametre, yeniden dallanmakta olan daldır ve geçerli dalı yeniden birleştirirken ayarlanmaz.

post-checkout

“git checkout” ve “git clone” komutları ile ilişkilendirilmiştir. Genel olarak koşulları doğrulamak, farklılıkları göstermek ve gerekirse ortamı yapılandırmak için kullanılır.

3 parametre alır: Önceki HEAD referansı, yeni HEAD referansı, branch checkout (1) veya file checkout (0) olduğunu gösteren bayrak

post-merge

“git merge” ya da “git pull” komutları ile ilişkilendirilmiştir. Merge’den(birleştirmeden) sonra çağrılır. Bu nedenle merge iptal edilemez. Git’in işleyemediği izinleri veya diğer veri türlerini kaydetmek veya uygulamak için kullanılabilir.

1 parametre alır: yapılan merge(birleştirme) işleminin squash birleştirme olup olmadığını belirten bir durum bayrağı

pre-push

“git push” komutu ile ilişkilendirilmiştir. Bir itmenin gerçekleşmesini önlemek için kullanılabilir.

2 parametre alır: Uzak hedefin adını ve konumunu sağlayan iki parametre ile çağrılır.

pre-receive

Uzak repodaki “git-receive-pack” komutu ile ilişkilendirilir. Uzak depodaki referansları güncellemeye başlamadan hemen önce pre-receive hook başlatılır. Çıkış durumu, güncellenenin başarısını veya başarısızlığını belirler.

Parametre almaz.

update

Uzak repodaki “git-receive-pack” komutu ile ilişkilendirilir. Uzak depodaki ref güncellemeden hemen önce, update hook çağrılır. Çıkış durumu, ref güncellemesinin başarısını veya başarısızlığını belirler.

3 tane parametre alır. Güncellenen ref adı, eski nesne adı, yeni nesne adı

post-receive

Uzak repodaki “git-receive-pack” komutu ile ilişkilendirilir. Tüm referanslar güncellendikten sonra uzak depoda bir kez çalıştırılır. Güncellemelerden sonra çağrıldığından işlemi iptal edemez.

Parametre almaz, ancak stdin üzerinden "<old-value> <new-value> <ref-name>" şeklinde bilgi alır.

post-update

Uzak repodaki “git-receive-pack” komutu ile ilişkilendirilir. Tüm referanslar gönderildikten sonra yalnızca bir kez çalıştırılır. Bu bağlamda post-receive hook a benzer, ancak eski veya yeni değerleri almaz. Daha çok, itilen referanslar için bildirimleri uygulamak için kullanılır.

güncellenen ref adlarının her biri için değişken sayıda parametre alır.

pre-auto-gc

“git gc –auto” komutu ile ilişkilendirilir. Depoları otomatik olarak temizlemeden önce bazı kontrolleri yapmak için kullanılır.

Parametre almaz.

post-rewrite

“git commit --amend, git-rebase” komutları ile ilişkilendirilir. Bu, git komutları zaten işlenmiş verileri yeniden yazarken çağrılır. Parametrelere ek olarak, "<old-sha1> <new-sha1>" şeklinde stdin içindeki karakter dizileri alır.

1 parametre alır: İlişkilendirilen komutun adı (amend ya da rebase)