Introducción a los Repositorios de Código Distribuido

Git Hooks

Diego Madariaga

Contenidos de la clase

1. Git Hooks
2. Clasificación de Git Hooks
3. Consideraciones al usar Git Hooks
4. Ejemplos

1.
Git Hooks

Git Hooks

- Herramienta de Git, que permite la automatización de tareas

- Los Git Hooks se usan para ejecutar scripts antes o después de que ocurra algún evento en particular
  - Commit
  - Push
  - Merge
  - etc

Git Hooks

- Algunos eventos pueden componerse de múltiples pasos
- Se puede programar la ejecución de un script distinto para cada uno de dichos pasos
- Los scripts pueden escribirse en cualquier lenguaje de programación
  - Más comunes: Shell, Python, Perl y Ruby

Git Hooks

- Pertenecen y afectan a un repositorio específico
  - No son copiados durante un **git clone**

- Si se necesita compartir un Hook, se tiene que copiar por una vía alternativa

2.

Clasificación de Git Hooks

Clasificación de Git Hooks

Según el tipo de repositorio:

- ***Client-side* hooks**: Se ejecutan en repositorio local
  - Ej: al hacer un commit o al traer cambios desde un repositorio remoto

- ***Server-side* hooks**: Se ejecutan en un repositorio remoto
  - Ej: al hacer un push para confirmar cambios en el repositorio remoto
  - *Sitios como GitHub no admiten hooks

Clasificación de Git Hooks

Según cuándo se ejecutan:

- ***pre* hooks**
- ***post* hooks**

Pre hooks

- Se ejecutan antes que una acción o evento se lleve a cabo
- Se pueden utilizar para aprobar, modificar o rechazar un cambio antes que sea aplicado
- Si el código de retorno del script es distinto a 0, esto indica un error, por lo que la operación Git es abortada

Post hooks

- Se ejecutan después de que una acción o evento se ha completado
- Se pueden utilizar para ejecutar procesamiento adicional, o para enviar notificaciones (email)
- El código de retorno del script no tiene mucha utilidad, ya que la operación Git no puede ser cancelada en este punto

Creación de Git Hooks

- Se almacenan en el directorio .git/hooks de cada repositorio git
- Para activar un hook, es necesario crear un archivo ejecutable y colocarlo en la carpeta .git/hooks
- El nombre del archivo está predefinido por Git (existe un listado de posibles Git Hooks)

Nombres de Git Hooks

- **pre-commit** invocado por git-commit
- **pre-merge-commit** invocado por git-merge
- **post-commit** invocado por git-commit
- **post-merge** invocado por git-merge
- **pre-push** invocado por git-push
- …

Son invocados por distintas operaciones de Git. Algunos hooks pueden ser ignorados explícitamente:

- Ej: git commit --no-verify

3.
Consideraciones al usar Git Hooks

Implicancias al usar Git Hooks

- Se cambiará el flujo y el comportamiento normal de Git
- Las operaciones de Git se hacen más lentas
  - Ej: Ejecutar tests antes de cada commit
- Un script con errores, puede interferir grandemente en la productividad
- No es sencillo compartir hooks dentro de un grupo de desarrollo

¿Cuándo usar Git Hooks?

- Para contrarrestar la decisión tomada por una operación Git. Ej: **pre-commit** para aprobar/cancelar un commit.
- Para manipular los datos después de que un comando comience a ejecutarse. Ej: **commit-msg** para modificar el mensaje de commit.
- Para operar en el extremo remoto en donde sólo puedes acceder a través del protocolo Git. Ej: **post-update**.
- Para ejecutar una de varias operaciones posibles, dependiendo del resultado de un comando Git.

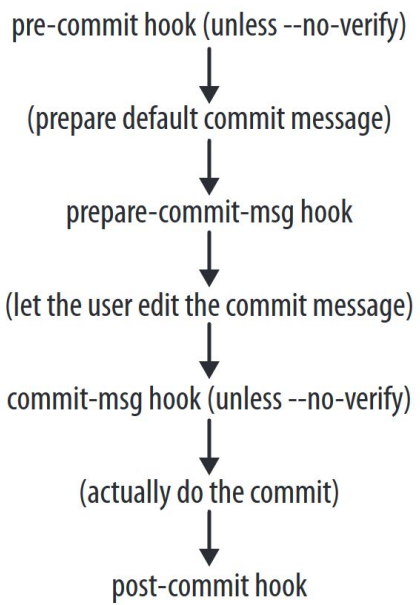
4.
Ejemplos

Ejemplos de uso

- Estandarizar mensajes de commit (contenido y largo)
- Ejecutar tests
- Generar documentación automáticamente
- Estilizar código

Hooks relacionados a un commit

- Invocados exclusivamente por git-commit



Permite abortar inmediatamente un commit si hay algo mal con su contenido. Se ejecuta antes de que el usuario ingrese el mensaje de commit.

Permite modificar el mensaje por defecto de Git antes de que se muestre al usuario. Puede usarse para cambiar el "mensaje de commit por defecto".

Permite validar o modificar el mensaje de commit ingresado por el usuario. Puede usarse para chequear contenido o largo del mensaje.

Se ejecuta una vez ha finalizado la operación de commit. Puede usarse para enviar un email, actualizar un archivo de log, etc.

Introducción a los Repositorios de Código Distribuido

Git Hooks

Diego Madariaga