

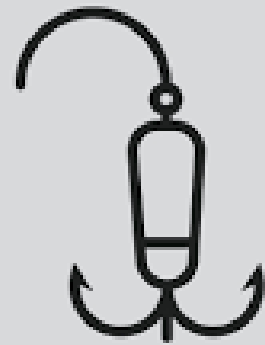
Lecture 11

Advanced Git: Hooks and Plumbing

**Sign in on the
attendance
sheet!**

Git Hooks

- “Hooks” give you control of what happens before/after certain things
 - Pre-push
 - Pre-commit
 - Post-commit
- Stored in `.git/hooks`



**Git
Hooks**

Scenario

- I have a collaborative website (like Wikipedia), and I'd like to receive an email every time someone commits a change
- How would I do this?
 - Post-receive hook notifies me every time someone pushes to the remote repository

Scenario (cont.)

Next I created the `post-receive` hook:

```
1  #!/bin/sh
2
3  # 1. Create symlinks to the real upgrade scripts for each branch, for example:
4  #   ./upgrade-beta.sh # will only be called when "beta" branch is updated
5  #   ./upgrade-prod.sh # will only be called when "prod" branch is updated
6  #
7  # 2. Put these symlinks in the repository root, NOT in the hooks/ directory
8
9  # the post-receive hook receives parameters on stdin
10 while read oldrev newrev refname
11 do
12     branch=$(git rev-parse --symbolic --abbrev-ref $refname)
13     upgrade_sh=./upgrade-$branch.sh
14     if test -e $upgrade_sh; then
15         upgrade_sh=$(readlink $upgrade_sh || echo $upgrade_sh)
16         echo calling upgrade script: $upgrade_sh
17         $upgrade_sh
18     else
19         echo NOT calling non-existent upgrade script: $upgrade_sh
20     fi
21 done
```

Scenario II

- Check that I don't have any "TODO" comments left in my code before I commit
- A pre-commit hook would be perfect for this
- For an example, "cat .git/hooks/pre-commit.sample" in any Git repository, and you should have at least one example. Feel free to "ls .git/hooks" and look at other ones too

Git Internals: Key-Value Store

- One major part of git is a hash-based key-value store
- Blobs, trees, and commits live in `.git/objects`
- `git hash-object -w file.txt`
- `git cat-file -p <hash_value>`

Git Internals: Updating the Staging Area

- `git update-index --add --cacheinfo 100644 <hash_value> file.txt`
- `git write-tree`

Git Internals: Committing

- `git commit-tree <tree_hash> -p <parent_hash> -m "Commit message"`

Git Internals: Updating Refs

- `git update-ref refs/heads/master <hash_value>`
- `git symbolic-ref HEAD refs/heads/my-branch`