



pre-commit hooks

This is a different take on hooks from [Pre Commit Hooks: Combating Laziness](#), written two years ago

There are times when it would be awesome if we could force ourselves (or our development team) to perform some actions before committing code to the project's repository.

Git has a set of tools called [hooks](#) that can help with this enforcement process.

The idea behind hooks is that they will run at set times during the request lifecycle and run the code you specify in the hook file. You can use hooks to run linting and accessibility checks before you commit code and reject the commit if any of the check fails.

In this example we want to run `gulp axe` and `gulp eslint` before each and every commit and fail the commit if there are errors returned from either command.

We'll leverage the `pre-commit` hook to accomplish this. This hook is run first, before you type a commit message. It's used to inspect the snapshot that's about to be committed, complete tasks before the commit happens. Exiting non-zero from this hook aborts the commit.

Move this code into the `hooks/pre-commit` file inside your `.git` directory.

```
#!/bin/sh

# Stash non-committed changes
git stash -q --keep-index
# if node_modules directory doesn't exist
# then run npm install
if [ ! -d "/node_modules/" ]
then
    echo "Directory node_modules not created"
    npm install
```

```
fi
"/node_modules/"# Run gulp axe to check accessibility
gulp axe
# Run gulp eslint to check for syntax
gulp eslint
# Assigns the exit status to a variable
RESULT=$?
# Pop the changes back to current directory
git stash pop -q
[ $RESULT -ne 0 ] && exit 1
# Otherwise exit with error code 0
exit 0
```

This example makes the following assumptions:

- You've added node_modules to .gitignore
- You've added axe and eslint as gulp tasks

The downside of hooks is that they are not copied when you clone a repository. If your intent with these scripts is to enforce a policy, you'll want to do that on the server side. The [Git Book](#) provides examples of server side scripts to enforce Git Policies.