

# SPIN 2020 / 2021 Exercise

## Prerequisites

Make sure you have a version of git installed and have a [GitHub](#) account.

## Setup

Fork the repository <https://github.com/rhaas80/spin2020> into a repository of your own and clone your forked repository to your laptop.

## Exercise

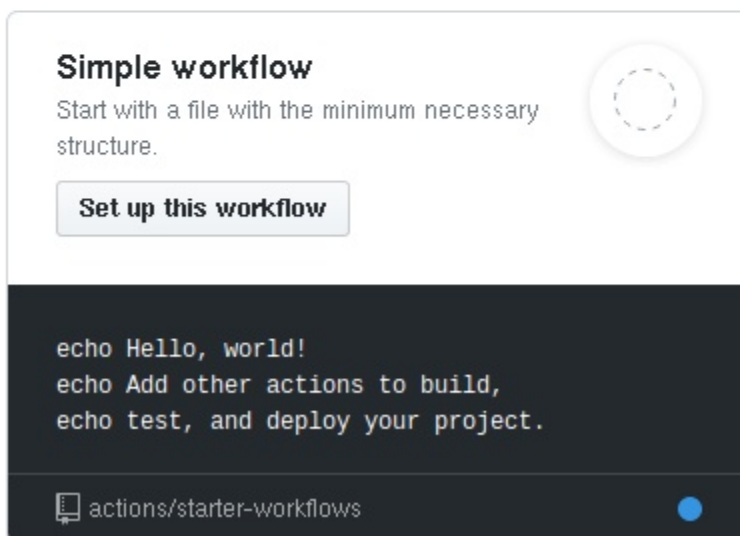
- Add a new "Simple workflow" using the Action tab of your repository page on GitHub (this would be <https://github.com/rhaas80/spin2020/actions/new> for the demo repo).



## Get started with GitHub Actions

Choose a workflow to build, test, and deploy your code. Manage branch management, and issue triaging work the way you want

### Build and test your repository



- Configure an action that outputs the commit hash of HEAD using the command `git rev-parse` and whose output it is in a form similar to

```
HEAD is bb8c42de11e171e22a7df0615981608faf35ea4d
```

- Once the action is in place on GitHub, add a line consisting of your name to README.md and commit it to trigger the workflow. Check that the workflow triggered and produced output on the Actions pane. You should see output similar to this:

- Finally, write and push to the repository a script using Perl or Python that counts the number of lines in README.md and creates a HTML document whose body reads "Number of lines in README.md is N" where "N" is the number of lines in README.md.

## Sending in results

Once done, please send me the URL of your repository on GitHub using my [rhaas@illinois.edu](mailto:rhaas@illinois.edu) email address. Please make sure that your repository:

- contains you workflow yaml file
- shows output of the workflow in the Actions pane
- contains your script to count the number of lines in README.md

## Current Jenkins setup

As part of the project we would like to implement as much of the functionality currently offered by Jenkins as feasible. You can have a look at a sample Jenkins report for a test run that resulted in some failures on: <https://build-test.barrywardell.net/job/EinsteinToolkit/1672> using the user name Albert and password Einstein .

Items to implement:

- identifying the updates to submodules that triggered each test run
- capture log output
- capture information which test failed and keep track of changes in the number of failing tests
- send email to Einstein Toolkit mailing list when tests start to fail and are no longer failing