# Github Actions for Process Automation

A lot of times we need to build a project every time we make changes. I picked my Gulp publishing process to test if Github actions work with this.

The idea is that every time we push a commit or we close a Pull Request, Github Actions will run the Gulp default build process from the content in the `main` branch.

See the Github Actions Documentation for more information about the tool, how it works and what you can do with it.

## The First Try

In the first pass, we want to validate that the project is set up correctly and that can run the build process inside the action.

```
name: NodeJS with Gulp

on:
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v2

    - name: Use Node.js 14.x
      uses: actions/setup-node@v1
      with:
        node-version: 14.x

    - name: Build
      run: |
        npm install
        gulp
```

The idea remains the same. We want to run the build process for every commit and every pull request made on the `main` branch.

We run the build process usint the `ubuntu-latest` runner available from Github.

We use two actions to run the process:

- Checkout will checkout the code from the repository's `main` branch
- Setup Node will setup one or more versions of Node.js to run the actions with.

We then create a run action to install the packages and run the default Gulp build task.

## Publish the results

Right now we can build the files but they stay inside the action and we can't see the result, even though we tell Gulp to put the files in the `dist` directory.

To do this we add two more actions, one to commit the files and a second one to push the results to the `main` branch.

We first set the user's name and email to use in the commit using `git config --local`.

We add all the files to the commit.

If the result of running `git status --porcelain` is empty (has a length of 0), then we set the workflow command to set the push flag to false. There is nothing for us to commit.

Otherwise we commit the changes and set the push flag to true.

The second action checks if the push flag is set to true. If it is it uses the GitHub Action for GitHub Push action to push the changes to the `main` branch using Github Tokens generated as part of the Actions workflow.

```
    - name: Commit files
      id: commit
      run: |
        git config --local user.email "action@github.com"
        git config --local user.name "github-actions"
        git add --all
        if [-z "$(git status --porcelain)"]; then
           echo "::set-output name=push::false"
        else
           git commit -m "Add changes" -a
           echo "::set-output name=push::true"
        fi
      shell: bash

    - name: Push changes
      if: steps.commit.outputs.push == 'true'
      uses: ad-m/github-push-action@master
      with:
         github_token: ${{ secrets.GITHUB_TOKEN }}
```

## Using Github Pages to show the results

Now we have the code generated by the Actions in the repository's main branch the final question is how to show the results to the users.

We could change the build process to point to the `docs` directory instead of `dist`. Once we do that then Github Pages, as configured in the repository settings should take care of the rest.

Another option is to use the GitHub Pages Deploy Action to deploy the `dist` folder to Github Pages.

Rather than deploy it to the `docs` folder in `main`. I've chosen to deploy it to a gh-pages branch in the repository. My first naive attempt had it push everything in the dist directory to main, without realizing that it would wipe out the entire repository.

I was able to fix it by force pushing the content back into main. Yes, there are many posts telling you that what I did is not safe and I shouldn't do it; but I'm the only one working on the project and I know I'm not losing any work, so it's OK in this case, but I wouldn't do it in a team setting.

```
    - name: Deploy to Github Pages
      uses: JamesIves/github-pages-deploy-action@4.1.5
      with:
        branch: gh-pages
        folder: dist
```

The action is simple. It tells the action what packages to use via the `uses` attribute and then, inside the `with` attribute it tells it what branch to deploy to (`branch`) and what directory to deploy from (`folder`).

## Conclusion

As far as actions go, this project is a very simple one. I like that it requires no changes to the existing code in the repository, if we want to update the way we build the HTML documents or if we want to add more features, all we have to do is add them to the gulp build file and add any third-party installation to the actios YAML file.

That said this particular project has some warnings attached to it.

The first one is that it might timeout when compressing images with Squoosh. When first developing the Gulp build process I thought it would only be used on my laptop where I was OK with taking several minutes to complete. Using Github actions I'm not certain if it's a good idea and whether it will timeout on the assigned runner. So far it hasn't but the more images we add, the longer it'll take.

The actions are run in a Ubuntu machine. Trying to run them on a macOS runner causes errors. So far there is no mac-specific thing in the build process. That may change.

If none of these issues affect you, actions are a great way to automate your workflow.