**Practical Exercise 26 - Adding the Logic to Create PRs**

**Exercise Description**

**In this practical exercise, our goal is to finish the JavaScript part of our code in order to correctly create PRs from within our custom action.**

Here are the instructions for the exercise:

1. Extend the file named 17-2-custom-actions-js.yaml by:
   1. Extending the permissions of the GITHUB\_TOKEN token by adding a top-level permissions key with two parameters, contents and pull-requests, both set to write.
   2. Pass the value of secrets.GITHUB\_TOKEN to the gh-token input of the custom action.
2. Allow GitHub Actions to create PRs by modifying the repository settings as follows:
   1. Click on Settings at the top-right of the menu tabs in the repository page.
   2. On the left-side menu, click on Actions and then on General.
   3. Scroll to the bottom until the Workflow permissions header.
   4. Tick the box next to Allow GitHub Actions to create and approve pull requests.
3. **[Optional - If you don't want to code in JavaScript, simply copy the code from the link in the resources of this lecture]** Update the index.js file to run the following commands if the stdout of the git status command is not empty:
   1. Run a git command to change to the new branch provided via the target-branch input.
   2. Add both the package.json and package-lock.json files to the staged files for a commit.
   3. Commit both files with whatever message you see fit.
   4. Push the changes to the remote branch provided via the target-branch input. You might have to add a -u origin ${targetBranch} after git push for it to work properly.
   5. Open a PR using the Octokit API. Here is the snippet necessary to open the PR:
      1. // At the beginning of the file, import the @actions/github package
      2. const github = require('@actions/github');

      5. // Remaining code

      8. const octokit = github.getOctokit(ghToken);
      10. try {
      11. await octokit.rest.pulls.create({
      12. owner: github.context.repo.owner,
      13. repo: github.context.repo.repo,
      14. title: `Update NPM dependencies`,
      15. body: `This pull request updates NPM packages`,
      16. base: baseBranch,
      17. head: targetBranch
      18. });
      19. } catch (e) {
      20. core.error('[js-dependency-update] : Something went wrong while creating the PR. Check logs below.');
      21. core.setFailed(e.message);
      22. core.error(e);
      23. }
4. Commit the changes and push the code. Trigger the workflow from the UI, and take a few moments to inspect the output of the workflow run. What happened when the workflow was run a second time and the PR was already open?

Resources for this lecture: <https://github.com/lm-academy/github-actions-course/blob/cc6735c4934f1584c6e5b16fe2e6b938b6f66b41/.github/actions/js-dependency-update/index.js>