Google Developer Student Clubs

Automatyzacja z Git Hooks

WARSZTAT

Krzysztof Kaczyński
GDSC Lead

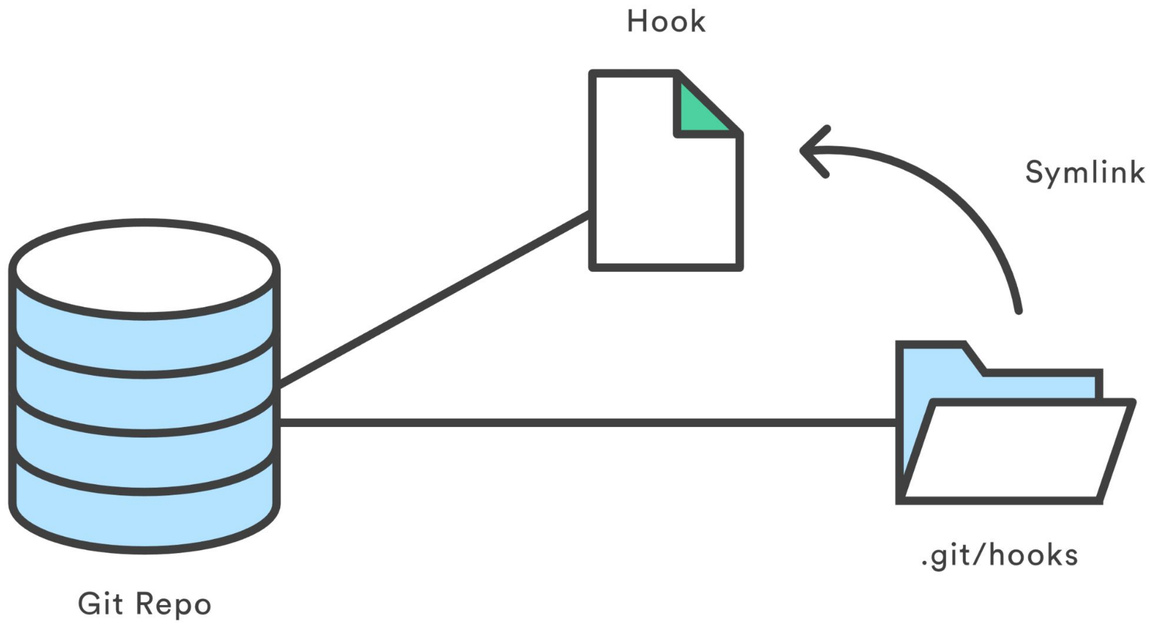
Agenda

- Czym jest Git Hook,
- Do czego można korzystać z Git Hook,
- Praktyczny przykład automatyzacji sprawdzania jakości kodu,
- Praktyczny przykład automatyzacji linkowania issue z Jiira/GitHub

Czym jest Git Hook

`pre-commit, prepare-commit-msg, update`

Git Hooks to programy, które mogą zostać wywołane w pewnym momencie wykonywania poleceien git



Git Hooks use case

`pre-commit, prepare-commit-msg, update`

- Sprawdzanie jakości kodu przed zacommitowaniem
- Sprawdzanie, czy aplikacja przechodzi testy przed wypchnięciem zmiany
- Sprawdzanie, czy testy przechodzą, zanim zmiana znajdzie się na repozytorium zdalnym

pre-commit

```
eslint_autofix_flag=""
stylelint_autofix_flag=""
ends_with_ts_or_tsx=".*\.tsx?$"
ends_with_scss=".*\.scss$"
eslint_exit_code=0
stylelint_exit_code=0
not_staged_ts_tsx_files=$(git diff --diff-filter=b --name-only | grep -E "$ends_with_ts_or_tsx")
not_staged_scss_files=$(git diff --diff-filter=b --name-only | grep -E "$ends_with_scss")
staged_ts_tsx_files=$(git diff --cached --diff-filter=d --name-only | grep -E "$ends_with_ts_or_tsx")
staged_scss_files=$(git diff --cached --diff-filter=d --name-only | grep -E "$ends_with_scss")
ts_tsx_filter_regex=$(create_pattern "${not_staged_ts_tsx_files[*]}")
scss_filter_regex=$(create_pattern "${not_staged_scss_files[*]}")
ts_tsx_files_to_add_after_linting=$(filter_array_with_inverted_regEx "${staged_ts_tsx_files[*]}"
"${ts_tsx_filter_regex}")
scss_files_to_add_after_linting=$(filter_array_with_inverted_regEx "${staged_scss_files[*]}"
"${scss_filter_regex}")

if [ ${#ts_tsx_files_to_add_after_linting} = ${#staged_ts_tsx_files} ]
then
eslint_autofix_flag="--fix"
fi
if [ ${#scss_files_to_add_after_linting} = ${#staged_scss_files} ]
then
stylelint_autofix_flag="--fix"
fi
if [[ -n "$staged_ts_tsx_files" ]]
then
   echo "Running ESLint..."
   ./node_modules/.bin/eslint $staged_ts_tsx_files --quiet $eslint_autofix_flag
   eslint_exit_code=$?
fi
if [[ -n "$staged_scss_files" ]]
then
   echo "Running Stylelint..."
   ./node_modules/.bin/stylelint $staged_scss_files --stdin --quiet $stylelint_autofix_flag
   stylelint_exit_code=$?
fi
if [ $eslint_exit_code = 0 ] && [ $stylelint_exit_code = 0 ]
then
   add_files_to_staged_tree "${ts_tsx_files_to_add_after_linting[*]}
${scss_files_to_add_after_linting[*]}"
   echo -e "${GREEN} Committed successfully ${NC}"
   exit 0
else
   echo -e "${RED} Couldn't commit changes dues to above errors ${NC}"
   exit 1
fi
```

Google Developer Student Clubs

prepare-commit-msg

```
COMMIT_MSG_FILE=$1
issue_id=$(git symbolic-ref --short HEAD | grep -o '^[[:digit:]]*')
cat $COMMIT_MSG_FILE | grep -q GH-$issue_id
if [ $? = 1 ] && [[ -n "${issue_id// /}" ]]; then
 echo >>$COMMIT_MSG_FILE
 echo "Related work item: GH-$issue_id" >>$COMMIT_MSG_FILE
fi
```

Grafiki użyte w prezentacji

Git hook
https://www.atlassian.com/git/tutorials/git-hooks

Materiały

Artykuł na temat git hook
https://dev.to/krzysztofkaczy9/do-you-really-need-husky-247b

Repozytorium z git-hooks
https://github.com/kaczor6418/git-hooks-example

Dokumentacja git-hooks
https://git-scm.com/docs/githooks