perplexity

a standardowe include jak działa?

Keyword `include` pozwala importować konfigurację z innych plików YAML do twojego pipeline'u. Istnieje kilka typów include'ów, które działają w różny sposób.[1] [2] [3]

Typy `include`

`include:local`

Includuje pliki z tego samego repozytorium:[3]

```
include:
  - local: 'templates/.gitlab-ci-template.yml'
  - local: '/ci/build.yml'
```

Możesz używać wildcardów:[3]

```
include: 'configs/*.yml'          # wszystkie .yml w katalogu configs
include: 'configs/**.yml'         # configs i wszystkie podkatalogi
```

`include:remote`

Includuje pliki ze zdalnych URLi przez HTTP(S):[4] [3]

```
include:
  - remote: 'https://gitlab.com/awesome-project/raw/main/.gitlab-ci-template.yml'
```

`include:project`

Includuje pliki z innych projektów w tej samej instancji GitLab:[3]

```
include:
  - project: 'my-group/my-project'
    ref: main
    file: 'templates/.gitlab-ci-template.yml'
```

`include:template`

Includuje gotowe szablony dostarczane przez GitLab:[3]

```
include:
  - template: Auto-DevOps.gitlab-ci.yml
```

Mergowanie i nadpisywanie konfiguracji

Include używa **deep merge** - konfiguracja z includowanych plików jest łączona z głównym `.gitlab-ci.yml`:[3]

```
# common.yml
variables:
  POSTGRES_USER: common_username
  POSTGRES_PASSWORD: testing_password

test:
  script:
    - rake spec

# .gitlab-ci.yml
include: 'common.yml'

variables:
  POSTGRES_USER: username  # nadpisuje wartość z common.yml

test:
  artifacts:               # dodaje artifacts do joba test
    reports:
      junit: rspec.xml
```

Ważne: Tablice (`script`, `rules`) nie są mergowane - są całkowicie nadpisywane przez główny plik.[3]

Warunkowe includowanie z `rules`

Możesz użyć `rules` do warunkowego includowania plików:[3]

```
include:
  - local: builds.yml
    rules:
      - if: $INCLUDE_BUILDS == "true"

  - local: deploys.yml
    rules:
      - if: $CI_COMMIT_BRANCH == "main"

  - local: tests.yml
    rules:
```

```
      - exists:
          - test-file.md
```

Zagnieżdżone include'y

Include'y mogą być zagnieżdżone - plik includowany może sam includować inne pliki, maksymalnie do **150 poziomów**:[3]

```
# .gitlab-ci.yml
include:
  - local: /.gitlab-ci/another-config.yml

# another-config.yml
include:
  - local: /.gitlab-ci/config-defaults.yml
```

⁂

1. https://docs.gitlab.com/ci/yaml/
2. https://gitlab-docs-d6a9bb.gitlab.io/ee/ci/yaml/
3. https://www.testmo.com/guides/gitlab-ci-parallel-test-automation/
4. https://www.hifis.net/workshop-materials/gitlab-ci/episodes/08-using-includes/
5. https://docs.gitlab.com/ci/examples/
6. https://docs.gitlab.co.jp/ee/ci/examples/
7. https://docs.gitlab.com/ci/yaml/includes/
8. https://gitlab.cn/docs/14.0/ee/ci/yaml/includes.html
9. https://docs.gitlab.co.jp/ee/ci/yaml/includes.html
10. https://docs.gitlab.co.jp/ee/ci/yaml/
11. https://docs.gitlab.com/ci/ci_cd_for_external_repos/