

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Практическая работа 6

Выполнил:

Никитин Павел

Группа

K33402

Проверил:

Добряков Д. И.

Санкт-Петербург

2022 г.

Задача

Необходимо настроить автодеплой (с триггером на обновление кода в вашем репозитории, на определённой ветке) для вашего приложения на удалённый сервер с использованием Github Actions или Gitlab CI (любая другая CI-система также может быть использована).

Ход работы

Для этой практической я решил взять проект с работы, так как у нашей команды много публичных репозиторий, нам нужно было выстроить единый процесс проверок(линтинг + форматирование) и публикации, причем с возможностью депрекейта старых версий выкатки альфа и бета версий.

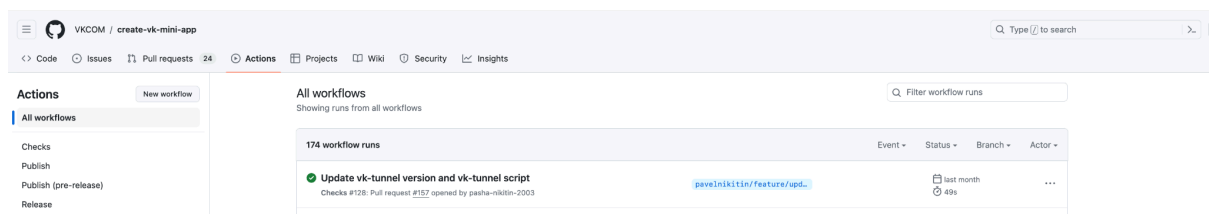


Рисунок 1 - actions для обновления версий, проверок

```
name: 'Checks'

on:
  pull_request:

jobs:
  lint:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
        with:
          token: ${ secrets.DEVTOOLS_GITHUB_TOKEN }
      - uses: actions/setup-node@v4
        with:
          node-version: 20
          cache: yarn
      - name: Install Node.js dependencies
        run: yarn install
      - name: Lint Prettier
        run: yarn lint:prettier
      - name: Lint ESLint
        run: yarn lint:eslint

  check_script:
    runs-on: ubuntu-latest
    strategy:
      matrix:
        node-version: [18.x, 20.x, 'latest']
    needs: lint
    steps:
      - name: Checkout repository
        uses: actions/checkout@v3
        with:
          token: ${ secrets.DEVTOOLS_GITHUB_TOKEN }
          ref: ${ github.head_ref }
      - name: Install Node.js and dependencies
        uses: actions/setup-node@v4
        with:
          node-version: ${ matrix.node-version }
          cache: yarn
      - name: Install dependencies
        run: yarn install && yarn run build && yarn link
      - name: Run create-vk-mini-app script
        run: create-vk-mini-app mini-app --typescript --template=vkui-only
```

Рисунок 2 - action проверок

Проверки триггерятся на каждый pull request, после этого запускаются проверки eslint + prettier, далее на разных версиях nodejs мы запускаем создание мини аппа и таким образом проверяем что ничего не сломали.

```
name: 'Publish (pre-release)'\n\non:\n  workflow_dispatch:\n    inputs:\n      new_version:\n        description: 'version type (use prerelease for bumping pre-release version):'\n        required: true\n        type: choice\n        default: 'prerelease'\n        options:\n          - prepatch\n          - preminor\n          - premajor\n          - prerelease\n      npm_tag:\n        required: true\n        description: 'NPM tag:'\n        type: choice\n        default: 'beta'\n        options:\n          - alpha\n          - beta\n          - rc\n      custom_version:\n        description: 'use syntax x.y.z-beta.0, without "v" (it ignore "version type" parameter):'\n        required: false\n\nrun-name: Publish ${ inputs.new_version } create-vk-mini-app ${ inputs.custom_version }\n\njobs:\n  publish:\n    runs-on: ubuntu-latest\n    steps:\n      - name: Checkout\n        uses: actions/checkout@v4\n        with:\n          token: ${ secrets.DEVTOOLS_GITHUB_TOKEN }\n\n      - name: Setting up the repository environment\n        uses: ./.github/actions/setup\n\n      - name: Bump version\n        id: updated_versions_info\n        uses: ./.github/actions/bump-version\n        with:\n          version: ${ github.event.inputs.custom_version || github.event.inputs.new_version }\n          npm_tag: ${ github.event.inputs.npm_tag }\n\n      - name: Complete publish\n        uses: ./.github/actions/complete-publish\n        with:\n          branch: ${ github.ref }\n          prev_version: ${ steps.updated_versions_info.outputs.prev_version }\n          next_version: ${ steps.updated_versions_info.outputs.next_version }\n          npm_tag: ${ github.event.inputs.npm_tag }\n          token: ${ secrets.DEVTOOLS_GITHUB_TOKEN }\n          npm_token: ${ secrets.NPMJS_PUBLISH_TOKEN }
```

Рисунок 3 - pre publish скрипт

Здесь мы можем выбрать ветку для релиза + тег + номер версии и при помощи секретов публиковать изменения.

Вывод

Я научился использовать github actions - это помогло мне настраивать качественный ci/cd