Build Documentation

Sarah Julia Kriesch

July 24, 2022

General Build Information

There are two build paths available in our project "OpenID Connect Doctor". Developers should use the "Build with the command line" to test their software and for a review before every release.

Then there is an automated build path included into the CI/CD pipeline with pkg to create binaries for MacOS, Windows and Linux. These binaries will be created automatically with every release and are also available for downloads under Assets of our releases. The names are openid-doctor-linux, openid-doctor-macos and openid-doctor-win.exe.

Manual Builds

Developers or people, who want to build the application manually via the command line based on the main branch, should have installed Node.js Version 16.15.0 or later. Nodejs includes the command npm, which is required for manual builds.

All required npm packages (referenced in our package.json as dependencies will be installed with:

```
$ npm install
```

Afterwards the build process can be started with:

```
$ npm run build
```

The application will be opened with the following command then:

```
$ npm start
```

Automated Build

There has been the requirement to create binaries for executing after every release. The most efficient method is the integration of automated builds in the CI/CD pipeline within the release cycle. That has been done with pkg[1]. The installation will integrate some additional dependencies for package-lock.json.

Pkg has to be referenced in the file package.json under "scripts" with:

```
"binary": "pkg package.json"
```

[1] https://github.com/vercel/pkg

Then the target architectures have to be defined in the same file, the same as the content which has to be packaged:

```
”bin”: ”bundle/index.js”,
 ”pkg”: {
    ”assets”: [
      ”views/**/*”,
      ”schema/**/*”,
      ”public/**/*”,
      ”logfiles/**/*”,
      ”node_modules/hbs/**/*.js”
    ],
    ”targets”: [
      ”node16–linux–x64”,
      ”node16–win–x64”,
      ”node16–macos–x64”
    ]
```

Afterwards, the automated build of binaries should have been integrated into the CI/CD pipeline of Github.

That will be done with the file .github/workflows/release-binary.yml.
The release job of binaries should be executed only during release. That is defined with:

```
name: release–binary
on:
  release:
    types: [published]
```

The job is defined with executing npm commands to build the application and to upload all build artifacts with executing permissions of a Github Token:

```
jobs:
  build-binary:
    name: build-binary
    runs-on: ubuntu-latest
    steps:
      - name: Checkout code
        uses: actions/checkout@main
      - name: Setup Node.js version
        uses: actions/setup-node@main
        with:
          node-version: 16.15.0
          cache: 'npm'
      - name: Install dependencies
        run: npm install
      - name: Build application
        run: npm run build
      - name: Bundle
        run: npm run bundle
      - name: Generate Binary
        run: npm run binary
      - run: chmod -R +x openid-*
      - name: Upload artifacts
        uses: skx/github-action-publish-binaries@master
        env:
          GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
        with:
          args: 'openid-*'
```