Seminar: Softwareentwicklung mit Dev(Sec)Ops

Seminararbeit - License Checker

Jonas Kell: jonas.kell@student.uni-augsburg.de

Studiengang:	B.Sc. Informatik
Abgegeben am:	12. September 2021

Inhaltsverzeichnis

1 Diskussion: Dev-Sec-Ops

1.1 Warum Dev-Sec-Ops

1.2 Für wen ist Dev-Sec-Ops?

2 Der „ideale" Dev-Sec-Ops Prozess

2.1 Welche Werkzeuge kommen zum Einsatz?

2.2 Wie ordnet man seine Pipeline am besten an?

3 Dev-Sec-Ops Werkzeug: License Checker

3.1 Auswahl des geeignetsten Werkzeugs

3.2 Inbetriebnahme

4 Literaturverzeichnis

[1] D. Glass, *NPM License Checker*, https://www.npmjs.com/package/license-checker (besucht am 12. 09. 2021).

[2] D. Bauernfeind, *Composer License Checker*, https://github.com/dominikb/composer-license-checker (besucht am 12. 09. 2021).

5 Anhänge

.gitlab-ci.yml

```yaml
check-npm-licenses:
  image: node:16
  script:
    - /bin/bash ./license-checker/npm-licenses.sh
      "MIT;ISC;Apache-2.0;0BSD;BSD-2-Clause;BSD-3-Clause;CC0-1.
      0;CC-BY-4.0;Unlicense"
  artifacts:
    when: always
    paths:
      - npm_license_summary.txt
      - npm_license_detailed.json
    expire_in: 3 days
    reports:
      junit:
        - npm_licenses.xml

check-composer-licenses:
  image: composer:2.1.6
  script:
    - /bin/bash ./license-checker/composer-licenses.sh
      "MIT;LGPL-2.1;BSD-3-Clause;LGPL-3.0;Apache-2.0"
  artifacts:
    when: always
    paths:
      - composer_license_summary.txt
      - composer_license_detailed.txt
    expire_in: 3 days
    reports:
      junit:
        - composer_licenses.xml
```

license-checker/npm-licenses.sh

```
1   # install the npm dependency
2   npm install
3   npm install -g license-checker
4   npm install -g yui-lint
5
6   # list the summary of licenses used
7   license-checker --summary --out "npm_license_summary.txt"
8   # generate a overview of licenses
9   license-checker --json --out "npm_license_detailed.json"
10
11
12  fails=0
13  # Generate Output.xml
14  echo '<?xml version="1.0" encoding="UTF-8"?><testsuites><testsuite
    ↪  id="NPM-CUSTOM-LICENSE-CHECKER">' > npm_licenses.xml
15  echo '<testcase name="NPM check allowed Licenses">' >>
    ↪  npm_licenses.xml
16
17  # execute test and capture error stream in variable
18  ERROR=$(license-checker --onlyAllow "$1" 2>&1 >/dev/null)
19
20  if [ $? != 0 ];
21  then
22      let "fails++"
23      echo '<failure type="FAILURE">' >> npm_licenses.xml
24      echo $ERROR >> npm_licenses.xml
25      echo '</failure>' >> npm_licenses.xml
26  else
27      echo '<passed type="PASSED"></passed>' >> npm_licenses.xml
28  fi
29
30  echo '</testcase>' >> npm_licenses.xml
31  echo '</testsuite></testsuites>' >> npm_licenses.xml
32
33  # 0 if all succeeded, larger 0 otherwise
34  exit $fails
```

license-checker/composer-licenses.sh

```
1   # install the composer dependency
2   composer require dominikb/composer-license-checker
3
4   # list the summary of licenses used
5   ./vendor/bin/composer-license-checker report >
    ↪   "composer_license_summary.txt"
6
7   # generate a overview of licenses
8   # as there's no real option for this, run a check that lists
    ↪   everything and remove the Error-messages
9   ./vendor/bin/composer-license-checker check --allowlist
    ↪   NAMEOFNOLICENSE >  "composer_license_detailed.txt"
10  sed -i '/\[ERROR\]/d' composer_license_detailed.txt
11
12  # format the first operand correctly to be used in the command
13  OPTIONS=""
14  IFS=';' read -ra ADDR <<< "$1"
15  for i in "${ADDR[@]}"; do
16    OPTIONS="$OPTIONS --allowlist $i"
17  done
18
19
20  fails=0
21  # Generate Output.xml
22  echo '<?xml version="1.0" encoding="UTF-8"?><testsuites><testsuite
    ↪   id="COMPOSER-CUSTOM-LICENSE-CHECKER">' > composer_licenses.xml
23  echo '<testcase name="Composer check allowed Licenses">' >>
    ↪   composer_licenses.xml
24
25  # execute test and capture output stream in variable
26  OUTPUT=$(./vendor/bin/composer-license-checker check $OPTIONS)
27
28  if [ $? != 0 ];
29  then
30      let "fails++"
31      echo '<failure type="FAILURE">' >> composer_licenses.xml
32      echo $OUTPUT >> composer_licenses.xml
33      echo '</failure>' >> composer_licenses.xml
34  else
```

```
35      echo '<passed type="PASSED"></passed>' >> composer_licenses.xml
36  fi
37
38  echo '</testcase>' >> composer_licenses.xml
39  echo '</testsuite></testsuites>' >> composer_licenses.xml
40
41  # 0 if all succeeded, larger 0 otherwise
42  exit $fails
```