GÉRER LES DÉPENDANCES AVEC PYTHON POETRY ET SNYK

Pourquoi pas Pip?

All in one, au lieu de pip + venv + virtualenv +setuptools

Génération de poetry.lock

Environnements virtuels automatiques.

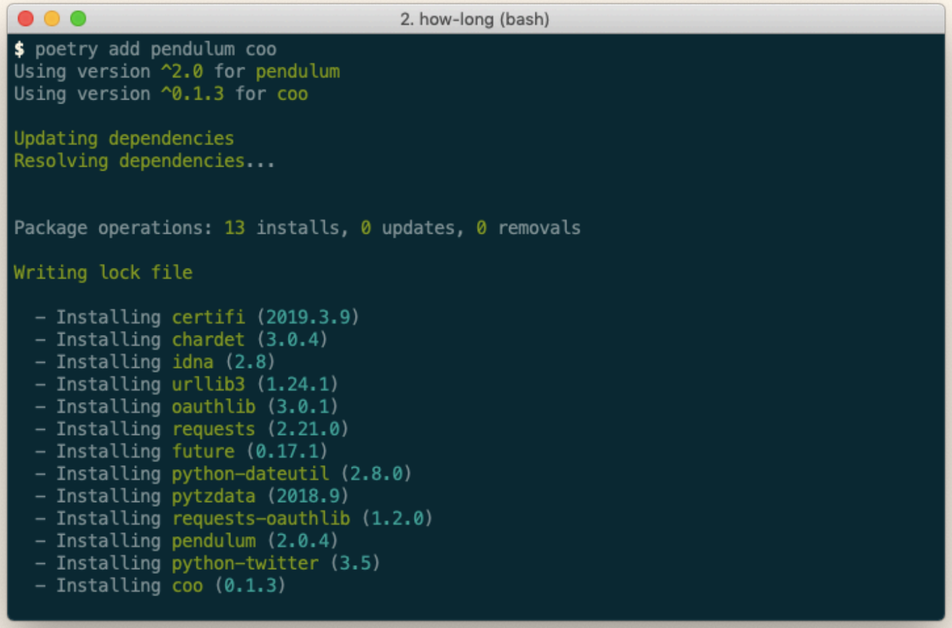
pyproject.toml pour démarrer le projet

Python Poetry

Packaging simplifié pour publier des projets Python.

Gestion des dépendances via pyproject.toml.

Environnements virtuels automatiques.



```
[build-system]
requires = ["poetry-core>=1.0.0"]
build-backend = "poetry.core.masonry.api"

[tool.poetry]
name = "my-project"
version = "0.1.0"
description = "My awesome project"
authors = ["Your Name <your@email.com>"]

[tool.poetry.dependencies]
python = "^3.9"
requests = "^2.26.0"
numpy = "^1.21.2"
matplotlib = "^3.4.3"

[build-system]
requires = ["poetry-core>=1.0.0"]
build-backend = "poetry.core.masonry.api"
```

```yaml
name: CI

on: [push, pull_request]

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
    - name: Checkout code
      uses: actions/checkout@v3

    - name: Set up Python
      uses: actions/setup-python@v4
      with:
        python-version: '3.9'

    - name: Install Poetry
      run: curl -sSL https://install.python-poetry.org | python3 -

    - name: Install dependencies
      run: poetry install

    - name: Run tests
      run: poetry run pytest

```

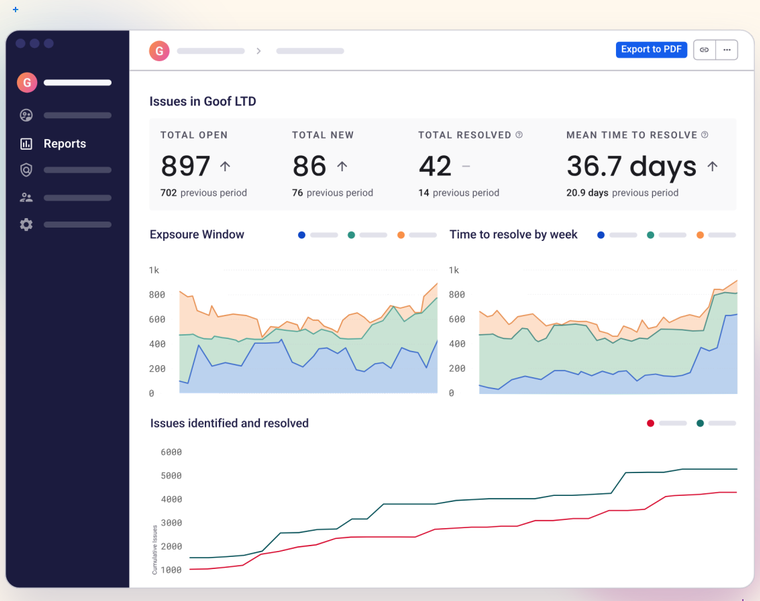
Snyk

Analyse de vulnérabilités dans les bibliothèques open-source.

Correctifs automatiques

Surveillance continue

Intégration fluide

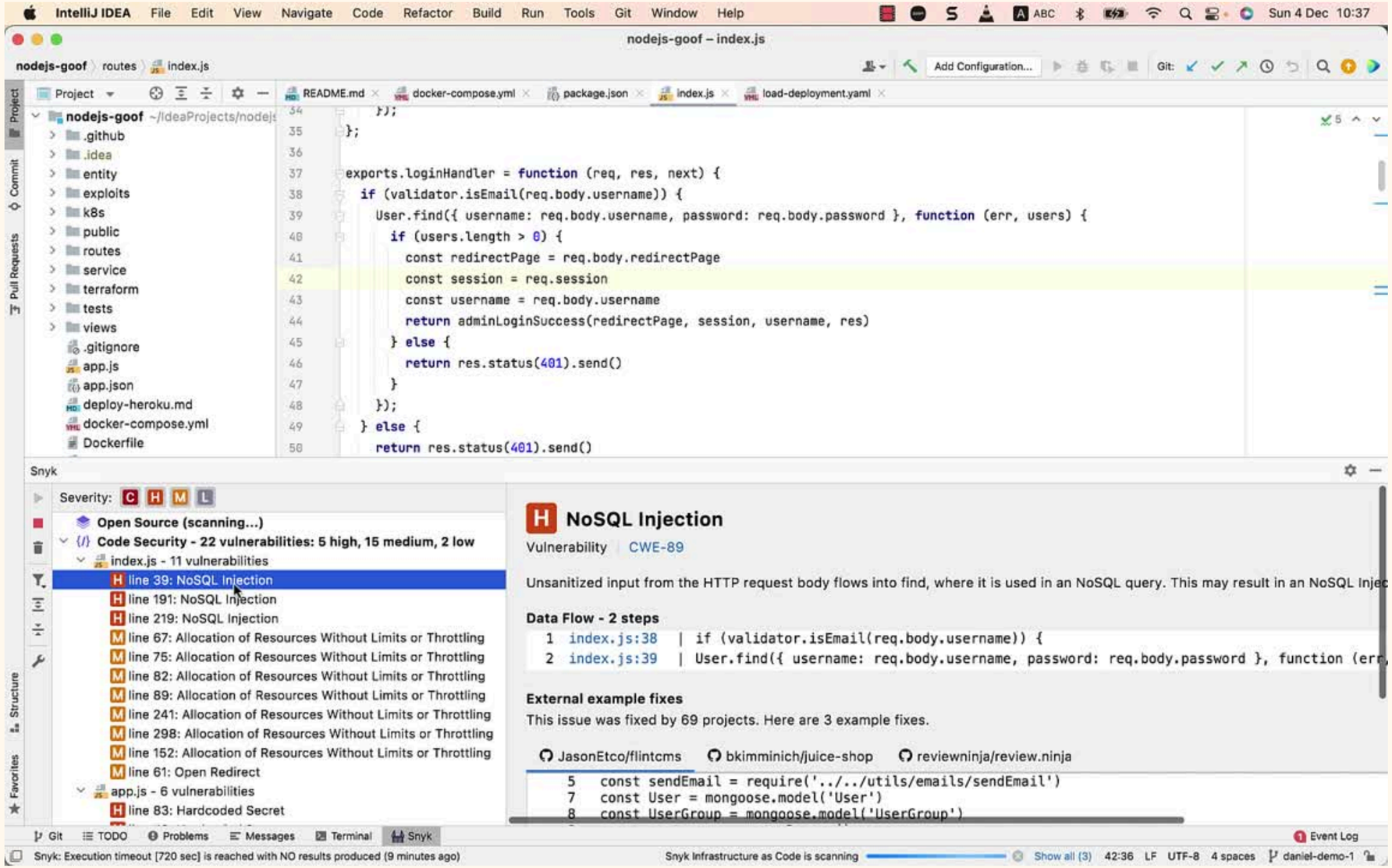


Plus!

Lien du advisor


n
Welcome
on: [push, pull_request]
jobs:
build:
runs-on: ubuntu-latest
steps:
- name: Checkout code
uses: actions/checkout@v3
- name: Set up Python
uses: actions/setup-python@v4
with:
python-version: '3.9'
- name: Install Poetry
run: curl -sSL https://install.python-poetry.org | python3 -
- name: Install dependencies
run: poetry install
- name: Run tests
run: poetry run pytest
- name: Install Snyk for Python
run: poetry add snyk
§
- name: Run Snyk to check vulnerabilities
run: poetry run snyk test
env:
SNYK_TOKEN: ${{ secrets.SNYK_TOKEN }}

Snyk


IntelliJ IDEA File Edit View Navigate Code Refactor Build Run Tools Git Window Help
Sun 4 Dec 10:37
nodejs-goof – index.js
nodejs-goof routes index.js
Add Configuration...
Git:
Project
README.md
docker-compose.yml
package.json
index.js
load-deployment.yaml
nodejs-goof ~/IdeaProjects/nodejs
.github
.idea
entity
exploits
k8s
public
routes
service
terraform
tests
views
.gitignore
app.js
app.json
deploy-heroku.md
docker-compose.yml
Dockerfile
};
exports.loginHandler = function (req, res, next) {
if (validator.isEmail(req.body.username)) {
User.find({ username: req.body.username, password: req.body.password }, function (err, users) {
if (users.length > 0) {
const redirectPage = req.body.redirectPage
const session = req.session
const username = req.body.username
return adminLoginSuccess(redirectPage, session, username, res)
} else {
return res.status(401).send()
}
});
} else {
return res.status(401).send()
Snyk
Severity: C H M L
Open Source (scanning...)
Code Security - 22 vulnerabilities: 5 high, 15 medium, 2 low
index.js - 11 vulnerabilities
line 39: NoSQL Injection
line 191: NoSQL Injection
line 219: NoSQL Injection
line 67: Allocation of Resources Without Limits or Throttling
line 75: Allocation of Resources Without Limits or Throttling
line 82: Allocation of Resources Without Limits or Throttling
line 89: Allocation of Resources Without Limits or Throttling
line 241: Allocation of Resources Without Limits or Throttling
line 298: Allocation of Resources Without Limits or Throttling
line 152: Allocation of Resources Without Limits or Throttling
line 61: Open Redirect
app.js - 6 vulnerabilities
line 83: Hardcoded Secret
NoSQL Injection
Vulnerability CWE-89
Unsanitized input from the HTTP request body flows into find, where it is used in an NoSQL query. This may result in an NoSQL Injec
Data Flow - 2 steps
1 index.js:38 | if (validator.isEmail(req.body.username)) {
2 index.js:39 | User.find({ username: req.body.username, password: req.body.password }, function (err,
External example fixes
This issue was fixed by 69 projects. Here are 3 example fixes.
JasonEtco/flintcms
bkimminich/juice-shop
reviewninja/review.ninja
5 const sendEmail = require('../../utils/emails/sendEmail')
7 const User = mongoose.model('User')
8 const UserGroup = mongoose.model('UserGroup')
Git TODO Problems Messages Terminal Snyk
Event Log
Snyk: Execution timeout [720 sec] is reached with NO results produced (9 minutes ago)
Snyk Infrastructure as Code is scanning
Show all (3)
42:36 LF UTF-8 4 spaces
daniel-demo-1

Pourquoi Snyk et Poetry ensemble?

Poetry gère les dépendances de manière efficace et reproductible.

Snyk garantit la sécurité des dépendances dès le départ.

Workflow idéal : Poetry pour la gestion des dépendances, Snyk pour la sécurité continue.

Les bénéfices

Efficacité : Simplifie la gestion des dépendances et des environnements.

Sécurité : Identifie et corrige les vulnérabilités dès l'étape de développement.

Consistance : Environnements reproductibles, de l'intégration continue à la production.

Conformité : Assure le respect des normes de sécurité.

QUESTIONS?