

Configuration du projet

Lorsque le projet est créé, les configurations suivantes ont été mise en place :

Description

Projet calculatrice avec tests associés

[Plain text] [Prévisualisation](#)

☐ Ce build a des paramètres ?

☒ GitHub project

Project url ?

<https://github.com/AlexandreCotier/b3-c1-jenkins-cotier-baudequin-merle/>

Avancé ▾

☐ Supprimer les anciens builds ?

☐ Throttle builds ?

☐ Exécuter des builds simultanément si nécessaire ?

Avancé ▾

Git ?

Repositories ?

Repository URL ?

<https://github.com/AlexandreCotier/b3-c1-jenkins-cotier-baudequin-merle/>

Credentials ?

- aucun - ▾

Ajouter ▾

Avancé ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

On donne à Jenkins l'url du projet Github afin qu'il puisse le récupérer.

Ce qui déclenche le build

☐ Déclencher les builds à distance (Par exemple, à partir de scripts) ?

☐ Construire après le build sur d'autres projets ?

☐ Construire périodiquement ?

☐ GitHub hook trigger for GITScm polling ?

☒ Scrutation de l'outil de gestion de version ?

Planning ?

⚠ Vouslez-vous vraiment dire "chaque minute" avec l'expression "*****"? Peut-être voulez-vous dire "H*****"?
Aurait été lancé à Tuesday, April 11, 2023 at 1:18:13 PM Coordinated Universal Time; prochaine exécution à Tuesday, April 11, 2023 at 1:18:13 PM Coordinated Universal Time.

☐ Ignore post-commit hooks ?

Jenkins étant installé en local, il n'est pas possible de mettre des webhooks sur github, j'ai donc appliqué la « Scrutation de l'outil de gestion de version », cet outil observe le repository tous les x temps donné, dans notre cas, il observera toutes les minutes afin de voir si des changements ont été opérés.

Environnements de Build

☐ Delete workspace before build starts

☐ Use secret text(s) or file(s) ?

☐ Provide Configuration files ?

☐ Add timestamps to the Console Output

☒ Ajouter le répertoire bin/ de Node/npm au PATH

NodeJS Installation

Specify needed nodejs installation where npm installed packages will be provided to the PATH

NodeJS

npmrc file

utiliser le système par défaut

Cache location

Default (~/.npm or %APP_DATA%\npm-cache)

Afin de pouvoir utiliser les commandes NPM, le plugin NodeJS a été installé, on ajoute donc npm au PATH afin d’y avoir accès dans les commandes shell.

Exécuter un script shell ?

Commande

Voir [la liste des variables d'environnement disponibles](#)

```
npm install
npm run test
```

Avancé

Ajouter une étape au build

Dans les « build steps » on renseigne donc npm install, qui nous permettra d’installer toutes les dépendances du projet et npm run test, qui a été défini dans le projet pour exécuter les tests avec Mocha.

```
> b3-dev-tu-cotier-alexandre@1.0.0 test
> mocha

Testing the Calculator Functions
  1) 1. Sum
    ✓ 2. Multiply
    ✓ 3. Subtract
    ✓ 4. Division
    ✓ 5. Percentil
    ✓ 6. Sum float
    ✓ 7. Multiply float
    ✓ 8. Division float
    ✓ 9. Division float result
    ✓ 10. Division zero
    ✓ 12. ToPower
    ✓ 13. ToPower2
    ✓ 14. Square root tests

12 passing (7ms)
1 failing

1) Testing the Calculator Functions
   1. Sum:

    AssertionError: expected 6 to equal 9
    + expected - actual

    -6
    +9

    at Context.<anonymous> (test/app.test.js:6:40)
    at process.processImmediate (node:internal/timers:478:21)

Build step 'Execute shell' marked build as failure
The recommended git tool is: NONE
No credentials specified
Finished: FAILURE
```

Lorsque Mocha lève une exception, Jenkins l'interprète et définit le test comme échoué.

```
> b3-dev-tu-cotier-alexandre@1.0.0 test
> mocha
```

Testing the Calculator Functions

- ✓ 1. Sum
- ✓ 2. Multiply
- ✓ 3. Subtract
- ✓ 4. Division
- ✓ 5. Percentil
- ✓ 6. Sum float
- ✓ 7. Multiply float
- ✓ 8. Division float
- ✓ 9. Division float result
- ✓ 10. Division zero
- ✓ 12. ToPower
- ✓ 13. ToPower2
- ✓ 14. Square root tests

13 passing (5ms)

Lorsque Mocha ne lève aucune exception, Jenkins défini le test comme réussi.