

Devops – TP initiation

<http://bit.do/leadtechgob>

I. Préparation du projet

1. Installez node JS si ce n'est pas déjà fait
2. Dézippez le dossier disponible [ici](#) et ouvrez le dans une console
3. Exécutez "npm install"
4. Exécutez "npm test", tous les tests unitaires devraient fonctionner

II. Ajout à github

1. Créez un compte [Github](#) si vous n'en avez pas un
2. Créez un repository sur votre github
3. Ajoutez le projet à votre compte github (les instructions sont notées sur votre nouveau repository ("...or push an existing repository from the command line"))

III. Initialisation de travis

1. Allez sur travis-ci.org, autorisez l'accès à votre github

Travis-ci est un outil d'intégration continue, de multiples autres existent, la quasi-totalité fonctionnent de la même manière, c'est-à-dire un fichier de configuration .yaml qui définit le fonctionnement

2. Forcez une synchronisation sur la page d'accueil une fois connecté

Vous devriez voir votre repository

3. Activez votre repository

Maintenant, travis va recevoir un « Hook » à chaque commit, il regardera alors si vous avez un fichier .travis.yaml et s'exécutera en fonction de ce fichier

4. Créez un fichier .travis.yaml à la racine de votre repository, ce fichier contiendra le contenu ci-dessous (code venant d'ici)

```
language: node_js
```

```
node_js:
```

```
- "8.6.0"
```

5. Commit puis push de vos changements.

*En regardant sur travis-ci, vous pourrez voir votre projet se build automatiquement.
Vous avez mis en place une solution d'automation de vos tests unitaires*

IV. Mise en place de git flow

1. Install git flow (disponible [ici](#))
2. Initialisez votre repository avec git flow en executant « git flow init »
3. Gardez la configuration de base de git flow.

Vous devriez maintenant être sur develop.

Rappel :

-Develop = staging dans notre exemple

-Master = production

Dans le lien donné ci-dessus, vous trouverez toutes les commandes utiles à git flow (notamment les releases)

V. Automation de mise en production/staging

1. Créez un nouveau compte sur Heroku.

Heroku est un service d'hosting comme un autre, gratuit, ce qui nous est utile dans ce cas

2. Créez une nouvelle app sur le site heroku appelée XXX-staging.
3. Créez une seconde app pour la production
4. Récupérez votre api key depuis les settings d'heroku.
5. D'après l'article disponible [ici](#), deployez pour la production lorsqu'un commit est fait sur la branche master et deployez sur staging lorsqu'un commit est fait sur develop.
6. Modifier le titre de votre application (Express App Testing Demo) en « My own demo »

7. Faites un commit de vos changements sur develop puis un push.

Vérifiez que cela a bien été pris en compte sur votre app heroku staging.

8. Faites une release avec git flow (explication ci-dessous)
 - git flow release start VOTRENOMDERELEASE
 - git flow release finish VOTRENOMDERELEASE
 - Sauvegardez le premier fichier text (nom de commit de merge depuis votre branch release sur master)
 - Tapez votre tag de release (MAJOR.MINOR.BUILD) et un petit message en dessous selon votre envie (en général ce sont les releases notes avec breaking changes) puis sauvegardez
 - Sauvegardez ce dernier fichier text (nom de commit de votre release vers develop)
 - Allez sur develop (git checkout develop), git push
 - Allez sur master, push

- Faites un git push --tags pour pousser votre release

VI. Outil de suivi de qualité (code coverage)

1. Créez un compte sur coveralls.io
2. Ajoutez votre repository
3. Modifiez votre travis.yml pour faire fonctionner coveralls comme expliqué lorsque vous cliquez sur votre repository

Le code coverage est un indicateur intéressant de qualité de code. Il permet de connaître tous les lignes que vos tests unitaires n'ont pas parcouru.

VII. Encryption

Afin de simplifier le TP, les api keys sont en clair dans votre repository. Cela ne doit SURTOUT PAS ARRIVER.

Vous pouvez assez facilement, grâce à travis, encrypter vos API KEY.

...

Faites-le (voir [ICI](#))

Vous pouvez supprimer vos api key et les renew avant de les encrypter. Comme cela elle ne seront plus visibles publiquement.