

Formation Git Avancée

Exercices - les Hooks



Scénario

Votre équipe souhaite automatiser certaines tâches en passant par exemple des testU sur le code lors de la mise en gestion de conf

Objectifs

1- Mettre en place un hook qui permet de faire des testU automatiques lors d'une tentative de commit. Si les testU sont en échec, empêcher le commit. Pour ce TP, le test consistera à vérifier que le dossier images est bien présent dans le WD.

2- Mettre en place un hook qui oblige d'avoir, dans les messages de commit, un numéro de fait technique sous la forme **"id: msg"**: *exemple "123: correction du crash"*

Exercice

EXO 2

Scénario

Votre équipe souhaite mettre en place une sécurité sur le dépôt du projet via les hooks. l'objectif est alors d'éviter que les développeurs puissent supprimer un tag sur le dépôt distant.

Objectifs

1- Mettre en place un hook dans le dépôt DISTANT qui permet de vérifier si le push correspond à une création de tag. Dans ce cas, vérifier que le tag possède bien une annotation (git tag -a -m"msg"). Pour cela, utiliser la configuration hooks.allowunannotated. Si ce n'est pas le cas, refuser le push

2- Mettre en place un hook dans le dépôt DISTANT qui permet de vérifier si le push correspond à une suppression de tag. Dans ce cas, vérifier que la configuration hooks.allowdeletetag est à true. Dans le cas contraire, refuser le push.

Il faut ici partir des hooks .sample

Exercise

AIDE

SCRIPT TEST_U

```
DIRECTORY=./images

echo "---- Start TEST-U"
if [ ! -d "$DIRECTORY" ]; then
    echo "> TEST-U failed: commit rejected"
    exit 1
fi
echo "> TEST-U ok"
```

SCRIPT ISSUE ID

```
echo "---- Check issue ID in $1"
test "" != "$(egrep '^[0-9]+:' "$1")" || {
    echo "> No issue ID found".
    exit 1
}
echo "> issue ID = $(egrep -o '^[0-9]+' "$1")"
```

Bilan

Local

Remote

