Versionado

1) Preparación

Es necesario que el hook loca tengan permiso en la raíz del proyecto

- chmod +x /hooks/pre-push

Nota: Si falla

- chmod +x hooks/pre-push

Es necesario que git apunte al directorio hocks que está en la raíz del proyecto (Por defecto lo hace .git/hocks)

- git config core.hooksPath hooks

```
# RECETA PRE-PUSH

#!/bin/sh

# Leer datos de entrada estándar proporcionados por Git
while read local_ref local_sha remote_ref remote_sha
do
    # Verificar si se está realizando un push de tags
    if [[ "$remote_ref" == refs/tags/* ]]; then
        echo "Ejecución de 'git push --tags' detectada. No se ejecutará el
script."
        exit 0
    fi
done

echo "Ejecutando npm run release antes de realizar el commit..."

# Ejecutar npm run release con --no-verify para evitar bucles
npm run release -- --no-verify

# Capturar el código de salida de npm run release
npm_run_release_status=$?

# Verificar el código de salida de npm run release
if [ $npm_run_release_status -ne 0 ]; then
    echo "Error: Fallo al ejecutar npm run release. Abortando script."
    exit 1
fi

# Obtener la nueva versión del package.json después de ejecutar npm run
release
new_version=$(grep '"version"' package.json | sed -E
's/.*"([0-9]+\.[0-9]+\.[0-9]+)".*/\1/')
```

```
echo "Nueva versión extraída de package.json: v$new_version"

# Crear un tag con la nueva versión
git tag -a "v$new_version" -m "Release version $new_version"
echo "Se agrega tag al despliegue v$new_version. Para enviar tags ejecute:
'git push --tags'"

# Añadir los archivos modificados por npm run release al staging area
git add package.json package-lock.json CHANGELOG.md

# Continuar con el commit
git commit -m "fix: Descripción de los cambios realizados"
echo "Commit: Nueva versión package.json y cambios en CHANGELOG.md"

# Verificar si estamos ejecutando el script desde un push convencional
# Si es un push convencional, no ejecutar git push --tags para evitar el
bucle
if [[ "$1" != "--tags" ]]; then
    git push --tags -f
fi

exit 0
```

2) Agregar libreria

```
npm install standard-version
```

3) crear archivo Changelog si no existe

En la raíz crear un archivo vacío llamado CHANGELOG

```
CHANGELOG.md
```

4) Ejecutar versionado

Una vez el Front esté preparado para Versionado, debe agregar todos los cambios por enviar y se debe ejecutar los siguientes comandos;

```
  * git add .
  * git commit -m 'fix: cambios realizados'
  * git push
```

Nota: Esto genera una nueva versión en el package.json con tag para Git, Docker.

Importante

Para los siguientes casos se debe utilizar **git push –no-verify**

- Si están trabajando en su hotfix local y necesita subir un commit sin generar el versionado final.

- Si se requiere hacer un push tradicional sin que le afecte el hook y por ende no genere versionado se debe utilizar la siguiente bandera git push –no-verify
- También aplica para posibles problemas con la receta hook .
- Si no ha preparado el proyecto local para el hook como se indicó anteriormente.

From:
http://wiki.adacsc.co/ - **Wiki**

Permanent link:
http://wiki.adacsc.co/doku.php?id=ada:howto:sicoferp:factory:new-migracion-sicoferp:front:versionado&rev=1722532031

Last update: **2024/08/01 17:07**