



Git clone repos

Clonar repositórios com filtro de empresa e nome do projeto

1. Instalar o `jq`

```
brew install jq
```

2. Criar o arquivo

```
touch clone_repos.sh  
code clone_repos.sh
```

3. Adicionar o script

```
#!/bin/zsh  
  
# Substitua pelo seu token de acesso pessoal do GitHub  
github_token="ghp_"  
  
# Configurações do repositório  
org_name="viavarejo-internal"  
repo_filter="vv-viaunica"  
destination_folder=~/.development/git/via2  
  
# Verifique se a pasta de destino existe e crie-a se necessário  
if [[ ! -d $destination_folder ]]; then  
    echo "Criando a pasta de destino: $destination_folder"  
    mkdir -p $destination_folder  
fi
```

```

# Função para obter repositórios usando a API GraphQL
get_repos_graphql() {
    query=$(cat <<-GQL
    {
        organization(login: "$org_name") {
            repositories(first: 100) {
                nodes {
                    name
                    url
                }
            }
        }
    }
    GQL
    )

    response=$(curl -H "Authorization: token $github_token" -X POST \
    echo "$response"
}

# Obtenha a lista de repositórios
echo "Obtendo repositórios..."
response=$(get_repos_graphql)
echo "Resposta da API GraphQL:"
echo "$response"

repos=$(echo "$response" | jq -r ".data.organization.repositories")

echo "Repositórios filtrados:"
echo "$repos"

# Clone os repositórios selecionados
for repo in ${repos}; do
    echo "Clonando $repo..."
    git clone $repo $destination_folder/$(basename -s .git $repo)
done

```

```
echo "Clonagem dos repositórios concluída."
```

4. Tornar o script executavel

```
chmod +x clone_repos.sh
```

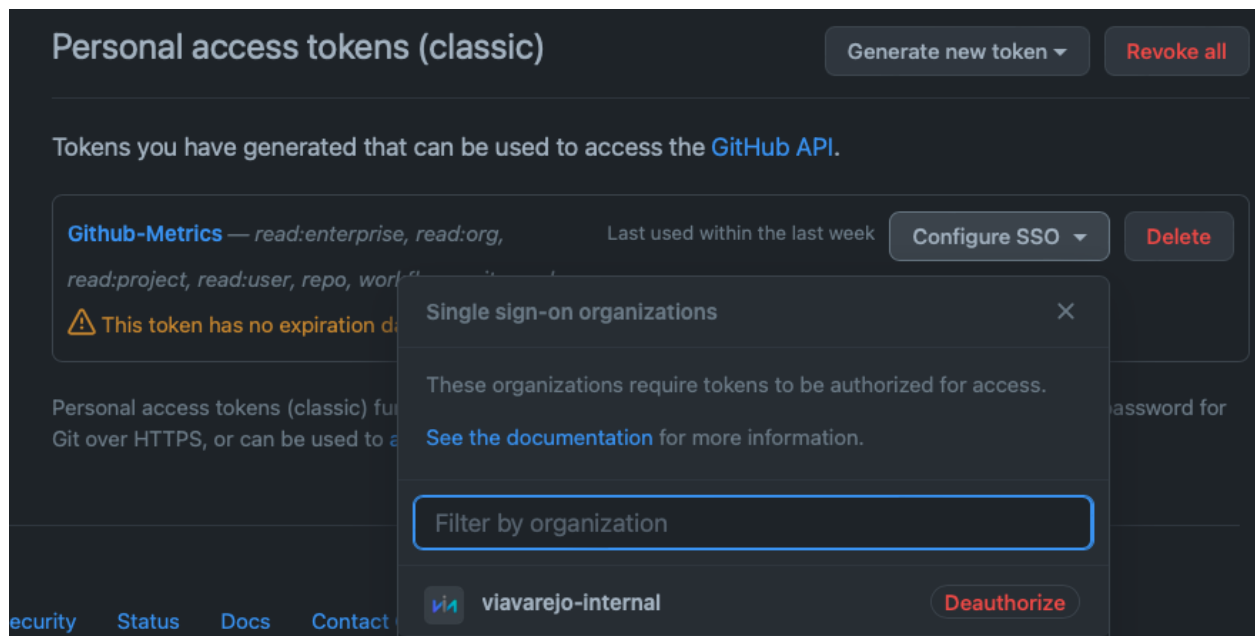
5. Executar o script

```
./clone_repos.sh
```

Github Token

Para conseguir executar o Token do Github acessar: <https://github.com/settings/tokens>

1. Acessar Developer Settings
2. Personal Access Tokens
3. Tokens Classic
4. Configure SSO



Script alternativo com variaveis

```
#!/bin/zsh

# Solicitar e armazenar informações do usuário
echo "Por favor, informe sua chave do Github:"
read github_token

echo "Por favor, informe o nome da sua organização:"
read org_name

echo "Deseja informar um filtro para o nome dos repositórios? (s/n):"
read use_filter
if [[ $use_filter == "s" ]]; then
    echo "Por favor, informe um filtro para o nome dos repositórios:"
    read repo_filter
fi

echo "Por favor, informe o diretório:"
read destination_folder

# Verificar se a pasta de destino existe e criar se necessário
if [[ ! -d $destination_folder ]]; then
    echo "Criando a pasta de destino: $destination_folder"
    mkdir -p $destination_folder
fi

# Função para obter repositórios usando a API GraphQL
get_repos_graphql() {
    query=$(cat <<-GQL
    {
        organization(login: "$org_name") {

```

```

        repositories(first: 100) {
            nodes {
                name
                url
            }
        }
    }
}
GQL
)

response=$(curl -H "Authorization: token $github_token" -X POST $url)
echo "$response"
}

# Obtenha a lista de repositórios
echo "Obtendo repositórios..."
response=$(get_repos_graphql)
echo "Resposta da API GraphQL:"
echo "$response"

repos=$(echo "$response" | jq -r ".data.organization.repositories")

echo "Repositórios filtrados:"
echo "$repos"

# Clone os repositórios selecionados
for repo in ${repos}; do
    echo "Clonando $repo..."
    git clone $repo $destination_folder/$(basename -s .git $repo)
done

echo "Clonagem dos repositórios concluída."

```