






Micro-frontends Playground — Git & Déploiement Étape Par Étape

(Init Git, .gitignore, commits, remote GitHub, branches, commandes utiles)

Contexte : Windows + PowerShell, repo local microfrontends-playground, 4 micro-apps (React/Vue/Angular/Vanilla) + root-config, monorepo pnpm.

0) Prérequis

-  **Git for Windows** installé (`git --version`)
-  Compte **GitHub**
-  Accès à PowerShell dans le dossier du projet

Vérif rapide :

```
git --version
```

1) Initialisation du dépôt local

Place-toi à **la racine** du projet :

```
cd C:\Users\remib\DEV\microfrontends-playground
git init
```

Configure ton identité (obligatoire pour commit) :

```
git config --global user.name "Rimeryx"
git config --global user.email "ton-email-github@exemple.com"
```

```
git config --global core.autocrlf true # recommandé sous Windows
```

💡 `core.autocrlf true` évite les soucis d'EOL (CRLF vs LF).

🧹 2) Nettoyage des sous-dépôts (si nécessaires)

Si des micro-apps contiennent leur propre `.git/`, supprime-les (monorepo = **un seul** dépôt) :

```
Remove-Item -Recurse -Force "mfe-react/.git"
Remove-Item -Recurse -Force "mfe-vue/.git"
Remove-Item -Recurse -Force "mfe-angular/.git"
Remove-Item -Recurse -Force "root-config/.git"
```

🧱 3) `.gitignore` (indispensable)

Crée `.gitignore` à la racine :

```
# Dependencies
node_modules/
.pnpm-store/

# Builds
dist/
build/

# Logs
*.log
npm-debug.log*
pnpm-debug.log*
yarn-error.log

# IDE / OS
.vscode/
.idea/
```

```
.DS_Store
Thumbs.db
```

✅ Empêche de pousser des fichiers lourds/inutiles.

4) Premier commit

```
git add .
git commit -m "Initial commit: Microfrontends Playground"
```

❗ Si Git te demande qui tu es → revois l'étape 1 (user.name + user.email).

5) Création du repo GitHub

1. Va sur <https://github.com/new>
2. **Repository name** : `microfrontends-playground`
3. **Public**
4. Ne coche **rien** (pas de README/.gitignore/Licence)

Récupère l'URL du repo, par ex. :

```
https://github.com/Rimeryx/microfrontends-playground.git
```

6) Lier local ↔ GitHub & pousser

```
git remote add origin https://github.com/Rimeryx/microfrontends-playground.git
git branch -M main
git push -u origin main
```

🔑 Si GitHub demande un mot de passe, utilise un Personal Access Token (PAT)

GitHub → Settings → Developer settings → **Tokens (classic)** → scope `repo`.

7) Branches de travail (recommandé)

Créer une branche `dev` pour bosser sans toucher `main` :

```
git checkout -b dev
git push -u origin dev
```

Workflow conseillé :

- `dev` → commits réguliers
- Pull Request vers `main` quand tu veux fusionner

8) Cycle de travail quotidien

Mettre à jour ta branche :

```
git pull
```

Ajouter/committer :

```
git add .
git commit -m "feat(react): ajout composant Header"
git push
```

Changer de branche :

```
git checkout main
git checkout dev
```

Lister les branches :

```
git branch -a
```

9) Commandes utiles (cheat-sheet)

Besoin	Commande
État courant	<code>git status</code>
Historique	<code>git log --oneline --graph --decorate --all</code>

Besoin	Commande
Diff des modifs	<code>git diff</code>
Annuler fichiers ajoutés	<code>git reset</code>
Annuler modifs locales (fichier)	<code>git checkout -- chemin/vers/fichier</code>
Stash (mettre de côté)	<code>git stash</code> / <code>git stash pop</code>
Changer l'URL du remote	<code>git remote set-url origin <nouvelle-url></code>
Cloner (nouveaux PC)	<code>git clone <url>.git</code>

10) Dépannage rapide

- `Author identity unknown` → configure `user.name` et `user.email` (Étape 1).
- `error: src refspec main does not match any` → aucun commit : fais le **premier commit** avant de pousser (Étape 4).
- `remote origin already exists` → change l'URL :

```
git remote set-url origin https://github.com/Rimeryx/microfrontends-playground.git
```

- **Sous-repo détecté** (`does not have a commit checked out`) → supprime les `.git/` des sous-dossiers (Étape 2).
- **Auth GitHub** → si 2FA, utilise un **PAT** comme "mot de passe".

11) Ajouter des collaborateurs (optionnel)

Sur GitHub → Repo → **Settings** → **Collaborators** → Add people.