



# Micro-frontends Playground — Guide Complet (Single-SPA + Webpack + Monorepo)

📌 Objectif : Obtenir un playground complet avec 4 micro-frontends :

- React
- Vue
- Angular
- Vanilla HTML/CSS/JS (+ jQuery)

➡️ Tous synchronisés, lancés avec **une seule commande** : `pnpm dev`



**Étape 0 — Pré-requis & installation (✓ Windows • ✓ Node.js 20+ • ✓ pnpm installée • ✓ Git installé)**

| Si tu as déjà Node et un gestionnaire de paquets, tu peux passer à l'étape 1.

## 1. Node.js 18+

- Vérifie la version : `node -v`
- Si besoin : installe via <https://nodejs.org>

## 2. Gestionnaire de paquets (au choix)

- **pnpm** (recommandé pour un monorepo) :

```
npm i -g pnpm  
pnpm -v
```

- ou **npm** (déjà inclus avec Node) : `npm -v`
- ou **yarn** : `npm i -g yarn`

### 3. Un terminal (VS Code + son terminal intégré est parfait)

## ✓ Étape 1 — Préparation du dossier

Dans ton `DEV/` :

```
mkdir microfrontends-playground  
cd microfrontends-playground
```

Créer ces deux fichiers :

📌 `pnpm-workspace.yaml`

```
packages:  
- "root-config"  
- "mfe-*"
```

📌 `package.json`

```
{  
  "name": "microfrontends-playground",  
  "private": true,  
  "scripts": {  
    "dev": "pnpm -r --parallel start"  
  }  
}
```

## 🏛️ Étape 2 — Root Config (Single-SPA Layout Engine)

```
pnpm dlx create-single-spa
```

Réponses :

Question	Réponse
Directory for project	root-config
Select type	root config
Package manager	pnpm

Question	Réponse
TypeScript	<input checked="" type="checkbox"/> Yes
Layout Engine	<input checked="" type="checkbox"/> Yes
Organization name	Rimeryx

Puis :

```
cd root-config
git init # ⚠ si Git n'était pas installé lors du wizard
pnpm install
pnpm start
```

👉 Le shell tourne sur :

➡ <http://localhost:9000/>

## 🛠 Étape 3 — Micro-frontend React

Depuis la racine du playground :

```
pnpm dlx create-single-spa
```

Réponses :

Question	Réponse
Directory for new project	mfe-react
Type	application/parcels
Framework	React
Package manager	pnpm
TypeScript	<input checked="" type="checkbox"/> Yes
Organization name	Rimeryx

```
cd mfe-react
pnpm install
pnpm start
```

⚠ Ne ferme **aucune fenêtre** webpack pendant la suite.

## 🌐 Étape 4 — Micro-frontend Vanilla JS + jQuery

```
mkdir mfe-vanilla  
cd mfe-vanilla  
pnpm init -y  
pnpm add jquery
```

Créer :

📌 public/site.html

```
<div class="vanilla-app">  
  <h3>Vanilla + jQuery ✓</h3>  
  <p id="txt">Hello depuis Vanilla !</p>  
  <button id="btn">Changer texte</button>  
</div>
```

📌 src/styles.css

```
.vanilla-app {  
  padding: 16px;  
  background: #dff9fb;  
  border-left: 4px solid #22a6b3;  
}
```

📌 src/site.js

```
import $ from "jquery";  
import "./styles.css";  
  
export function mount() {  
  $("#btn").on("click", () => $("#txt").text("Texte modifié ✓"));  
}  
  
export function unmount() {  
  $("#btn").off();  
}
```

📌 webpack.config.js (UMD output)

```

const HtmlWebpackPlugin = require("html-webpack-plugin");
const { merge } = require("webpack-merge");

module.exports = merge({
  entry: "./src/site.js",
  devServer: {
    port: 8082,
    static: "./public"
  },
  output: {
    library: "Rimeryx-mfe-vanilla",
    libraryTarget: "umd"
  },
  plugins: [
    new HtmlWebpackPlugin({
      template: "./public/site.html"
    })
  ],
});

```

 Déclarer dans **root-config** (import-map local)

```

"imports": {
  "@Rimeryx/mfe-vanilla": "//localhost:8082/Rimeryx-mfe-vanilla.js"
}

```

## Étape 5 — Micro-frontend Vue 3

```
pnpm dlx create-single-spa
```

Réponses :

Question	Réponse
Directory	mfe-vue
Framework	Vue
Package Manager	pnpm

Question	Réponse
TypeScript	✗ No

Puis :

```
cd mfe-vue  
pnpm install  
pnpm start
```

✓ Ajouter dans `pnpm-workspace.yaml` :

```
- "mfe-vue"
```

✓ Déclarer dans import-map

```
"@Rimeryx/mfe-vue": "//localhost:8081/Rimeryx-mfe-vue.js"
```

→ App Vue fonctionnelle dans /vue ✓

## A Étape 6 — Micro-frontend Angular

Global (si manquant) :

```
pnpm add -g @angular/cli
```

Créer projet :

```
cd microfrontends-playground  
ng new mfe-angular --routing=false --style=css --package-manager=pnp  
m --skip-git
```

Configurer single-spa :

```
cd mfe-angular  
ng add single-spa-angular
```

Réponses :

Question	Réponse
Use Angular Routing ?	No

Modifier `angular.json` (éviter hash filenames) :

```
"outputHashing": "none"
```

## 💡 Étape 7 — Root-Config Layout update

Dans `root-config/src/microfrontend-layout.html`

```
<single-spa-router>
  <main>
    <route path="react">
      <application name="@Rimeryx/mfe-react"></application>
    </route>
    <route path="vanilla">
      <application name="@Rimeryx/mfe-vanilla"></application>
    </route>
    <route path="vue">
      <application name="@Rimeryx/mfe-vue"></application>
    </route>
    <route path="angular">
      <application name="@Rimeryx/mfe-angular"></application>
    </route>
  </main>
</single-spa-router>
```

## 🚀 Étape 8 — Démarrage global

Dans la racine :

```
pnpm dev
```

Tous les micro-fronts démarrent automatiquement ✅

- 🟦 Angular → <http://localhost:4200>
- 🟩 Vue → <http://localhost:8081>

-  React → <http://localhost:8080>
-  Vanilla → <http://localhost:8082>
-  Shell root-config → <http://localhost:9000>

➡️ Navigation via routes :

```
/react
/vue
/angular
/vanilla
```

## Playground 100% fini !

Tu as désormais :

Micro-frontend	Framework	Port
root-config	Single SPA Shell	9000
mfe-react	React 18	8080
mfe-vue	Vue 3	8081
mfe-vanilla	HTML/CSS/JS/jQuery	8082
mfe-angular	Angular 17	4200

 Monorepo

 Chargement dynamique

 Déploiables séparément

 Navigation centralisée

 Un seul `pnpm dev`

## Bonus — Idées de mini-projets à jouer dans ton Playground

Projet	Framework
To-do list	React
Calculatrice UI	Vue
Dashboard RH (faux)	Angular

Projet	Framework
UI Effets jQuery	Vanilla
Chat temps-réel	React + Vue
Module Panier	Angular
Menu global partagé	root-config