Jak to działa *npm*

1. Utwórz główny katalog projektu:

   ```
   mkdir mój-projekt
   ```

2. Przejdź do katalogu:

   ```
   cd mój-projekt
   ```

3. Zainicjuj projekt główny:

   ```
   npm init -y
   ```

   (Tworzy plik package.json).

4. Zdefiniuj workspace'y w package.json projektu głównego:

   ```
   {
     "name": "mój-projekt",
     "version": "1.0.0",
     "private": true,  // Ważne: NIE do publikacji w npm
     "workspaces": [
       "workspace-1",
       "workspace-2",
       "workspace-3"
     ],
     "devDependencies": {  // Zainstalowane w projekcie głównym
        "svelte": "^4.0.0",
        "@sveltejs/kit": "^2.0.0"
     }
     // ... inne
   }
   ```

   workspaces: Tablica z nazwami katalogów, które są workspace'ami.

   private: *true* : Ważne! Zaznacza projekt jako prywatny.

   devDependencies: Możesz zainstalować pakiety dostępne we wszystkich workspace'ach (np. *Svelte*, @*sveltejs/kit*).

5. Utwórz katalogi workspace'ów:

   ```
   mkdir workspace-1 workspace-2 workspace-3
   ```

6. Zainicjuj *package.json* dla każdego workspace'a: Przejdź do każdego katalogu workspace'a i uruchom

   ```
   npm init -y
   ```

   Zdecyduj, które workspace'y mają być projektami *SvelteKit* (*SSR*, *SSG*, *SPA*)

7. Zainstaluj zależności w workspace'ach: Przejdź do każdego katalogu workspace'a. Zainstaluj zależności specyficzne dla każdego workspace'a.

   - Np.

     ```
     cd workspace-1 && npm install @sveltejs/kit
     ```

     (workspace-1 to *SvelteKit*)
   - Np.

     ```
     cd workspace-2 && npm install
     ```

     ... (workspace-2 może być innym typem projektu, biblioteką, itp.)

8. Używanie poleceń *npm*: Teraz możesz uruchamiać polecenia npm w workspace'ach:

   npm install Instaluje zależności we wszystkich workspace'ach.

   npm run $<$ *polecenie* $>$ Uruchamia polecenie zdefiniowane w *package.json* bieżącego katalogu.

   npm run build (jeśli zdefiniowane w package.json workspace'a SvelteKit)

9. Ważne: Symlinki: *npm* automatycznie tworzy symlinki między workspace'ami, jeśli jeden workspace importuje kod z innego. To ułatwia współdzielenie kodu.

Przykład z *SvelteKit* (*SSG/SSR*):

```
mkdir sveltekit-projekt
cd sveltekit-projekt
npm init -y
```

Edytuj *package.json* projektu głównego:

```
{
  "name": "sveltekit-projekt",
  "version": "1.0.0",
  "private": true,
  "workspaces": [
    "frontend"  // Zmieniasz nazwę na bardziej opisową
  ],
  "devDependencies": {
    "@sveltejs/adapter-auto": "^4.0.0",  // Zależności SvelteKit
    "@sveltejs/adapter-static": "^4.0.0",
    "@sveltejs/kit": "^2.0.0",
    "svelte": "^4.0.0",
    "vite": "^5.0.0"
  },
  "scripts": {
    "build": "npm run build --workspace=frontend",
    "dev": "npm run dev --workspace=frontend"
  }
}
```

Zwróć uwagę na *scripts*. Możesz zdefiniować polecenia, które uruchamiają się w konkretnych workspace'ach.

```
mkdir frontend
cd frontend
```

Utwórz package.json dla frontendu

```
npm init -y
```

Zainstaluj zależności SvelteKit w frontend

```
npm install @sveltejs/kit
            @sveltejs/adapter-auto
            @sveltejs/adapter-static
```

Konfiguracja SvelteKit (ważne):

Uruchom komendę *npm create svelte@latest* . w katalogu frontend. To utworzy podstawowy projekt *SvelteKit* (z następującymi opcjami):

- "Which Svelte app template?" -> "Skeleton project" (lub coś innego, co pasuje)
- "Use TypeScript?" -> "No" (lub "Yes" jeśli wolisz)
- "Add ESLint?" -> "Yes"
- "Add Prettier?" -> "Yes"
- "Add Playwright for browser testing?" -> "No"
- "Install dependencies?" -> "Yes"

Zmień *svelte.config.js* i *vite.config.js* w folderze frontend, aby skonfigurować *SSG*:

`svelte.config.js`:

```
import adapter from '@sveltejs/adapter-static';  // Zmień adapter
import { vitePreprocess } from '@sveltejs/kit/vite';
/** @type {import('@sveltejs/kit').Config} */
const config = {
    // Consult https://kit.svelte.dev/docs/integrations#preprocessors
    // for more information about preprocessors
    preprocess: vitePreprocess(),
    kit: {
        adapter: adapter({
           fallback: 'index.html' // Dodaj fallback
        }),
        // prerender: {
        //   enabled: true  // Jeśli chcesz pre-renderować wszystko
        // }
    }
};
export default config;
```

vite.config.js:

```
import { defineConfig } from 'vite';
import { sveltekit } from '@sveltejs/kit/vite';
export default defineConfig({
    plugins: [sveltekit()],
    build: {
        //  Dodatkowe opcje konfiguracji build
        sourcemap: false
    }
});
```

Uruchom:

cd ../ // Wróć do głównego katalogu

npm run build // Buduje projekt (w workspace'ie frontend)

npm run dev // Uruchamia serwer deweloperski (w workspace'ie frontend)