Angular

- Créer un projet Angular 16 avec Angular Material & Jest
  - Création du projet Angular avec Angular Material
  - Ajouter Jest au projet Angular
  - Ajouter des configurations - VSCode
- Créer un projet Angular 19 avec Angular Material & Jest
  - Création du projet Angular avec Angular Material
  - Ajouter Jest au projet Angular
  - Ajouter des configurations - VSCode
- Trucs & Astuces
  - Icônes : Material Symbols
- Architecture de code

Créer un projet Angular 16 avec Angular Material & Jest

Création du projet Angular avec Angular Material

> Angular v16 - `^16.2.0`
> Angular Material v16 - `^16.2.5`

Projet Angular

La création du projet se fait en une seule ligne de commande :

```
ng new project-name --routing --skip-git --strict --style=scss
```

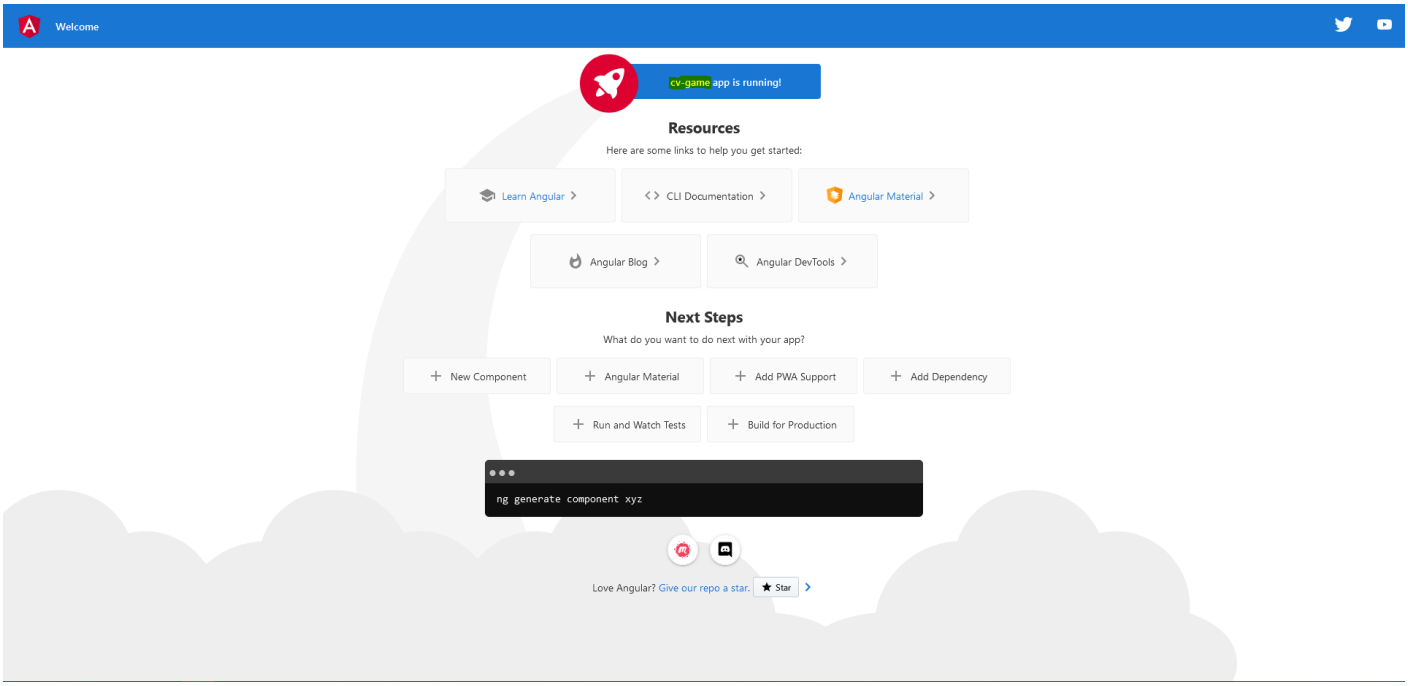
Cette ligne va créer tous les fichiers de base d'un projet Angular dans le dossier `project-name`. A partir de maintenant, tout sera fait en étant dans le dossier du projet : `project-name`.

> S'il faut créer le projet dans le dossier courant et pas dans un sous-dossier, il faut ajouter cet argument à la commande : `--directory=./`

Pour tester que tout c'est bien passé, il suffit de lancer la commande suivante puis d'ouvrir un navigateur et de taper : `localhost:4200`.

```
npm start
```

Si le résultat est celui-ci, alors l'installation basique d'Angular à fonctionné :



Ajouter Angular Material

Maintenant que le projet Angular existe, on va pouvoir ajouter Angular Material dans le projet.

Angular Material ajoute de nombreux composants pré-fabriqués pour Angular.

Toujours dans un terminal ouvert depuis le dossier du projet, la ligne suivante :

- Installe Angular Material
- Permet de choisir un thème (testables directement sur le site d'Angular Material)
- Permet d'appliquer le style `typographie` à tous le projet.
- Permet d'inclure ou non et d'activer ou non le module d'animations d'Angular

```
ng add @angular/material
```

Une fois cela fait, c'est terminé. Angular Material est maintenant utilisable dans le projet.

Ajouter Jest au projet Angular

> Angular v16 - `^16.2.0`
> Jest Preset Angular v13 - `^13.1.2`
> Jest v29 - `^29.7.0`

Pourquoi ajouter Jest ?

En effet, le projet Angular, à l'installation, contient déjà le module Karma qui permet de tester le code et de voir le résultat dans un navigateur. Ce qui, pendant la phase de développement, est plus pratique.

Cependant, dans le cas d'utilitaires comme Jenkins ou les Runners de Gitlab, le navigateur de Karma ne peut pas être lancé de manière simple. Dans ces cas là, Jest qui se lance en console est plus approprié.

Installation de Jest

Tout d'abord, il faut installer les dépendances nécessaires

```
npm install jest jest-preset-angular @types/jest --save-dev
```

Ensuite, créer un fichier `setup-jest.ts` dans un dossier `config` à la racine du projet et le remplir comme suit :

```
import 'zone.js';
import 'zone.js/testing';


import 'jest-preset-angular/setup-jest';
```

Une fois ce fichier créé, il faut encore modifier le fichier `package.json` :

- Remplacer :

```
"test": "ng test"
```

  Par :

```
"test": "jest --coverage --coverage-directory=coverage/jest",
"test:karma": "ng cache clean && ng test",
```

- Ajouter, avant le bloc `"dependencies"` :

```
"jest": {
  "preset": "jest-preset-angular",
  "setupFilesAfterEnv": [
    "./config/setup-jest.ts"
  ],
  "moduleNameMapper": {
    "^src/(.*)$": "./src/$1"
  }
},
```

Maintenant, en lançant `npm test`, les tests de Jest se lancent. Mais la commande `npm run test:karma` permet de passer les tests avec Karma et son navigateur.

Ajouter des configurations - VSCode

> Cette page va lister quelques configurations pouvant être utiles en cas d'utilisation de VSCode en tant qu'IDE (*Integrated Development Environment*)

VSCode

Il y a 3 fichiers de VSCode qui peuvent être modifiés/créés dans le dossier `.vscode` : `extensions.json`, `launch.json` et `settings.json`

extensions.json

```
{
  // For more information, visit: https://go.microsoft.com/fwlink/?linkid=827846
  "recommendations": [
    "angular.ng-template",
    "esbenp.prettier-vscode",
    "cyrilletuzi.angular-schematics",
    "formulahendry.auto-rename-tag",
    "zainchen.json",
    "davidanson.vscode-markdownlint",
    "pmneo.tsimporter",
    "orta.vscode-jest",
    "ryanluker.vscode-coverage-gutters",
    "pkief.material-icon-theme"
  ]
}
```

launch.json

```
{
  // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
  "version": "0.2.0",
```

```
  "configurations": [
    {
      "name": "ng serve",
      "type": "chrome",
      "request": "launch",
      "preLaunchTask": "npm: start",
      "url": "http://localhost:4200/"
    },
    {
      "name": "ng test",
      "type": "vscode-jest-tests",
      "request": "launch",
      "preLaunchTask": "npm: test"
    }
  ]
}
```

settings.json

```
{
  "javascript.preferences.importModuleSpecifier": "relative",
  "typescript.preferences.importModuleSpecifierEnding": "minimal",
  "typescript.preferences.importModuleSpecifier": "relative",
  "javascript.preferences.importModuleSpecifierEnding": "minimal",
  "js/ts.implicitProjectConfig.module": "ES2022",
  "editor.formatOnSave": true,
  "editor.codeActionsOnSave": {
    "source.organizeImports": "always"
  },
  "files.associations": {
    "*.prettierrc": "json"
  },
  "jest.useDashedArgs": true,
  "jest.coverageFormatter": "GutterFormatter",
  "jest.runMode": "on-demand",
  "coverage-gutters.showLineCoverage": true,
  "coverage-gutters.showGutterCoverage": false,
  "coverage-gutters.showRulerCoverage": true
}
```

Node

package.json

Il faut installer :

- `npm i @types/node`
- `npm i -D ng-mocks webpack-bundle-analyzer`

Puis il faut ajouter, dans les scripts ;

```
"ng": "ng",
"start": "ng serve",
"build": "ng build",
"build:prod": "ng build --configuration production",
"watch": "npm run build -- --watch --configuration development",
"test": "ng cache clean && ng test",
"build:stats": "npm run build -- --stats-json",
"build:stats:prod": "npm run build:prod -- --stats-json",
"analyze:bundle": "webpack-bundle-analyzer dist/project-name/stats.json",
"analyze": "npm run build:stats && npm run analyze:bundle",
"analyze:prod": "npm run build:stats:prod && npm run analyze:bundle"
```

Si jest est déjà installé, il faut remplacer la ligne

```
"test": "ng cache clean && ng test",
```

par les lignes

```
"test": "jest --coverage --coverage-directory=coverage/jest",
"test:karma": "ng cache clean && ng test",
```

TypeScript Config

Dans le fichier `tsconfig.json`, il est possible et recommandé d'ajouter cette ligne dans la catégorie `compilerOptions`

```
"esModuleInterop": true
```

Prettier

Il faut, pour le configurer, placer un fichier `.prettierrc` à la racine du projet. Voilà un exemple de ce qu'il pourrait contenir ;

```
{
  "singleQuote": true,
  "arrowParens": "avoid",
  "proseWrap": "always",
  "htmlWhitespaceSensitivity": "ignore",
  "bracketSameLine": true,
  "printWidth": 120,
  "useTabs": false,
  "jsxSingleQuote": true,
  "trailingComma": "none",

  "singleAttributePerLine": false,
  "semi": true,
  "tabWidth": 2,
  "bracketSpacing": true,
  "embeddedLanguageFormatting": "auto",
  "endOfLine": "lf",
  "insertPragma": false,
  "quoteProps": "as-needed",
  "requirePragma": false,
  "vueIndentScriptAndStyle": false
}
```

Créer un projet Angular 19 avec Angular Material & Jest

Création du projet Angular avec Angular Material

> Angular v19 - `^19.0.0`
> Angular Material v19 - `^19.1.1`

Projet Angular

La création du projet se fait en une seule ligne de commande :

```
ng new project-name --routing --skip-git --strict --style=scss
```

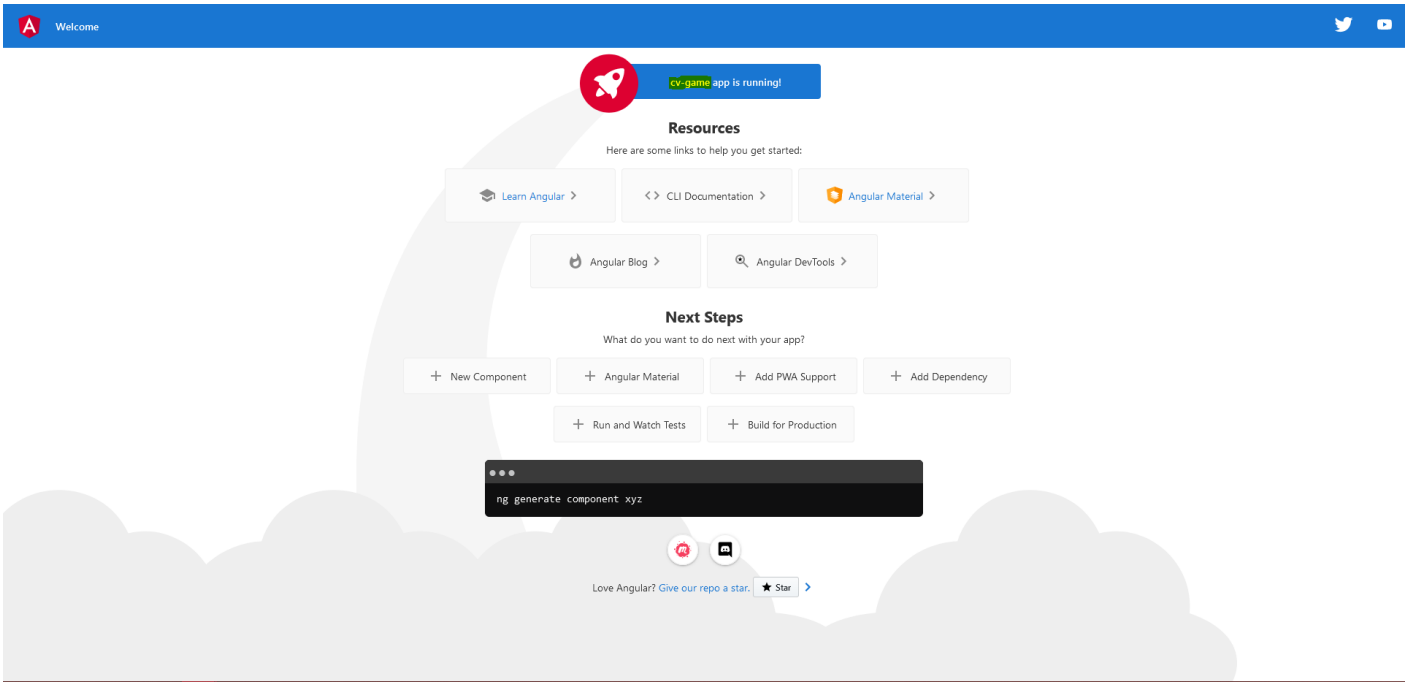
Cette ligne va créer tous les fichiers de base d'un projet Angular dans le dossier `project-name`. A partir de maintenant, tout sera fait en étant dans le dossier du projet : `project-name`.

> S'il faut créer le projet dans le dossier courant et pas dans un sous-dossier, il faut ajouter cet argument à la commande : `--directory=./`

Pour tester que tout c'est bien passé, il suffit de lancer la commande suivante puis d'ouvrir un navigateur et de taper : `localhost:4200`.

```
npm start
```

Si le résultat est celui-ci, alors l'installation basique d'Angular à fonctionné :



Ajouter Angular Material

Maintenant que le projet Angular existe, on va pouvoir ajouter Angular Material dans le projet.

Angular Material ajoute de nombreux composants pré-fabriqués pour Angular.

Toujours dans un terminal ouvert depuis le dossier du projet, la ligne suivante :

- Installe Angular Material
- Permet de choisir un thème (testables directement sur le site d'Angular Material)
- Permet d'appliquer le style `typographie` à tous le projet.
- Permet d'inclure ou non et d'activer ou non le module d'animations d'Angular

```
ng add @angular/material
```

Une fois cela fait, c'est terminé. Angular Material est maintenant utilisable dans le projet.

Ajouter Jest au projet Angular

> Angular v19 - `^19.0.0`
> Jest Preset Angular v14 - `^14.5.1`
> Jest v29 - `^29.7.0`

Pourquoi ajouter Jest ?

En effet, le projet Angular, à l'installation, contient déjà le module Karma qui permet de tester le code et de voir le résultat dans un navigateur. Ce qui, pendant la phase de développement, est plus pratique.

Cependant, dans le cas d'utilitaires comme Jenkins ou les Runners de Gitlab, le navigateur de Karma ne peut pas être lancé de manière simple. Dans ces cas là, Jest qui se lance en console est plus approprié.

Installation de Jest

Tout d'abord, il faut installer les dépendances nécessaires

```
npm install jest jest-preset-angular @types/jest --save-dev
```

Ensuite, créer un fichier `setup-jest.ts` dans un dossier `config` à la racine du projet et le remplir comme suit :

```
import 'zone.js';
import 'zone.js/testing';

import { setupZoneTestEnv } from 'jest-preset-angular/setup-env/zone';

setupZoneTestEnv();
```

Une fois ce fichier créé, il faut encore modifier le fichier `package.json` :

- Remplacer :

  ```
  "test": "ng test"
  ```

  Par :

  ```
  "test": "jest --verbose",
  "test:watch": "npm test -- --watch",
  "test:cov": "npm test -- --coverage --coverageReporters=lcov --coverageReporters=text --
  coverageReporters=text-summary --coverageReporters=html",
  "test:ci": "jest --ci",
  "test:karma": "ng cache clean && ng test",
  ```

- Ajouter, avant le bloc `"dependencies"` :

  ```
  "jest": {
    "preset": "jest-preset-angular",
    "setupFilesAfterEnv": [
      "./config/setup-jest.ts"
    ],
    "moduleNameMapper": {
      "^src/(.*)$": "./src/$1"
    }
  },
  ```

Maintenant, en lançant `npm test`, les tests de Jest se lancent. Mais la commande `npm run test:karma` permet de passer les tests avec Karma et son navigateur.

Ajouter des configurations - VSCode

> Cette page va lister quelques configurations pouvant être utiles en cas d'utilisation de VSCode en tant qu'IDE (*Integrated Development Environment*)

VSCode

Il y a 3 fichiers de VSCode qui peuvent être modifiés/créés dans le dossier `.vscode` : `extensions.json`, `launch.json` et `settings.json`

extensions.json

```
{
  // For more information, visit: https://go.microsoft.com/fwlink/?linkid=827846
  "recommendations": [
    "angular.ng-template",
    "esbenp.prettier-vscode",
    "cyrilletuzi.angular-schematics",
    "formulahendry.auto-rename-tag",
    "zainchen.json",
    "davidanson.vscode-markdownlint",
    "pmneo.tsimporter",
    "orta.vscode-jest",
    "ryanluker.vscode-coverage-gutters",
    "pkief.material-icon-theme"
  ]
}
```

launch.json

```
{
  // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
  "version": "0.2.0",
```

```
  "configurations": [
    {
      "name": "ng serve",
      "type": "chrome",
      "request": "launch",
      "preLaunchTask": "npm: start",
      "url": "http://localhost:4200/"
    },
    {
      "name": "ng test",
      "type": "vscode-jest-tests",
      "request": "launch",
      "preLaunchTask": "npm: test"
    }
  ]
}
```

settings.json

```
{
  "javascript.preferences.importModuleSpecifier": "relative",
  "typescript.preferences.importModuleSpecifierEnding": "minimal",
  "typescript.preferences.importModuleSpecifier": "relative",
  "javascript.preferences.importModuleSpecifierEnding": "minimal",
  "js/ts.implicitProjectConfig.module": "ES2022",
  "editor.formatOnSave": true,
  "editor.codeActionsOnSave": {
    "source.organizeImports": "always"
  },
  "files.associations": {
    "*.prettierrc": "json"
  },
  "jest.useDashedArgs": true,
  "jest.coverageFormatter": "GutterFormatter",
  "jest.runMode": "on-demand",
  "coverage-gutters.showLineCoverage": true,
  "coverage-gutters.showGutterCoverage": false,
  "coverage-gutters.showRulerCoverage": true
}
```

Node

package.json

Il faut installer :

- `npm i @types/node`
- `npm i -D ng-mocks webpack-bundle-analyzer prettier`
- `ng add @angular-eslint/schematics`

Puis il faut ajouter, dans les scripts ;

```
"format": "prettier --write \"src/**/*.ts\"",
"ng": "ng",
"start": "ng serve",
"build": "ng build",
"build:prod": "ng build --configuration production",
"watch": "npm run build -- --watch --configuration development",
"test": "ng cache clean && ng test",
"build:stats": "npm run build -- --stats-json",
"build:stats:prod": "npm run build:prod -- --stats-json",
"analyze:bundle": "webpack-bundle-analyzer dist/project-name/stats.json",
"analyze": "npm run build:stats && npm run analyze:bundle",
"analyze:prod": "npm run build:stats:prod && npm run analyze:bundle",
"lint": "ng lint",
"lint:txt": "ng lint > lint-report.txt --silent",
"lint:html": "ng lint > lint-report.html --format html --silent",
"lint:json": "ng lint > lint-report.json --format json --silent"
```

Si jest est déjà installé, il faut remplacer la ligne

```
"test": "ng cache clean && ng test",
```

par les lignes

```
"test": "jest --verbose",
"test:watch": "npm test -- --watch",
"test:cov": "npm test -- --coverage --coverageReporters=lcov --coverageReporters=text --
coverageReporters=text-summary --coverageReporters=html",
"test:ci": "jest --ci",
"test:karma": "ng cache clean && ng test",
```

Prettier

Il faut, pour le configurer, placer un fichier `.prettierrc` à la racine du projet. Voilà un exemple de ce qu'il pourrait contenir ;

```
{
  "singleQuote": true,
  "arrowParens": "avoid",
  "proseWrap": "always",
  "htmlWhitespaceSensitivity": "ignore",
  "bracketSameLine": true,
  "printWidth": 120,
  "useTabs": false,
  "jsxSingleQuote": true,
  "trailingComma": "all",

  "singleAttributePerLine": false,
  "semi": true,
  "tabWidth": 2,
  "bracketSpacing": true,
  "embeddedLanguageFormatting": "auto",
  "endOfLine": "lf",
  "insertPragma": false,
  "quoteProps": "as-needed",
  "requirePragma": false,
  "vueIndentScriptAndStyle": false
}
```

ESLint

Il faut ajouter, dans le fichier `eslint.config.js`, la règle suivante ;

```
  '@typescript-eslint/no-unused-vars': ['error', { argsIgnorePattern: '^_', varsIgnorePattern: '^_' }],
```

GitIgnore

Il faut ajouter également dans le fichier `.gitignore` le pattern suivant : `/lint-*.*`

Trucs & Astuces

Trucs & Astuces

Icônes : Material Symbols

Etapes pour utiliser les Material Symbols

1. Ajouter la police Material Symbols

Ajouter dans le `<head>` du `index.html` :

```
<link
  rel="stylesheet"

href="https://fonts.googleapis.com/css2?family=Material+Symbols+Outlined:opsz,wght,FILL,GRAD@20..48,100..700,0..1,-50..200" />
```

Il est aussi possible d'utiliser `Rounded`, `Sharp` ou `Filled` selon le style voulu.

Exemples :

- `Material+Symbols+Rounded`
- `Material+Symbols+Sharp`
- `Material+Symbols+Outlined`
- `Material+Symbols+Filled`

Configurer l'`app` pour utiliser ces fonts

Dans le fichier `app.config.ts` (Angular 15+) :

```
import { MatIconRegistry } from '@angular/material/icon';
import { DomSanitizer } from '@angular/platform-browser';

export const appConfig: ApplicationConfig = {
  providers: [
```

```
    ...,
    {
      provide: 'init-material-symbols-fontSet',
      useFactory: (iconRegistry: MatIconRegistry, _sanitizer: DomSanitizer) => {
        return () => {
          iconRegistry.registerFontClassAlias('material-symbols-outlined');
          // Possible également d'enregistrer d'autres variantes ici :
          // iconRegistry.registerFontClassAlias('material-symbols-rounded');
        };
      },
      deps: [MatIconRegistry, DomSanitizer],
      multi: true
    }
  ],
};
```

Utiliser les Material Symbols

```
<mat-icon fontSet="material-symbols-outlined">home</mat-icon>
```

Architecture de code