TP TypeScript

Nous verrons au cours des travaux pratiques les points suivants :

- TP 1 – Mise en place de l'environnement,

Objectifs

TP 1 - Installation de l'environnement

Prérequis

Vérifier que vous avez les éléments suivants d'installé sur votre poste :

- Node v6 ou plus avec la commande npm –v,
- Git, nous l'utiliserons pour récupérer le projet initial,
- Webstorm (ou un autre IDE)

Installation

Initialisation du projet

Nous allons utiliser l'outil `@angular/cli` pour générer le projet. Placez-vous dans votre workspace et lancez les commandes suivantes :

```
npm install -g @angular/cli
ng new tp-angular
cd tp-angular
npm install --save @angular/material
```

> Une fois le projet créé vous pouvez l'ouvrir dans votre IDE.

Pour lancer le serveur de développement :

```
ng serve
```

Installation d'Angular Material

Nous allons utiliser Angular material pour créer les interfaces de nos applications.

Lancer la commande suivante:

```
npm install --save @angular/material
```

Puis rajouter dans le fichier `app.module.ts` la dépendence du module comme suivant :

```
import { MaterialModule } from '@angular/material';
// other imports
@NgModule({
  imports: [MaterialModule],
  ...
})
export class AppModule { }
```

En complément vous pouvez rajouter la feuille de style suivante dans la page index si vous voulez utiliser les icônes de la librairie Material :

```
<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="styl
```

Générer des Composants, Directives, Pipe, etc...

Vous pouvez utiliser la commande `ng generate` (ou `ng g`) pour générer des composants Angular :

```
ng generate component my-new-component
ng g component my-new-component # using the alias

# components support relative path generation
# if in the directory src/app/feature/ and you run
ng g component new-cmp
# your component will be generated in src/app/feature/new-cmp
# but if you were to run
ng g component ../newer-cmp
# your component will be generated in src/app/newer-cmp
```

Voici la liste des commandes possibles pour générer une fonctionnalité d'Angular:

Génération	Usage
Component	`ng g component my-new-component`
Directive	`ng g directive my-new-directive`
Pipe	`ng g pipe my-new-pipe`
Service	`ng g service my-new-service`
Class	`ng g class my-new-class`
Guard	`ng g guard my-new-guard`
Interface	`ng g interface my-new-interface`
Enum	`ng g enum my-new-enum`
Module	`ng g module my-module`

TP 2 - Le composant Chronomètre

Nous allons dans ce TP créer un composant Chronomètre. Nous verrons ainsi comment créer un composant ainsi que la création d'un Pipe pour formater l'affiche du composant.

Notre premier composant

Dans le dossier `src/app`, créer un dossier `chrono`. Ce dossier contiendra les fichiers `html`, `css` et `ts`.

Voici la sctructure cible du projet :

```
src/app
├── app.component.css
├── app.component.html
├── app.component.spec.ts
├── app.component.ts
├── app.module.ts
└── chrono
    ├── chrono.component.css
    ├── chrono.component.html
    └── chrono.component.ts
```

Vous devez donc créer les fichiers nécessaires au componsant Chrono !

Ensuite