

Compte rendu du tp pokeapi

Tout d'abord j'ai installé nodeJs en suivant le tutoriel, puis installé ng cli. Une fois fait, j'ai initialisé le projet puis lancer le serveur sur localhost:4200 grâce à la commande `ng serve` dans le terminal. Au tout début, la page générée est celle qui est dans `app-component.html`.

Q1-3 :

Ensuite j'ai créé un composant "my-component" en utilisant la commande `ng generate component my-component`. Dans `my-component.component.html` j'ai rajouté une fonctionnalité de *databinding* : `<input [(ngModel)]="id">` mais pour que cela fonctionne il faut aussi ajouter l'import : `import { FormsModule } from '@angular/forms'`, ainsi que l'ajout de **FormsModule** dans imports : [...] dans `app.module.ts` et dans `my-component.component.ts`, il faut rajouter "`id! : string;`" où "`!`" signifie qu'`id`, de type `string`, n'est pas nulle ou alternativement on pouvait aussi faire comme suit : `id = ""`; À chaque fois qu'on fait du databinding dans le .html du composant il faut le rajouter l'élément utilisé dans le fichier `.component.ts`.

Et enfin afin de pouvoir charger le contenu du fichier `my-component.component.html`, il faut rajouter dans `app.component.html` "`<app-my-component></app-my-component>`".

Q3bis :

Il est difficile de faire une attaque XSS sur angular parce qu'angular va protéger tout affichage de type en html.

Q4 - 7 :

`ng g class pokemon` permet de créer la classe `pokemon`. Une fois créée, dans `pokemon.ts` -> `constructor()`, j'ai ajouté des attributs : 'id' et 'name' puis dans `my-component.component.ts` j'ai ajouté un tableau de pokémon, ensuite dans le `constructor(){ ... }` j'ai rajouté quelques pokémons dans ce tableau. Pour l'affichage des listes (déroulante) des pokémons, cela se passe dans `my-component.component.html` en utilisant la balise `<select *ngFor = " ... "> {{ ... }}</select>` (comme évoqué dans le tuto https://angular.io/guide/template-syntax#ngforof).

Q8 :

Pour faciliter le filtrage des données, on a utilisé "pipe" qui est généré grâce à la commande `ng g pipe filter-pokemon--pipe`. Dans la fonction `transform`, `value` correspond au donné en entrée, `property?` La propriété sur laquelle elle doit filtrer et enfin le dernier argument qui correspond à ce qu'on va y recherché.

