

16/04/2020

# Pokemon

Projet Angular



Présenté par : Drevet Mélanie et Richioud Amélie  
S4B1



# Sommaire

1

**Objectif général**

2

**Le fonctionnement technique et les fonctionnalités**

3

**Les maquettes**



# Objectif général

---

## Introduction

L'objectif de notre application repose sur le thème des pokémons. Cette application fonctionne via une API qui réunit 250 cartes Pokémons et qui les affichent. Le site va permettre d'afficher les cartes selon les types des pokémons, ainsi que d'effectuer des recherches pour trouver le pokémon souhaité. Les cartes pokémons étant en anglais nous avons décidé de faire l'entièreté du site en anglais pour donner une cohérence à notre site.

## Problématique

La problématique qui est survenue lors de la mise en place de notre site était d'arriver à réaliser un site autour de cartes pokémon. Arriver à les mettre en valeur tout en tenant compte du fait qu'elles étaient en anglais.

## Pourquoi l'utiliser

Mais du coup, pourquoi utiliser notre site ? Notre site s'adresse à tous les passionnés de carte Pokémon. Si vous cherchez des informations sur d'ancienne ou nouvelle carte, que vous soyez nostalgique de votre enfance ou simplement curieux, sur notre site vous trouverez plus de 250 cartes avec leurs informations. Via une simple recherche grâce au nom ou au type vous pourrez retrouver les cartes que vous aviez ou trouver les cartes que vous aimeriez avoir.

# Le fonctionnement technique et les fonctionnalités

---



## L'API :

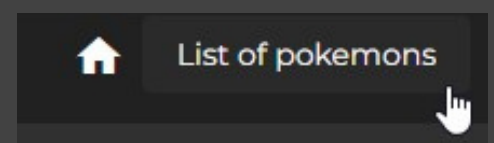
Nous utilisons l'API Pokemon TCG, cette api a pour objectif de nous fournir une base de données regroupant toutes les cartes pokémons. Elle fournit au sein de ces cartes toutes les caractéristiques nécessaires comme le nom du pokémon, son type ou encore ses attaques.

Cette API offre différentes fonctionnalités. Elle nous permet d'obtenir toutes les cartes, d'obtenir une carte en particulier, d'obtenir les types ainsi que les sous-types mais aussi d'autres fonctionnalités utiles. Pour notre site nous avons décidé de ne pas utiliser toutes les fonctionnalités de l'API pour nous concentrer que sur celles qui nous intéressent le plus. Nous avons donc choisi d'appliquer le fait de pouvoir lister toutes les cartes, d'en obtenir une seule en particulier et de pouvoir les trier selon le nom complet et le type du pokémon.

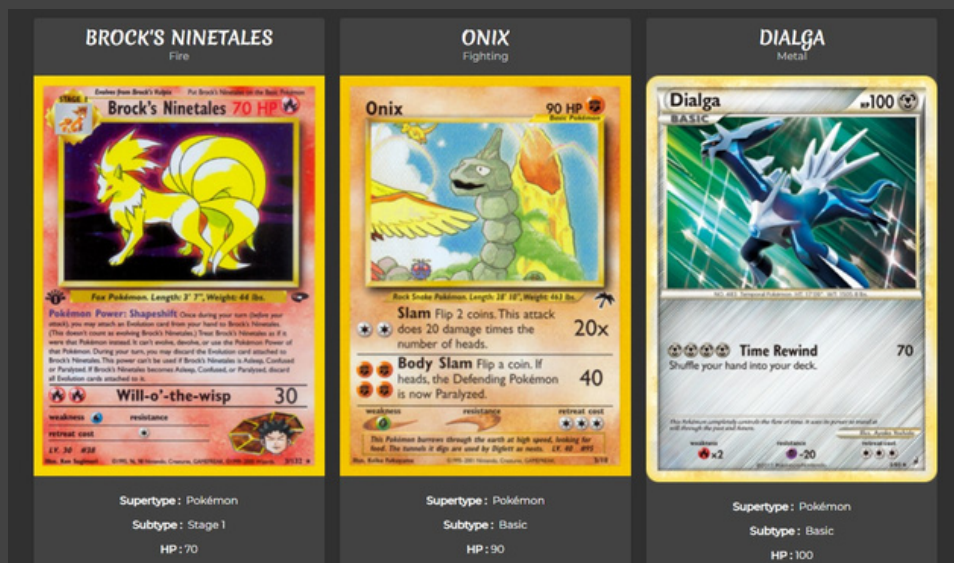
## La librairie externe - Angular Material

Nous avons décidé de travailler avec la librairie externe Angular Material. Nous avons fait ce choix pour apprendre à connaître et à exploiter cette librairie qui nous était inconnue auparavant. De plus, elle nous permet de réaliser un design original qui correspond bien à ce que nous voulions pour notre site pokemon. En utilisant Angular Material nous nous sommes servi de pas mal de composants qu'elle offre.

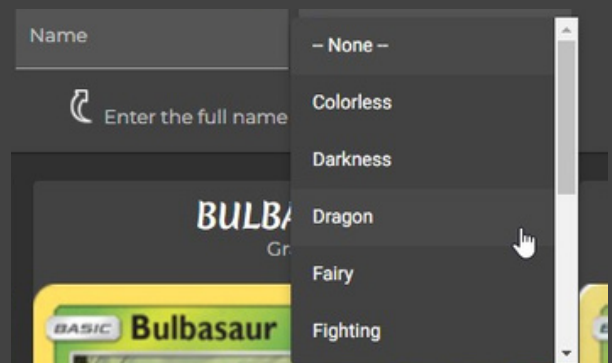
Le premier composant que nous avons utilisé est le bouton. Il permet d'apporter une interaction avec le site dans un design sobre et épuré.



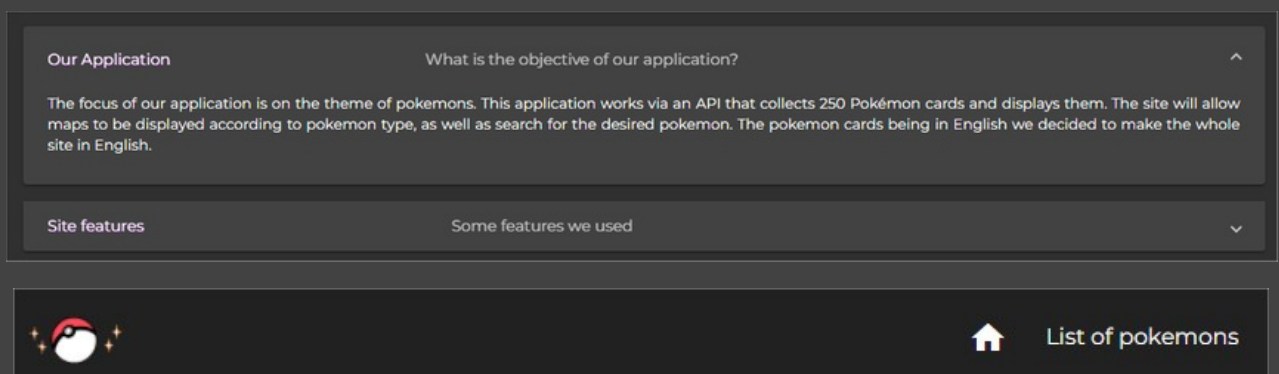
Le deuxième composant est la card. Elle nous permet de regrouper les informations d'une même carte pokémon dans un bloc. Elle facilite alors la lisibilité en ajoutant des séparation visible entre plusieurs cards. ✨🍷



Le troisième composant est le form field. Ce composant est utilisé pour saisir du texte ici le nom de votre pokémon ainsi que de pouvoir sélectionner via une liste un type pour le pokémon recherché.

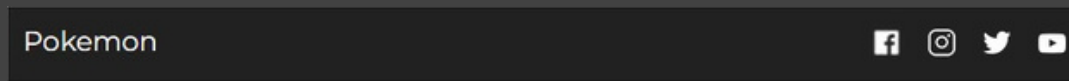


Le quatrième composant que nous avons choisi d'utiliser est l'expansion panel. Il nous permet de faire dérouler lors d'un clic des informations supplémentaires. L'utilité de ce composant ici est d'éviter de charger notre site avec des informations qui n'intéressent pas tous les visiteurs.

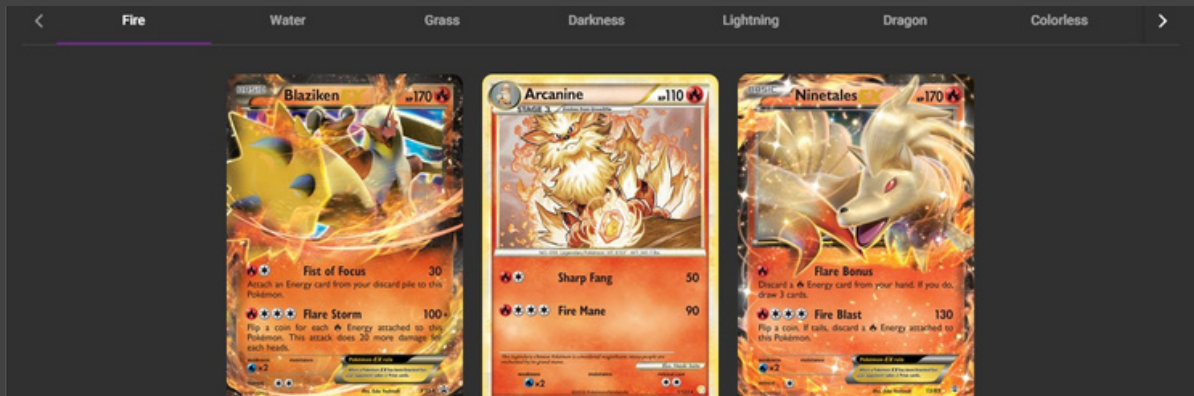


Le cinquième composant est utilisé 2 fois dans notre cas. Il s'agit de la toolbar qui nous permet de réaliser un menu fixe en haut du site. Ce menu est composé de 2 boutons ainsi que d'un petit logo animé.

Le composant toolbar est également utilisé pour réaliser notre footer en bas du site. Il contient les différents réseaux officiels de Pokémon ainsi que le nom du site soit Pokémon.



Enfin le dernier composant est le tabs. Il permet d'ajouter des onglets sur notre site pour afficher des exemples de cartes pokémons qu'on peut trouver. Ce composant nous permet donc d'illustrer notre page d'accueil dans un design plus attractif et original qu'un simple carrousel.



## Le filtre avec explication du code :

### La partie html :

Tout d'abord nous utilisons une card dans laquelle nous ajoutons une balise form. La balise form joue le rôle de formulaire qui interagit avec l'utilisateur. Le header de notre card contient une balise mat-card-title qui affiche le nom de la card. Au sein de la balise div que nous créons nous avons 2 parties.

```
<mat-form-field class="search-bar-field" appearance="fill">
  <mat-label>Name</mat-label>
  <input matInput type="text" FormControlName="name"/>
</mat-form-field>
```

La première partie correspond à une barre de recherche de type text où l'utilisateur pourra taper le nom du Pokémon. Nous avons une balise mat-form-field qui permet de faire un champ de formulaire. À l'intérieur nous ajoutons une balise mat-label qui permettra de faire afficher le nom du champ. Nous retrouvons également un input qui crée le champ de saisie et avec l'attribut FormControlName il permet de faire afficher le mot "Name" lorsque le champ n'est pas utilisé. Cela sera la valeur par défaut.





```
<mat-form-field class="search-bar-field" appearance="fill">
  <mat-select formControlName="types">
    <mat-option>-- None --</mat-option>
    <mat-option value="colorless">Colorless</mat-option>
    <mat-option value="darkness">Darkness</mat-option>
    <mat-option value="dragon">Dragon</mat-option>
    <mat-option value="fairy">Fairy</mat-option>
    <mat-option value="fighting">Fighting</mat-option>
    <mat-option value="fire">Fire</mat-option>
    <mat-option value="grass">Grass</mat-option>
    <mat-option value="lightning">Lightning</mat-option>
    <mat-option value="metal">Metal</mat-option>
    <mat-option value="psychic">Psychic</mat-option>
    <mat-option value="water">Water</mat-option>
  </mat-select>
  <mat-label>Types</mat-label>
</mat-form-field>
```

La deuxième partie correspond à un select grâce à la balise mat-select. Ici nous pouvons choisir une option parmi une liste définie grâce aux balises mat-option. Nous allons alors pouvoir choisir le type du pokémon que nous voulons.

Enfin nous ajoutons dans la balise mat-card-subtitle une image de flèche avec une petite phrase qui donne une petite indication sur comment remplir le champ name pour le faire bien fonctionner.

```
<mat-card-subtitle>
  
  Enter the full name of the desired pokemon
</mat-card-subtitle>
```

## La partie typescript :

```
export interface SearchParams {
  name: string;
  types:string;
}
```

Nous créons tout d'abord une interface avec les deux éléments de notre filtre, à savoir name et types.

Nous créons ensuite un Output, qui est un décorateur qui permet de marquer un champ comme "propriété de sortie". Cela permet de faire une mise à jour automatique lors de la détection de modifications. Cet Output est associé à un observable de type "EventEmitter", classe qui permet d'émettre des valeurs. Dans notre cas, cette classe émet les valeurs contenues dans l'interface SearchParams. Nous déclarons aussi une variable searchForm de type FormGroup, qui permet de récupérer les valeurs name et types du formulaire et de les lier à ce fichier ts.

```
@Output()
search = new EventEmitter<SearchParams>();

searchForm = new FormGroup({
  name: new FormControl(''),
  types: new FormControl(''),
});
```

Enfin, on utilise la fonction `ngOnInit`, appelée quand les composants sont affichés et initiés. Cette fonction prend les valeurs entrées dans le formulaire et appelle l'Output pour lancer la recherche et trier les Pokémons.



```
ngOnInit(): void {  
  this.searchForm.valueChanges.subscribe((value) => {  
    this.search.emit(value);  
  });  
}
```

## La navigation :

La navigation du site est composée de 3 pages principales. La première est la page d'accueil. Elle affiche tout d'abord le logo en gros de Pokémon, le thème sur lequel repose notre site. Elle affiche aussi 3 exemples de carte pour chaque type que l'utilisateur peut retrouver sur notre site. Elle est également composée de deux panneaux d'extensions qui affichent respectivement l'objectif de notre application ainsi que les fonctionnalités qu'on peut retrouver.

Il faut savoir que toutes nos pages sont composées d'un menu fixe qui permet de naviguer entre la page d'accueil et la liste des pokémons ainsi que d'un footer regroupant les réseaux sociaux de Pokémon.

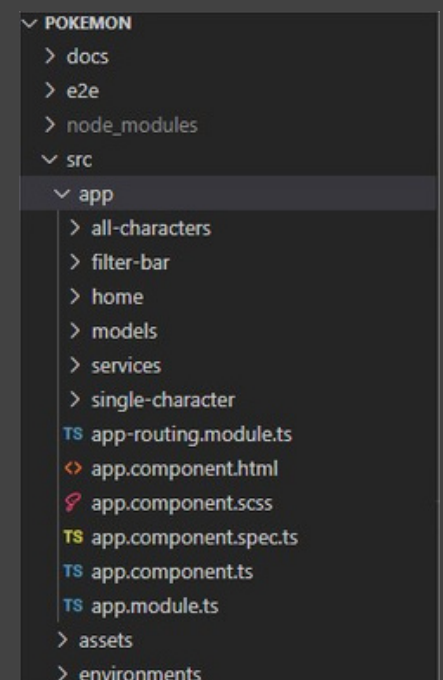
La deuxième page est celle qui regroupe tous les pokémons. Elle est aussi munie d'un filtre qui trie les pokémons selon leurs noms ou leur type.

La troisième est une page spécifique à chaque pokémon, elle affiche la carte du pokémon regroupant toutes ses caractéristiques.

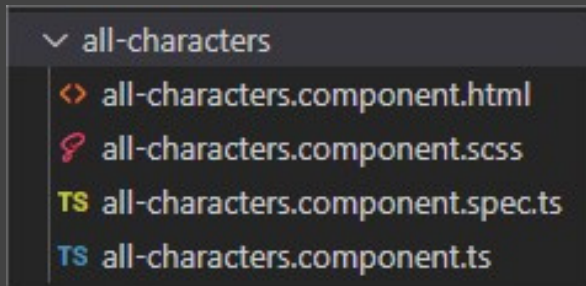
## L'architecture du site :

Pour l'architecture de notre site nous avons différents composants.

A savoir que dans chaque composant, nous avons 4 fichiers. Un fichier html et scss pour la partie design et visible du site. Un fichier spec.ts qui représente des tests unitaires, ici nous n'allons pas les toucher. Enfin un fichier ts (TypeScript) qui va nous permettre de rendre le site dynamique avec des interactions et des fonctionnalités que nous codons.







Le premier composant est “all-character”, comme son nom l’indique il nous permet d’afficher tous les caractères sur une même page, qui correspondent dans notre cas à toutes les cartes pokémons.

Le deuxième est “filter-bar” qui permet de gérer le filtre que nous appliquons sur la page all-character. Le troisième est “home”, il nous permet de gérer notre page d’accueil.

Le composant “models” est constitué d’un seul fichier, un fichier ts qui regroupe les différentes informations qui correspondent à ce que nous enverra l’API Pokémon.

Le suivant, “services” nous permet de gérer l’affichage d’une carte seule ou de plusieurs ainsi que de faire un appel à notre API pour pouvoir l’utiliser dans notre site.

Le composant “single-character” nous permet lui de faire afficher une carte individuellement dans une page.

A la suite de ces dossiers nous avons des fichiers individuels. Le premier est “app-routing.module.ts”, nous l’utilisons pour associer un composant à son lien, et ainsi gérer les liens entre chaque page.

Le deuxième et troisième sont les fichiers “app.component.html” et “app.component.scss” qui permettent de mettre en place le toolbar avec le menu ainsi que le footer qui seront présents sur toutes nos pages.

Le quatrième est “app.component.spec.ts” un fichier que nous ne touchons pas.

La cinquième est “app.component.ts” qui ici nous permet de définir le nom de notre site.

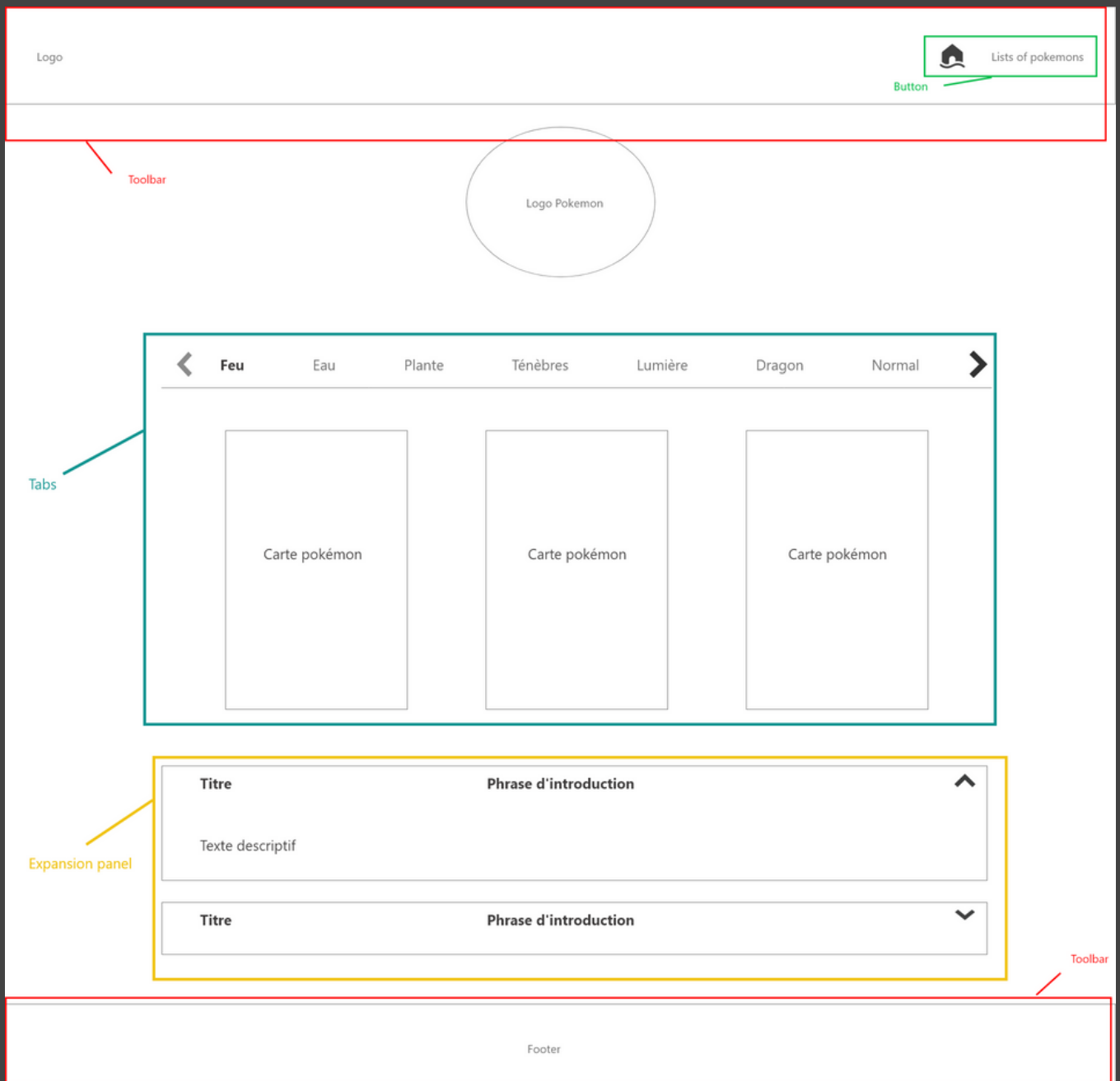
Enfin nous avons “app.module.ts” qui réunit tous les modules que nous utilisons dans le site comme les différents composants de Angular Material.

Il y a aussi le dossier assets qui réunit toutes nos images.



# Les maquettes

## La page d'accueil





# Les maquettes

## La page de toutes les cartes Pokémon

The wireframe illustrates the layout of a page for listing all Pokémon cards. It includes a top header with a logo and a navigation button, a central toolbar with search and filter options, a main content area with a grid of card placeholders, and a bottom footer. Annotations identify key UI components.

**Annotations:**

- Logo**: Located in the top left corner of the header.
- Button**: A button labeled "Lists of pokemons" in the top right corner of the header.
- Toolbar**: A horizontal bar containing the search and filter controls.
- Formulaire de recherche**: The search input field within the toolbar.
- Select type du pokémon**: The filter dropdown menu within the toolbar.
- Input**: Points to the search input field.
- From field**: Points to the filter dropdown menu.
- Select**: Points to the filter dropdown menu.
- Card**: Points to one of the six placeholder cards in the main content area.
- Footer**: Located at the bottom of the page.

**Card Structure:**

Nom du pokémon Type
Photo
Quelques caractéristiques



# Les maquettes

## La page d'une carte Pokémon

