



RESSOURCES

DÉVELOPPER VOTRE PREMIÈRE APPLICATION REACT.

« Vous êtes bien sur la page des ressources de la formation 'React : Développez votre première application avec TypeScript et les Hooks'.

Vous trouverez, comme promis, les éléments suivants :

Table des matières

- [L'application de démonstration](#)
- [La correction du code](#)
- [Les extraits de code](#)

Bon développement à vous,

Simon. »



Simon DIENY

Fondateur du site angularsenior.fr

L'application de démonstration

Lien vers [l'application de démonstration](#).

La correction du code

[react-pokemons-app-correction](#) Télécharger

Les extraits de code

package.json (React 16)

```
1 {
2   "name": "react-pokemons-app",
3   "version": "1.0.0",
4   "description": "An awesome application to handle some pokemons.",
5   "dependencies": {
6     "@types/node": "12.11.1",
7     "@types/react": "16.9.9",
8     "@types/react-dom": "16.9.2",
9     "@types/react-router-dom": "^5.1.2",
```

```

10     "react": "^16.10.2",
11     "react-dom": "^16.10.2",
12     "react-router-dom": "^5.1.2",
13     "react-scripts": "3.2.0",
14     "typescript": "3.6.4"
15   },
16   "scripts": {
17     "start": "react-scripts start",
18     "build": "react-scripts build"
19   },
20   "eslintConfig": {
21     "extends": "react-app"
22   },
23   "browserslist": {
24     "production": [
25       ">0.2%",
26       "not dead",
27       "not op_mini all"
28     ],
29     "development": [
30       "last 1 chrome version",
31       "last 1 firefox version",
32       "last 1 safari version"
33     ]
34   }
35 }

```

package.json (react 17)

```

1  {
2    "name": "react-pokemons-app",
3    "version": "1.0.0",
4    "description": "An awesome application to handle some pokemons.",
5    "dependencies": {
6      "@types/node": "12.11.1",
7      "@types/react": "16.9.9",
8      "@types/react-dom": "16.9.2",
9      "@types/react-router-dom": "^5.1.2",
10     "react": "^17.0.0",
11     "react-dom": "^17.0.0",
12     "react-router-dom": "^5.1.2",
13     "react-scripts": "^4.0.0",
14     "typescript": "3.6.4"
15   },
16   "scripts": {
17     "start": "react-scripts start",
18     "build": "react-scripts build"
19   },
20   "eslintConfig": {
21     "extends": "react-app"
22   },
23   "browserslist": {

```

```

24     "production": [
25         ">0.2%",
26         "not dead",
27         "not op_mini all"
28     ],
29     "development": [
30         "last 1 chrome version",
31         "last 1 firefox version",
32         "last 1 safari version"
33     ]
34 }
35 }

```

tsconfig.json (React 16)

```

1  {
2      "compilerOptions": {
3          "target": "es5",
4          "lib": [
5              "dom",
6              "dom.iterable",
7              "esnext"
8          ],
9          "allowJs": true,
10         "skipLibCheck": true,
11         "esModuleInterop": true,
12         "allowSyntheticDefaultImports": true,
13         "strict": true,
14         "forceConsistentCasingInFileNames": true,
15         "module": "esnext",
16         "moduleResolution": "node",
17         "resolveJsonModule": true,
18         "isolatedModules": true,
19         "noEmit": true,
20         "jsx": "react"
21     },
22     "include": [
23         "src"
24     ]
25 }

```

tsconfig.json (React 17)

```

1  {
2      "compilerOptions": {
3          "target": "es5",
4          "lib": [
5              "dom",
6              "dom.iterable",
7              "esnext"

```

```

8     ],
9     "allowJs": true,
10    "skipLibCheck": true,
11    "esModuleInterop": true,
12    "allowSyntheticDefaultImports": true,
13    "strict": true,
14    "forceConsistentCasingInFileNames": true,
15    "module": "esnext",
16    "moduleResolution": "node",
17    "resolveJsonModule": true,
18    "isolatedModules": true,
19    "noEmit": true,
20    "jsx": "react",
21    "noFallthroughCasesInSwitch": true
22  },
23  "include": [
24    "src"
25  ]
26 }

```

App.tsx

```

1  import React from 'react';
2
3  const App: React.FC = () => {
4    const name: String = 'React';
5
6    return (
7      <h1>Hello, {name} !</h1>
8    )
9  }
10
11 export default App;

```

pokemon.html

```

1  <!DOCTYPE html>
2  <html lang="fr">
3    <head>
4      <meta charset="utf-8" />
5      <meta name="viewport" content="width=device-width, initial-scal
6      <title>Pokédex</title>
7    </head>
8    <body>
9      <noscript>You need to enable JavaScript to run this app.</noscr
10     <div id="root">L'application est en cours de chargement...</div
11   </body>
12 </html>

```

models/pokemon.ts

```
1  export default class Pokemon {
2    // 1. Typage des propriétés d'un pokémon.
3    id: number;
4    hp: number;
5    cp: number;
6    name: string;
7    picture: string;
8    types: Array<string>;
9    created: Date;
10
11   // 2. Définition des valeurs par défaut des propriétés d'un pokémon.
12   constructor(
13     id: number,
14     hp: number = 100,
15     cp: number = 10,
16     name: string = 'name',
17     picture: string = 'http://...',
18     types: Array<string> = ['Normal'],
19     created: Date = new Date()
20   ) {
21     // 3. Initialisation des propriétés d'un pokémon.
22     this.id = id;
23     this.hp = hp;
24     this.cp = cp;
25     this.name = name;
26     this.picture = picture;
27     this.types = types;
28     this.created = created;
29   }
30 }
```

models/mock-pokemon.ts

```
1  import Pokemon from './pokemon';
2
3  export const POKEMONS: Pokemon[] = [
4    {
5      id: 1,
6      name: "Bulbizarre",
7      hp: 25,
8      cp: 5,
9      picture: "https://assets.pokemon.com/assets/cms2/img/pokedex/detail/001.png",
10     types: ["Plante", "Poison"],
11     created: new Date()
12   },
13   {
14     id: 2,
```

```
15     name: "Salamèche",
16     hp: 28,
17     cp: 6,
18     picture: "https://assets.pokemon.com/assets/cms2/img/pokedex/det
19     types: ["Feu"],
20     created: new Date()
21 },
22 {
23     id: 3,
24     name: "Carapuce",
25     hp: 21,
26     cp: 4,
27     picture: "https://assets.pokemon.com/assets/cms2/img/pokedex/det
28     types: ["Eau"],
29     created: new Date()
30 },
31 {
32     id: 4,
33     name: "Aspicot",
34     hp: 16,
35     cp: 2,
36     picture: "https://assets.pokemon.com/assets/cms2/img/pokedex/det
37     types: ["Insecte", "Poison"],
38     created: new Date()
39 },
40 {
41     id: 5,
42     name: "Roucool",
43     hp: 30,
44     cp: 7,
45     picture: "https://assets.pokemon.com/assets/cms2/img/pokedex/det
46     types: ["Normal", "Vol"],
47     created: new Date()
48 },
49 {
50     id: 6,
51     name: "Rattata",
52     hp: 18,
53     cp: 6,
54     picture: "https://assets.pokemon.com/assets/cms2/img/pokedex/det
55     types: ["Normal"],
56     created: new Date()
57 },
58 {
59     id: 7,
60     name: "Piafabec",
61     hp: 14,
62     cp: 5,
63     picture: "https://assets.pokemon.com/assets/cms2/img/pokedex/det
64     types: ["Normal", "Vol"],
65     created: new Date()
66 },
67 {
68     id: 8,
```

```

69     name: "Abo",
70     hp: 16,
71     cp: 4,
72     picture: "https://assets.pokemon.com/assets/cms2/img/pokedex/det
73     types: ["Poison"],
74     created: new Date()
75 },
76 {
77     id: 9,
78     name: "Pikachu",
79     hp: 21,
80     cp: 7,
81     picture: "https://assets.pokemon.com/assets/cms2/img/pokedex/det
82     types: ["Electrik"],
83     created: new Date()
84 },
85 {
86     id: 10,
87     name: "Sabelette",
88     hp: 19,
89     cp: 3,
90     picture: "https://assets.pokemon.com/assets/cms2/img/pokedex/det
91     types: ["Normal"],
92     created: new Date()
93 },
94 {
95     id: 11,
96     name: "Mélofée",
97     hp: 25,
98     cp: 5,
99     picture: "https://assets.pokemon.com/assets/cms2/img/pokedex/det
100    types: ["Fée"],
101    created: new Date()
102 },
103 {
104     id: 12,
105     name: "Groupix",
106     hp: 17,
107     cp: 8,
108     picture: "https://assets.pokemon.com/assets/cms2/img/pokedex/det
109     types: ["Feu"],
110     created: new Date()
111 }
112 ];
113
114 export default POKEMONS;

```

CODE .formatType

```

1  const formatType = (type: string): string => {
2      let color: string;
3

```



```

4   switch (type) {
5     case 'Feu':
6       color = 'red lighten-1';
7       break;
8     case 'Eau':
9       color = 'blue lighten-1';
10      break;
11     case 'Plante':
12       color = 'green lighten-1';
13       break;
14     case 'Insecte':
15       color = 'brown lighten-1';
16       break;
17     case 'Normal':
18       color = 'grey lighten-3';
19       break;
20     case 'Vol':
21       color = 'blue lighten-3';
22       break;
23     case 'Poison':
24       color = 'deep-purple accent-1';
25       break;
26     case 'Fée':
27       color = 'pink lighten-4';
28       break;
29     case 'Psy':
30       color = 'deep-purple darken-2';
31       break;
32     case 'Electrik':
33       color = 'lime accent-1';
34       break;
35     case 'Combat':
36       color = 'deep-orange';
37       break;
38     default:
39       color = 'grey';
40       break;
41   }
42
43   return `chip ${color}`;
44 }

```

pages/pokemon-list.tsx

```

1   import React, { FunctionComponent, useState, useEffect } from 'react';
2   import Pokemon from '../models/pokemon';
3   import POKEMONS from '../models/mock-pokemon';
4   import PokemonCard from '../components/pokemon-card';
5
6   const PokemonList: FunctionComponent = () => {
7     const [pokemons, setPokemons] = useState<Pokemon[]>([]);
8

```

```

9     useEffect(() => {
10         setPokemons(POKEMONS);
11     }, []);
12
13     return (
14         <div>
15             <h1 className="center">Pokédex</h1>
16             <div className="container">
17                 <div className="row">
18                     {pokemons.map(pokemon => (
19                         <PokemonCard key={pokemon.id} pokemon={pokemon}/>
20                     ))}
21                 </div>
22             </div>
23         </div>
24     );
25 }
26
27 export default PokemonList;

```

pages/pokemon-card.tsx

```

1  import React, { FunctionComponent } from 'react';
2  import Pokemon from '../models/pokemon';
3
4  type Props = {
5      pokemon: Pokemon
6  };
7
8  const PokemonCard: FunctionComponent<Props> = ({pokemon}) => {
9
10     return (
11         <div className="col s6 m4">
12             <div className="card horizontal">
13                 <div className="card-image">
14                     <img src={pokemon.picture} alt={pokemon.name}/>
15                 </div>
16                 <div className="card-stacked">
17                     <div className="card-content">
18                         <p>{pokemon.name}</p>
19                         <p><small>{pokemon.created.toString()}</small></p>
20                     </div>
21                 </div>
22             </div>
23         </div>
24     );
25 }
26
27 export default PokemonCard;

```

pages/pokemon-detail.tsx

```

1  import React, { FunctionComponent, useState, useEffect } from 'react'
2  import { RouteComponentProps, Link } from 'react-router-dom';
3  import Pokemon from '../models/pokemon';
4  import POKEMONS from '../models/mock-pokemon';
5  import formatDate from '../helpers/format-date';
6  import formatType from '../helpers/format-type';
7
8  type Params = { id: string };
9
10 const PokemonsDetail: FunctionComponent<RouteComponentProps<Params>
11
12     const [pokemon, setPokemon] = useState<Pokemon|null>(null);
13
14     useEffect(() => {
15         POKEMONS.forEach(pokemon => {
16             if (match.params.id === pokemon.id.toString()) {
17                 setPokemon(pokemon);
18             }
19         })
20     }, [match.params.id]);
21
22     return (
23         <div>
24             { pokemon ? (
25                 <div className="row">
26                     <div className="col s12 m8 offset-m2">
27                         <h2 className="header center">{ pokemon.name }</h2>
28                         <div className="card hoverable">
29                             <div className="card-image">
30                                 <img src={pokemon.picture} alt={pokemon.name} style
31                             </div>
32                             <div className="card-stacked">
33                                 <div className="card-content">
34                                     <table className="bordered striped">
35                                         <tbody>
36                                             <tr>
37                                                 <td>Nom</td>
38                                                 <td><strong>{ pokemon.name }</strong></td>
39                                             </tr>
40                                             <tr>
41                                                 <td>Points de vie</td>
42                                                 <td><strong>{ pokemon.hp }</strong></td>
43                                             </tr>
44                                             <tr>
45                                                 <td>Dégâts</td>
46                                                 <td><strong>{ pokemon.cp }</strong></td>
47                                             </tr>
48                                             <tr>
49                                                 <td>Types</td>
50                                                 <td>
51                                                     {pokemon.types.map(type => (
52                                                         <span key={type} className={formatType(t
53

```

```

54         </tr>
55         <tr>
56             <td>Date de création</td>
57             <td>{formatDate(pokemon.created)}</td>
58         </tr>
59     </tbody>
60 </table>
61 </div>
62 <div className="card-action">
63     <Link to="/">Retour</Link>
64 </div>
65 </div>
66 </div>
67 </div>
68 </div>
69     ) : (
70         <h4 className="center">Aucun pokémon à afficher !</h4>
71     )}
72 </div>
73 );
74 }
75
76 export default PokemonsDetail;

```

pages/page-not-found.tsx

```

1  import React, { FunctionComponent } from 'react';
2  import { Link } from 'react-router-dom';
3
4  const PageNotFound: FunctionComponent = () => {
5
6      return (
7          <div className="center">
8              
9              <h1>Hey, cette page n'existe pas !</h1>
10             <Link to="/" className="waves-effect waves-teal btn-flat">
11                 Retourner à l'accueil
12             </Link>
13         </div>
14     );
15 }
16
17 export default PageNotFound;

```

components/pokemon-form.tsx

```

1  import React, { FunctionComponent } from 'react';
2  import Pokemon from '../models/pokemon';
3  import formatType from '../helpers/format-type';

```

```

4
5 type Props = {
6   pokemon: Pokemon
7 };
8
9 const PokemonForm: FunctionComponent<Props> = ({pokemon}) => {
10
11   const types: string[] = [
12     'Plante', 'Feu', 'Eau', 'Insecte', 'Normal', 'Electrik',
13     'Poison', 'Fée', 'Vol', 'Combat', 'Psy'
14   ];
15
16   return (
17     <form>
18       <div className="row">
19         <div className="col s12 m8 offset-m2">
20           <div className="card hoverable">
21             <div className="card-image">
22               <img src={pokemon.picture} alt={pokemon.name} style={
23             </div>
24             <div className="card-stacked">
25               <div className="card-content">
26                 {/* Pokemon name */}
27                 <div className="form-group">
28                   <label htmlFor="name">Nom</label>
29                   <input id="name" type="text" className="form-cont
30                 </div>
31                 {/* Pokemon hp */}
32                 <div className="form-group">
33                   <label htmlFor="hp">Point de vie</label>
34                   <input id="hp" type="number" className="form-cont
35                 </div>
36                 {/* Pokemon cp */}
37                 <div className="form-group">
38                   <label htmlFor="cp">Dégâts</label>
39                   <input id="cp" type="number" className="form-cont
40                 </div>
41                 {/* Pokemon types */}
42                 <div className="form-group">
43                   <label>Types</label>
44                   {types.map(type => (
45                     <div key={type} style={{marginBottom: '10px'}}>
46                       <label>
47                         <input id={type} type="checkbox" className=
48                         <span>
49                           <p className={formatType(type)}>{ type }<
50                         </span>
51                       </label>
52                     </div>
53                   ))}
54                 </div>
55               </div>
56               <div className="card-action center">
57                 {/* Submit button */}

```

```

58         <button type="submit" className="btn">Valider</butt
59     </div>
60 </div>
61 </div>
62 </div>
63 </div>
64 </form>
65 );
66 };
67
68 export default PokemonForm;

```

pages/pokemon-edit.tsx

```

1  import React, { FunctionComponent, useState, useEffect } from 'react'
2  import { RouteComponentProps } from 'react-router-dom';
3  import PokemonForm from '../components/pokemon-form';
4  import Pokemon from '../models/pokemon';
5  import POKEMONS from '../models/mock-pokemon';
6
7  type Params = { id: string };
8
9  const PokemonEdit: FunctionComponent<RouteComponentProps<Params>> =
10
11     const [pokemon, setPokemon] = useState<Pokemon|null>(null);
12
13     useEffect(() => {
14         POKEMONS.forEach(pokemon => {
15             if (match.params.id === pokemon.id.toString()) {
16                 setPokemon(pokemon);
17             }
18         })
19     }, [match.params.id]);
20
21     return (
22         <div>
23             { pokemon ? (
24                 <div className="row">
25                     <h2 className="header center">Éditer { pokemon.name }</h2>
26                     <PokemonForm pokemon={pokemon}></PokemonForm>
27                 </div>
28             ) : (
29                 <h4 className="center">Aucun pokémon à afficher !</h4>
30             )}
31         </div>
32     );
33 }
34
35 export default PokemonEdit;

```

services/pokemon-service.ts

```
1 import Pokemon from "../models/pokemon";
2
3 export default class PokemonService {
4
5     static getPokemons(): Promise<Pokemon[]> {
6         return fetch('http://localhost:3001/pokemons')
7             .then(response => response.json());
8     }
9
10    static getPokemon(id: number): Promise<Pokemon|null> {
11        return fetch(`http://localhost:3001/pokemons/${id}`)
12            .then(response => response.json())
13            .then(data => this.isEmpty(data) ? null : data);
14    }
15
16    static isEmpty(data: Object): boolean {
17        return Object.keys(data).length === 0;
18    }
19 }
```

models/db.json

```
1 {
2     "pokemons": [
3         {
4             "id": 1,
5             "name": "Bulbizarre",
6             "hp": 25,
7             "cp": 5,
8             "picture": "https://assets.pokemon.com/assets/cms2/img/pokec",
9             "types": ["Plante", "Poison"]
10        },
11        {
12            "id": 2,
13            "name": "Salamèche",
14            "hp": 28,
15            "cp": 6,
16            "picture": "https://assets.pokemon.com/assets/cms2/img/pokec",
17            "types": ["Feu"]
18        },
19        {
20            "id": 3,
21            "name": "Carapuce",
22            "hp": 21,
23            "cp": 4,
24            "picture": "https://assets.pokemon.com/assets/cms2/img/pokec",
25            "types": ["Eau"]
26        },
27    ]
28 }
```

```
27 {
28   "id": 4,
29   "name": "Aspicot",
30   "hp": 16,
31   "cp": 2,
32   "picture": "https://assets.pokemon.com/assets/cms2/img/pokec
33   "types": ["Insecte", "Poison"]
34 },
35 {
36   "id": 5,
37   "name": "Roucool",
38   "hp": 30,
39   "cp": 7,
40   "picture": "https://assets.pokemon.com/assets/cms2/img/pokec
41   "types": ["Normal", "Vol"]
42 },
43 {
44   "id": 6,
45   "name": "Rattata",
46   "hp": 18,
47   "cp": 6,
48   "picture": "https://assets.pokemon.com/assets/cms2/img/pokec
49   "types": ["Normal"]
50 },
51 {
52   "id": 7,
53   "name": "Piafabec",
54   "hp": 14,
55   "cp": 5,
56   "picture": "https://assets.pokemon.com/assets/cms2/img/pokec
57   "types": ["Normal", "Vol"]
58 },
59 {
60   "id": 8,
61   "name": "Abo",
62   "hp": 16,
63   "cp": 4,
64   "picture": "https://assets.pokemon.com/assets/cms2/img/pokec
65   "types": ["Poison"]
66 },
67 {
68   "id": 9,
69   "name": "Pikachu",
70   "hp": 21,
71   "cp": 7,
72   "picture": "https://assets.pokemon.com/assets/cms2/img/pokec
73   "types": ["Electrik"]
74 },
75 {
76   "id": 10,
77   "name": "Sabelette",
78   "hp": 19,
79   "cp": 3,
80   "picture": "https://assets.pokemon.com/assets/cms2/img/pokec
```



```

81         "types": ["Normal"]
82     },
83     {
84         "id": 11,
85         "name": "Mélofée",
86         "hp": 25,
87         "cp": 5,
88         "picture": "https://assets.pokemon.com/assets/cms2/img/pokec
89         "types": ["Fée"]
90     },
91     {
92         "id": 12,
93         "name": "Groupix",
94         "hp": 17,
95         "cp": 8,
96         "picture": "https://assets.pokemon.com/assets/cms2/img/pokec
97         "types": ["Feu"]
98     }
99 ]
100 }

```

components/pokemon-search.tsx

```

1  import React, { FunctionComponent, useState } from 'react';
2  import { Link } from 'react-router-dom';
3  import Pokemon from '../models/pokemon';
4  import PokemonService from '../services/pokemon-service';
5
6  const PokemonSearch: FunctionComponent = () => {
7
8      const [term, setTerm] = useState<string>('');
9      const [pokemons, setPokemons] = useState<Pokemon[]>([]);
10
11      const handleInputChange = (e: React.ChangeEvent<HTMLInputElement>) => {
12          const term = e.target.value;
13          setTerm(term);
14
15          if (term.length <= 1) {
16              setPokemons([]);
17              return;
18          }
19
20          PokemonService.searchPokemon(term).then(pokemons => setPokemons(pokemons));
21      }
22
23      return (
24          <div className="row">
25              <div className="col s12 m6 offset-m3">
26                  <div className="card">
27                      <div className="card-content">
28                          <div className="input-field">
29                              <input type="text" placeholder="Rechercher un pokémon" value={term} />

```

```

30     </div>
31     <div className='collection'>
32       {pokemons.map((pokemon) => (
33         <Link key={pokemon.id} to={` /pokemons/${pokemon.id}`} cla
34           {pokemon.name}
35         </Link>
36       ))}
37     </div>
38   </div>
39 </div>
40 </div>
41 </div>
42 );
43 }
44
45 export default PokemonSearch;

```

Components/loader.tsx

```

1  import React, { FunctionComponent } from 'react';
2
3  const Loader: FunctionComponent = () => {
4
5    return (
6      <div className="preloader-wrapper big active">
7        <div className="spinner-layer spinner-blue">
8          <div className="circle-clipper left">
9            <div className="circle"></div>
10         </div>
11         <div className="gap-patch">
12           <div className="circle"></div>
13         </div><div className="circle-clipper right">
14           <div className="circle"></div>
15         </div>
16       </div>
17     </div>
18   );
19 }
20
21 export default Loader;

```

pages/login.tsx

Récupérer le code initial du composant [login.tsx](#)

Important: Ce fichier doit être récupéré depuis un autre site où est hébergé le code de correction du cours. En effet, dans ce composant j'utilise des emojis ainsi que le

symbole « && », qui ne sont pas pris en compte par le plugin de coloration syntaxique utilisé sur ce site. Merci de votre compréhension ! 😊

PrivateRoute.tsx

```
1  import React from 'react';
2  import { Route, Redirect } from 'react-router-dom';
3  import AuthenticationService from '../services/authentication-service';
4
5  const PrivateRoute = ({ component: Component, ...rest }: any) => (
6    <Route {...rest} render={props => {
7      const isAuthenticated = AuthenticationService.isAuthenticated;
8      if (!isAuthenticated) {
9        return <Redirect to={{ pathname: '/login' }} />
10     }
11
12     return <Component {...props} />
13   }} />
14 );
15
16 export default PrivateRoute;
```

services/pokemon-service.ts (version de PRODUCTION)

```
1  import Pokemon from '../models/pokemon';
2  import POKEMONS from '../models/mock-pokemon';
3
4  export default class PokemonService {
5
6    static pokemons: Pokemon[] = POKEMONS;
7
8    static isDev = (!process.env.NODE_ENV || process.env.NODE_ENV ==
9
10   static getPokemons(): Promise<Pokemon[]> {
11     if (this.isDev) {
12       return fetch('http://localhost:3001/pokemons')
13         .then(response => response.json())
14         .catch(error => this.handleError(error));
15     }
16
17     return new Promise(resolve => {
18       resolve(this.pokemons);
19     });
20   }
21
22   static getPokemon(id: number): Promise<Pokemon | null> {
23     if (this.isDev) {
24       return fetch(`http://localhost:3001/pokemons/${id}`)
```

```

25     .then(response => response.json())
26     .then(data => this.isEmpty(data) ? null : data)
27     .catch(error => this.handleError(error));
28 }
29
30 return new Promise(resolve => {
31     resolve(this.pokemons.find(pokemon => id === pokemon.id));
32 });
33 }
34
35 static updatePokemon(pokemon: Pokemon): Promise<Pokemon> {
36     if(this.isDev) {
37         return fetch(`http://localhost:3001/pokemons/${pokemon.id}`,
38             method: 'PUT',
39             body: JSON.stringify(pokemon),
40             headers: { 'Content-Type': 'application/json' }
41         )
42         .then(response => response.json())
43         .catch(error => this.handleError(error));
44     }
45
46     return new Promise(resolve => {
47         const { id } = pokemon;
48         const index = this.pokemons.findIndex(pokemon => pokemon.id
49             === pokemon.id);
50         this.pokemons[index] = pokemon;
51         resolve(pokemon);
52     });
53 }
54
55 static deletePokemon(pokemon: Pokemon): Promise<{}> {
56     if(this.isDev) {
57         return fetch(`http://localhost:3001/pokemons/${pokemon.id}`,
58             method: 'DELETE',
59             headers: { 'Content-Type': 'application/json' }
60         )
61         .then(response => response.json())
62         .catch(error => this.handleError(error));
63     }
64
65     return new Promise(resolve => {
66         const { id } = pokemon;
67         this.pokemons = this.pokemons.filter(pokemon => pokemon.id !== id);
68         resolve({});
69     });
70 }
71
72 static addPokemon(pokemon: Pokemon): Promise<Pokemon> {
73     pokemon.created = new Date(pokemon.created);
74
75     if(this.isDev) {
76         return fetch(`http://localhost:3001/pokemons`, {
77             method: 'POST',
78             body: JSON.stringify(pokemon),
79             headers: { 'Content-Type': 'application/json' }

```

```

79         })
80         .then(response => response.json())
81         .catch(error => this.handleError(error));
82     }
83
84     return new Promise(resolve => {
85         this.pokemons.push(pokemon);
86         resolve(pokemon);
87     });
88 }
89
90 static searchPokemon(term: string): Promise<Pokemon[]> {
91     if(this.isDev) {
92         return fetch(`http://localhost:3001/pokemons?q=${term}`)
93             .then(response => response.json())
94             .catch(error => this.handleError(error));
95     }
96
97     return new Promise(resolve => {
98         const results = this.pokemons.filter(pokemon => pokemon.name
99         resolve(results);
100     });
101
102 }
103
104 static isEmpty(data: Object): boolean {
105     return Object.keys(data).length === 0;
106 }
107
108 static handleError(error: Error): void {
109     console.error(error);
110 }
111 }

```