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

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)

```
{
  "name": "react-pokemons-app",
  "version": "1.0.0",
  "description": "An awesome application to handle some pokemons.",
  "dependencies": {
    "@types/node": "12.11.1",
    "@types/react": "16.9.9",
    "@types/react-dom": "16.9.2",
    "@types/react-router-dom": "^5.1.2",
    "react": "^16.10.2",
    "react-dom": "^16.10.2",
    "react-router-dom": "^5.1.2",
    "react-scripts": "3.2.0",
    "typescript": "3.6.4"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build"
  },
  "eslintConfig": {
    "extends": "react-app"
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}
```

package.json (React 17)

```
{
  "name": "react-pokemons-app",
  "version": "1.0.0",
  "description": "An awesome application to handle some pokemons.",
  "dependencies": {
    "@types/node": "12.11.1",
    "@types/react": "16.9.9",
    "@types/react-dom": "16.9.2",
    "@types/react-router-dom": "^5.1.2",
    "react": "^17.0.0",
    "react-dom": "^17.0.0",
    "react-router-dom": "^5.1.2",
    "react-scripts": "^4.0.0",
    "typescript": "3.6.4"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build"
  },
  "eslintConfig": {
    "extends": "react-app"
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}
```

tsconfig.json (React 16)

```
{
  "compilerOptions": {
    "target": "es5",
    "lib": [
      "dom",
      "dom.iterable",
      "esnext"
    ],
    "allowJs": true,
    "skipLibCheck": true,
    "esModuleInterop": true,
    "allowSyntheticDefaultImports": true,
    "strict": true,
    "forceConsistentCasingInFileNames": true,
    "module": "esnext",
    "moduleResolution": "node",
    "resolveJsonModule": true,
    "isolatedModules": true,
    "noEmit": true,
    "jsx": "react"
  },
  "include": [
    "src"
  ]
}
```

tsconfig.json (React 17)

```
{
  "compilerOptions": {
    "target": "es5",
    "lib": [
      "dom",
      "dom.iterable",
      "esnext"
    ],
    "allowJs": true,
    "skipLibCheck": true,
    "esModuleInterop": true,
    "allowSyntheticDefaultImports": true,
    "strict": true,
    "forceConsistentCasingInFileNames": true,
    "module": "esnext",
    "moduleResolution": "node",
    "resolveJsonModule": true,
    "isolatedModules": true,
    "noEmit": true,
    "jsx": "react",
    "noFallthroughCasesInSwitch": true
  },
  "include": [
    "src"
  ]
}
```

App.tsx

```
import React from 'react';

const App: React.FC = () => {
 const name: String = 'React';

 return (
  <h1>Hello, {name} !</h1>
 )
}

export default App;
```

pokemon.html

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scal
    <title>Pokédex</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscr
    <div id="root">L'application est en cours de chargement...</di
  </body>
</html>
```

models/pokemon.ts

```typescript
export default class Pokemon {
 // 1. Typage des propiétés d'un pokémon.
 id: number;
 hp: number;
 cp: number;
 name: string;
 picture: string;
 types: Array<string>;
 created: Date;

 // 2. Définition des valeurs par défaut des propriétés d'un pokémo
 constructor(
  id: number,
  hp: number = 100,
  cp: number = 10,
  name: string = 'name',
  picture: string = 'http://...',
  types: Array<string> = ['Normal'],
  created: Date = new Date()
 ) {
  // 3. Initialisation des propiétés d'un pokémons.
  this.id = id;
  this.hp = hp;
  this.cp = cp;
  this.name = name;
  this.picture = picture;
  this.types = types;
  this.created = created;
 }
}
```

models/mock-pokemon.ts

```typescript
import Pokemon from './pokemon';

export const POKEMONS: Pokemon[] = [
 {
  id: 1,
  name: "Bulbizarre",
  hp: 25,
  cp: 5,
  picture: "https://assets.pokemon.com/assets/cms2/img/pokedex/det
  types: ["Plante", "Poison"],
  created: new Date()
 },
 {
  id: 2,
   name: "Salamèche",
   hp: 28,
   cp: 6,
   picture: "https://assets.pokemon.com/assets/cms2/img/pokedex/det
   types: ["Feu"],
   created: new Date()
  },
  {
   id: 3,
   name: "Carapuce",
   hp: 21,
   cp: 4,
   picture: "https://assets.pokemon.com/assets/cms2/img/pokedex/det
   types: ["Eau"],
   created: new Date()
  },
  {
   id: 4,
   name: "Aspicot",
   hp: 16,
   cp: 2,
   picture: "https://assets.pokemon.com/assets/cms2/img/pokedex/det
   types: ["Insecte", "Poison"],
   created: new Date()
  },
  {
   id: 5,
   name: "Roucool",
   hp: 30,
   cp: 7,
   picture: "https://assets.pokemon.com/assets/cms2/img/pokedex/det
   types: ["Normal", "Vol"],
   created: new Date()
  },
  {
   id: 6,
   name: "Rattata",
   hp: 18,
   cp: 6,
   picture: "https://assets.pokemon.com/assets/cms2/img/pokedex/det
   types: ["Normal"],
   created: new Date()
  },
  {
   id: 7,
   name: "Piafabec",
   hp: 14,
   cp: 5,
   picture: "https://assets.pokemon.com/assets/cms2/img/pokedex/det
   types: ["Normal", "Vol"],
   created: new Date()
  },
  {
   id: 8,
    name: "Abo",
    hp: 16,
    cp: 4,
    picture: "https://assets.pokemon.com/assets/cms2/img/pokedex/det
    types: ["Poison"],
    created: new Date()
  },
  {
    id: 9,
    name: "Pikachu",
    hp: 21,
    cp: 7,
    picture: "https://assets.pokemon.com/assets/cms2/img/pokedex/det
    types: ["Electrik"],
    created: new Date()
  },
  {
    id: 10,
    name: "Sabelette",
    hp: 19,
    cp: 3,
    picture: "https://assets.pokemon.com/assets/cms2/img/pokedex/det
    types: ["Normal"],
    created: new Date()
  },
  {
    id: 11,
    name: "Mélofée",
    hp: 25,
    cp: 5,
    picture: "https://assets.pokemon.com/assets/cms2/img/pokedex/det
    types: ["Fée"],
    created: new Date()
  },
  {
    id: 12,
    name: "Groupix",
    hp: 17,
    cp: 8,
    picture: "https://assets.pokemon.com/assets/cms2/img/pokedex/det
    types: ["Feu"],
    created: new Date()
  }
];

export default POKEMONS;
```

CODE : formatType

```
const formatType = (type: string): string => {
  let color: string;

  switch (type) {
    case 'Feu':
      color = 'red lighten-1';
      break;
    case 'Eau':
      color = 'blue lighten-1';
      break;
    case 'Plante':
      color = 'green lighten-1';
      break;
    case 'Insecte':
      color = 'brown lighten-1';
      break;
    case 'Normal':
      color = 'grey lighten-3';
      break;
    case 'Vol':
      color = 'blue lighten-3';
      break;
    case 'Poison':
      color = 'deep-purple accent-1';
      break;
    case 'Fée':
      color = 'pink lighten-4';
      break;
    case 'Psy':
      color = 'deep-purple darken-2';
      break;
    case 'Electrik':
      color = 'lime accent-1';
      break;
    case 'Combat':
      color = 'deep-orange';
      break;
    default:
      color = 'grey';
      break;
  }

  return `chip ${color}`;
}
```

pages/pokemon-list.tsx

```
import React, { FunctionComponent, useState, useEffect } from 'reac
import Pokemon from '../models/pokemon';
import POKEMONS from '../models/mock-pokemon';
import PokemonCard from '../components/pokemon-card';

const PokemonList: FunctionComponent = () => {
  const [pokemons, setPokemons] = useState<Pokemon[]>([]);

  useEffect(() => {
    setPokemons(POKEMONS);
  }, []);

  return (
    <div>
      <h1 className="center">Pokédex</h1>
      <div className="container">
        <div className="row">
        {pokemons.map(pokemon => (
          <PokemonCard key={pokemon.id} pokemon={pokemon}/>
        ))}
        </div>
      </div>
    </div>
  );
}

export default PokemonList;
```

pages/pokemon-card.tsx

```
import React, { FunctionComponent } from 'react';
import Pokemon from '../models/pokemon';

type Props = {
  pokemon: Pokemon
};

const PokemonCard: FunctionComponent<Props> = ({pokemon}) => {

  return (
    <div className="col s6 m4">
      <div className="card horizontal">
        <div className="card-image">
          <img src={pokemon.picture} alt={pokemon.name}/>
        </div>
        <div className="card-stacked">
          <div className="card-content">
            <p>{pokemon.name}</p>
            <p><small>{pokemon.created.toString()}</small></p>
          </div>
        </div>
      </div>
    </div>
  );
}

export default PokemonCard;
```

pages/pokemon-detail.tsx

```
import React, { FunctionComponent, useState, useEffect } from 'reac
import { RouteComponentProps, Link } from 'react-router-dom';
import Pokemon from '../models/pokemon';
import POKEMONS from '../models/mock-pokemon';
import formatDate from '../helpers/format-date';
import formatType from '../helpers/format-type';

type Params = { id: string };

const PokemonsDetail: FunctionComponent<RouteComponentProps<Params>

  const [pokemon, setPokemon] = useState<Pokemon|null>(null);

  useEffect(() => {
    POKEMONS.forEach(pokemon => {
      if (match.params.id === pokemon.id.toString()) {
        setPokemon(pokemon);
      }
    })
  }, [match.params.id]);

  return (
    <div>
      { pokemon ? (
        <div className="row">
          <div className="col s12 m8 offset-m2">
            <h2 className="header center">{ pokemon.name }</h2>
            <div className="card hoverable">
              <div className="card-image">
                <img src={pokemon.picture} alt={pokemon.name} style
              </div>
              <div className="card-stacked">
                <div className="card-content">
                  <table className="bordered striped">
                    <tbody>
                      <tr>
                        <td>Nom</td>
                        <td><strong>{ pokemon.name }</strong></td>
                      </tr>
                      <tr>
                        <td>Points de vie</td>
                        <td><strong>{ pokemon.hp }</strong></td>
                      </tr>
                      <tr>
                        <td>Dégâts</td>
                        <td><strong>{ pokemon.cp }</strong></td>
                      </tr>
                      <tr>
                        <td>Types</td>
                        <td>
                          {pokemon.types.map(type => (
                           <span key={type} className={formatType(t
                          ))}</td>
```

```
           </tr>
           <tr>
            <td>Date de création</td>
            <td>{formatDate(pokemon.created)}</td>
           </tr>
          </tbody>
         </table>
        </div>
        <div className="card-action">
         <Link to="/">Retour</Link>
        </div>
       </div>
      </div>
     </div>
    </div>
   ) : (
    <h4 className="center">Aucun pokémon à afficher !</h4>
   )}
  </div>
 );
}

export default PokemonsDetail;
```

pages/page-not-found.tsx

```
import React, { FunctionComponent } from 'react';
import { Link } from 'react-router-dom';

const PageNotFound: FunctionComponent = () => {

 return (
  <div className="center">
   <img src="http://assets.pokemon.com/assets/cms2/img/pokedex/f
   <h1>Hey, cette page n'existe pas !</h1>
   <Link to="/" className="waves-effect waves-teal btn-flat">
    Retourner à l'accueil
   </Link>
  </div>
 );
}

export default PageNotFound;
```

components/pokemon-form.tsx

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

```

type Props = {
  pokemon: Pokemon
};

const PokemonForm: FunctionComponent<Props> = ({pokemon}) => {

  const types: string[] = [
    'Plante', 'Feu', 'Eau', 'Insecte', 'Normal', 'Electrik',
    'Poison', 'Fée', 'Vol', 'Combat', 'Psy'
  ];

  return (
    <form>
      <div className="row">
        <div className="col s12 m8 offset-m2">
          <div className="card hoverable">
            <div className="card-image">
              <img src={pokemon.picture} alt={pokemon.name} style={
            </div>
            <div className="card-stacked">
              <div className="card-content">
                {/* Pokemon name */}
                <div className="form-group">
                  <label htmlFor="name">Nom</label>
                  <input id="name" type="text" className="form-cont
                </div>
                {/* Pokemon hp */}
                <div className="form-group">
                  <label htmlFor="hp">Point de vie</label>
                  <input id="hp" type="number" className="form-cont
                </div>
                {/* Pokemon cp */}
                <div className="form-group">
                  <label htmlFor="cp">Dégâts</label>
                  <input id="cp" type="number" className="form-cont
                </div>
                {/* Pokemon types */}
                <div className="form-group">
                  <label>Types</label>
                  {types.map(type => (
                    <div key={type} style={{marginBottom: '10px'}}
                      <label>
                        <input id={type} type="checkbox" className=
                        <span>
                          <p className={formatType(type)}>{ type }
                        </span>
                      </label>
                    </div>
                  ))}
                </div>
              </div>
              <div className="card-action center">
                {/* Submit button */}
```

```
                  <button type="submit" className="btn">Valider</butt
                </div>
              </div>
            </div>
          </div>
        </div>
      </form>
    );
  };

  export default PokemonForm;
```

pages/pokemon-edit.tsx

```
import React, { FunctionComponent, useState, useEffect } from 'reac
import { RouteComponentProps } from 'react-router-dom';
import PokemonForm from '../components/pokemon-form';
import Pokemon from '../models/pokemon';
import POKEMONS from '../models/mock-pokemon';

type Params = { id: string };

const PokemonEdit: FunctionComponent<RouteComponentProps<Params>> =

  const [pokemon, setPokemon] = useState<Pokemon|null>(null);

  useEffect(() => {
    POKEMONS.forEach(pokemon => {
      if (match.params.id === pokemon.id.toString()) {
        setPokemon(pokemon);
      }
    })
  }, [match.params.id]);

  return (
    <div>
      { pokemon ? (
        <div className="row">
            <h2 className="header center">Éditer { pokemon.name }</
            <PokemonForm pokemon={pokemon}></PokemonForm>
        </div>
      ) : (
        <h4 className="center">Aucun pokémon à afficher !</h4>
      )}
    </div>
  );
}

export default PokemonEdit;
```

services/pokemon-service.ts

```ts
import Pokemon from "../models/pokemon";

export default class PokemonService {

  static getPokemons(): Promise<Pokemon[]> {
    return fetch('http://localhost:3001/pokemons')
      .then(response => response.json());
  }

  static getPokemon(id: number): Promise<Pokemon|null> {
    return fetch(`http://localhost:3001/pokemons/${id}`)
      .then(response => response.json())
      .then(data => this.isEmpty(data) ? null : data);
  }

  static isEmpty(data: Object): boolean {
    return Object.keys(data).length === 0;
  }
}
```

models/db.json

```json
{
  "pokemons": [
    {
      "id": 1,
      "name": "Bulbizarre",
      "hp": 25,
      "cp": 5,
      "picture": "https://assets.pokemon.com/assets/cms2/img/poked
      "types": ["Plante", "Poison"]
     },
     {
      "id": 2,
      "name": "Salamèche",
      "hp": 28,
      "cp": 6,
      "picture": "https://assets.pokemon.com/assets/cms2/img/poked
      "types": ["Feu"]
     },
     {
      "id": 3,
      "name": "Carapuce",
      "hp": 21,
      "cp": 4,
      "picture": "https://assets.pokemon.com/assets/cms2/img/poked
      "types": ["Eau"]
     },
    {
     "id": 4,
     "name": "Aspicot",
     "hp": 16,
     "cp": 2,
     "picture": "https://assets.pokemon.com/assets/cms2/img/poked
     "types": ["Insecte", "Poison"]
    },
    {
     "id": 5,
     "name": "Roucool",
     "hp": 30,
     "cp": 7,
     "picture": "https://assets.pokemon.com/assets/cms2/img/poked
     "types": ["Normal", "Vol"]
    },
    {
     "id": 6,
     "name": "Rattata",
     "hp": 18,
     "cp": 6,
     "picture": "https://assets.pokemon.com/assets/cms2/img/poked
     "types": ["Normal"]
    },
    {
     "id": 7,
     "name": "Piafabec",
     "hp": 14,
     "cp": 5,
     "picture": "https://assets.pokemon.com/assets/cms2/img/poked
     "types": ["Normal", "Vol"]
    },
    {
     "id": 8,
     "name": "Abo",
     "hp": 16,
     "cp": 4,
     "picture": "https://assets.pokemon.com/assets/cms2/img/poked
     "types": ["Poison"]
    },
    {
     "id": 9,
     "name": "Pikachu",
     "hp": 21,
     "cp": 7,
     "picture": "https://assets.pokemon.com/assets/cms2/img/poked
     "types": ["Electrik"]
    },
    {
     "id": 10,
     "name": "Sabelette",
     "hp": 19,
     "cp": 3,
     "picture": "https://assets.pokemon.com/assets/cms2/img/poked
```

```
      "types": ["Normal"]
     },
     {
      "id": 11,
      "name": "Mélofée",
      "hp": 25,
      "cp": 5,
      "picture": "https://assets.pokemon.com/assets/cms2/img/poked
      "types": ["Fée"]
     },
     {
      "id": 12,
      "name": "Groupix",
      "hp": 17,
      "cp": 8,
      "picture": "https://assets.pokemon.com/assets/cms2/img/poked
      "types": ["Feu"]
     }
  ]
}
```

components/pokemon-search.tsx

```
import React, { FunctionComponent, useState } from 'react';
import { Link } from 'react-router-dom';
import Pokemon from '../models/pokemon';
import PokemonService from '../services/pokemon-service';

const PokemonSearch: FunctionComponent = () => {

  const [term, setTerm] = useState<string>('');
  const [pokemons, setPokemons] = useState<Pokemon[]>([]);

  const handleInputChange = (e: React.ChangeEvent<HTMLInputElement>
    const term = e.target.value;
    setTerm(term);

    if(term.length <= 1) {
      setPokemons([]);
      return;
    }

    PokemonService.searchPokemon(term).then(pokemons => setPokemons
  }

  return (
    <div className="row">
    <div className="col s12 m6 offset-m3">
      <div className="card">
      <div className="card-content">
        <div className="input-field">
        <input type="text" placeholder="Rechercher un pokémon" valu
```

```
          </div>
          <div className='collection'>
          {pokemons.map((pokemon) => (
            <Link key={pokemon.id} to={`/pokemons/${pokemon.id}`} cla
              {pokemon.name}
            </Link>
          ))}
          </div>
        </div>
        </div>
      </div>
      </div>
    );
}

export default PokemonSearch;
```

components/loader.tsx

```
import React, { FunctionComponent } from 'react';

const Loader: FunctionComponent = () => {

  return (
    <div className="preloader-wrapper big active">
      <div className="spinner-layer spinner-blue">
        <div className="circle-clipper left">
          <div className="circle"></div>
        </div>
        <div className="gap-patch">
          <div className="circle"></div>
        </div><div className="circle-clipper right">
          <div className="circle"></div>
        </div>
      </div>
     </div>
   );
}

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

```tsx
import React from 'react';
import { Route, Redirect } from 'react-router-dom';
import AuthenticationService from './services/authentication-servic

const PrivateRoute = ({ component: Component, ...rest }: any) => (
  <Route {...rest} render={(props) => {
    const isAuthenticated = AuthenticationService.isAuthenticated;
    if (!isAuthenticated) {
      return <Redirect to={{ pathname: '/login' }} />
    }

    return <Component {...props} />
  }} />
);

export default PrivateRoute;
```

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

```ts
import Pokemon from "../models/pokemon";
import POKEMONS from "../models/mock-pokemon";

export default class PokemonService {

  static pokemons:Pokemon[] = POKEMONS;

  static isDev = (!process.env.NODE_ENV || process.env.NODE_ENV ==

  static getPokemons(): Promise<Pokemon[]> {
    if(this.isDev) {
      return fetch('http://localhost:3001/pokemons')
      .then(response => response.json())
      .catch(error => this.handleError(error));
    }

    return new Promise(resolve => {
      resolve(this.pokemons);
    });
  }

  static getPokemon(id: number): Promise<Pokemon|null> {
    if(this.isDev) {
      return fetch(`http://localhost:3001/pokemons/${id}`)
      .then(response => response.json())
      .then(data => this.isEmpty(data) ? null : data)
      .catch(error => this.handleError(error));
    }

    return new Promise(resolve => {
      resolve(this.pokemons.find(pokemon => id === pokemon.id));
    });
  }

  static updatePokemon(pokemon: Pokemon): Promise<Pokemon> {
    if(this.isDev) {
      return fetch(`http://localhost:3001/pokemons/${pokemon.id}`,
        method: 'PUT',
        body: JSON.stringify(pokemon),
        headers: { 'Content-Type': 'application/json'}
      })
      .then(response => response.json())
      .catch(error => this.handleError(error));
    }

    return new Promise(resolve => {
      const { id } = pokemon;
      const index = this.pokemons.findIndex(pokemon => pokemon.id
      this.pokemons[index] = pokemon;
      resolve(pokemon);
    });
  }

  static deletePokemon(pokemon: Pokemon): Promise<{}> {
    if(this.isDev) {
      return fetch(`http://localhost:3001/pokemons/${pokemon.id}`,
        method: 'DELETE',
        headers: { 'Content-Type': 'application/json'}
      })
      .then(response => response.json())
      .catch(error => this.handleError(error));
    }

    return new Promise(resolve => {
      const { id } = pokemon;
      this.pokemons = this.pokemons.filter(pokemon => pokemon.id !
      resolve({});
    });
  }

  static addPokemon(pokemon: Pokemon): Promise<Pokemon> {
    pokemon.created = new Date(pokemon.created);

    if(this.isDev) {
      return fetch(`http://localhost:3001/pokemons`, {
        method: 'POST',
        body: JSON.stringify(pokemon),
        headers: { 'Content-Type': 'application/json'}
      })
      .then(response => response.json())
      .catch(error => this.handleError(error));
    }

    return new Promise(resolve => {
      this.pokemons.push(pokemon);
      resolve(pokemon);
    });
  }

  static searchPokemon(term: string): Promise<Pokemon[]> {
    if(this.isDev) {
      return fetch(`http://localhost:3001/pokemons?q=${term}`)
      .then(response => response.json())
      .catch(error => this.handleError(error));
    }

    return new Promise(resolve => {
      const results = this.pokemons.filter(pokemon => pokemon.name
      resolve(results);
    });

  }

  static isEmpty(data: Object): boolean {
    return Object.keys(data).length === 0;
  }

  static handleError(error: Error): void {
    console.error(error);
  }
}
```