

Sprawozdanie nr 05

Temat: React

Teoria:

React – Obecnie jeden z najpopularniejszych frameworków JavaScript, a tak właściwie biblioteki stworzonej przez Facebook do budowania interfejsów stron internetowych.

Zalety React:

- lekkość
- szybkość
- modułowość (oparcie na komponentach)
- skalowalność
- reaktywność
- rozwijany i utrzymywany jest przez wielką organizację – Facebook (co gwarantuje, że projekt taki nie upadnie z dnia na dzień)
- jest popularny – dostępne jest do niego wiele gotowych rozwiązań
- istnieje dodatkowo ReactNative, dzięki któremu możemy stworzyć aplikację na systemy Android i IOS

Do głównych cech, które wyróżniają bibliotekę React jest wirtualny DOM. Gdy jest jakaś zmiana do wykonania np. zmiana nagłówka, wyszukuje on różnice pomiędzy wirtualnym oraz prawdziwym DOM i wprowadza zmiany tylko tam gdzie jest to potrzebne.

Dodatkowo posiada on nakładkę JSX, która niejako rozszerza język JavaScript o tagi HTML – pozwala wstawiać tagi HTML bezpośrednio w kodzie, zamiast ciągu znaków.

Przebieg zadania:

Na samym początku należy stworzyć projekt. Używamy w tym celu polecenia:

```
npx create-react-app [nazwa_projektu]
```

w moim wypadku projekt nazywać się będzie csgo-teams, jako że będzie on wyprowadzał profesjonalne drużyny Counter-Strike przez API stworzone w poprzednim zadaniu.

```
PS D:\Projekt\Sprawozdanie 5> npx create-react-app csgo-teams

Creating a new React app in D:\Projekt\Sprawozdanie 5\csgo-teams.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...
```

Po odczekaniu chwili projekt jest gotowy i pojawiają nam się następujące pliki:

```
▼ SPRAWOZDANIE 5
  ▼ csgo-teams
    > node_modules
    > public
    > src
    ◆ .gitignore
    {} package-lock.json
    {} package.json
    ⓘ README.md
```

Następnym krokiem jest zainstalowanie biblioteki do obsługi rutowania, jest to polecenie:

```
npm install react-router-dom
```

```
PS D:\Projekt\Sprawozdanie 5\csgo-teams> npm install react-router-dom
```

Po wykonaniu instalacji możemy odpalić serwer deweloperski, polecenie: `npm start`

```
You can now view csgo-teams in the browser.

Local:      http://localhost:3001
On Your Network: http://192.168.1.30:3001

Note that the development build is not optimized.
To create a production build, use npm run build.
```

Domyślnym portem serwera jest port 3000, ale jako że na tym porcie działa aktualnie inna aplikacja, React spytał się czy uruchomić aplikację na innym porcie, zgadzając się, React uruchomił ją na porcie 3001.

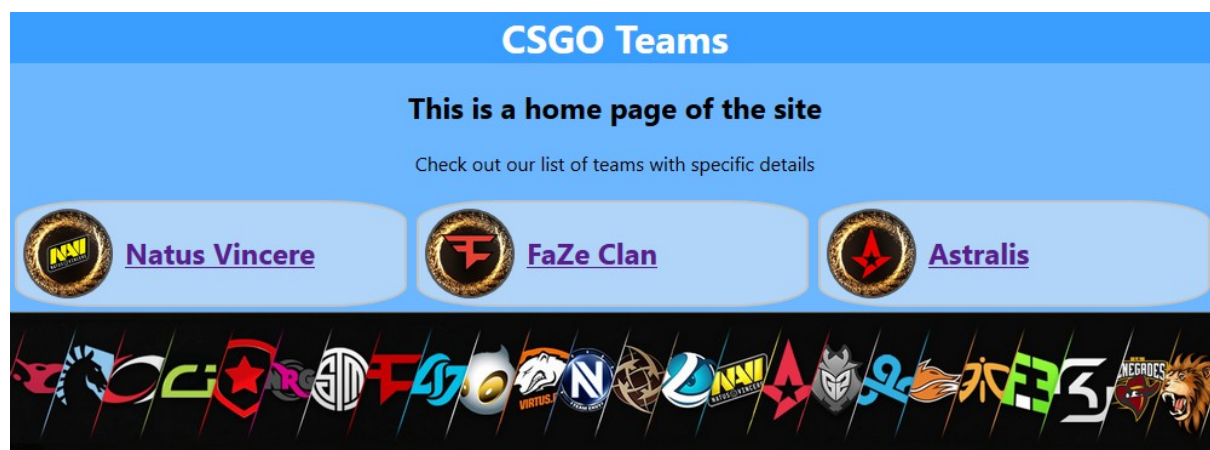
Wchodząc więc na adres <http://localhost:3001/> i widząc takie okno:



wiemy, że jesteśmy gotowi do działania i że aplikacja uruchomiła się poprawnie.

Zamysł aplikacji jest następujący – Użytkownik wchodząc na stronę główną widzi wszystkie drużyny dostępne za pośrednictwem TeamsAPI – API wystawionego w zadaniu 4. Klikając w nazwę drużyny, użytkownik przeniesiony zostanie na podstronę ze szczegółami drużyny, tj. nazwa, kraj pochodzenia, trener i aktualny skład.

Strona główna



Przykładowa strona drużyny Natus Vincere



Dodatkowo po kliknięciu w napis CSGO Teams, użytkownik zostaje przeniesiony z powrotem na stronę główną.

Struktura aplikacji jest następująca:

- App.js – główny plik aplikacji, w którym znajduje się definicja strony głównej oraz cała funkcjonalność routingu:
- Główną stroną jest strona główna – na niej wyświetlają się wszystkie dostępne drużyny.
- Po kliknięciu na odnośnik drużyny bądź wpisaniu adres/team/id, gdzie adres oznacza adres strony, a id oznacza adres drużyny, przekierowani jesteśmy do komponentu TeamDetail, który generuje stronę ze szczegółami o drużynie.
- App.css – zawiera arkusz stylów dla aplikacji.

- Header.js – komponent generowany cały czas, stanowiący nagłówek strony, wraz z linkiem do strony głównej.

CSGO Teams

- Teams.js – komponent pobierający dane o drużynach z TeamsAPI:

```
function Teams() {  
  
  useEffect(() => {  
    fetchItems();  
  }, []);  
  
  const [items, setItems] = useState([]);  
  
  const fetchItems = async () => {  
    const data = await fetch('http://127.0.0.1:8000/teams/');  
  
    const items = await data.json();  
  
    setItems(items);  
  };  
};
```

oraz generujący zakładki z dostępnymi drużynami na stronie

```
return (  
  <div className="team">  
    {items.map(item => (  
      <div className="entry" key={item.id}>  
        <div className="photo-prev"><img src={imgs[item.id-1]} alt="team-photo" className="preview"/></div>  
        <h2 className="team-desc">  
          <Link to={` /team/${item.id}`}>{item.name}</Link>  
        </h2>  
        <div className="clearStyle"></div>  
      </div>  
    ))}  
  </div>  
);
```



- TeamDetail.js – komponent pobierający dane o poszczególnej drużynie i generujący podstronę jej poświęconą. Adres takiej podstrony, to /team/id, gdzie id oznacza id danej drużyny
- imgs.js – plik pobierający i zwracający zdjęcia logo drużyn załączonych wraz z projektem.