

Typescript

SPIS TREŚCI

Spis treści.....	1
Cel zajęć.....	1
Rozpoczęcie.....	1
Uwaga.....	1
Wymagania.....	2
Instalacja Node.js.....	2
Konfiguracja projektu.....	2
Dynamicznie podłączany styl.....	3
Dynamicznie tworzony obszar z linkami.....	4
Commit projektu do GIT.....	5
Podsumowanie.....	5

CEL ZAJĘĆ

Celem głównym zajęć jest zdobycie następujących umiejętności:

- konfiguracja środowiska do programowania i kompilacji aplikacji z wykorzystaniem TypeScript
- budowa skryptów z wykorzystaniem języka TypeScript

W praktycznym wymiarze uczestnicy zmodyfikują witrynę z LAB A (CSS Garden) do postaci single-page application (SPA), w której style będą wczytywane dynamicznie za pomocą skryptów napisanych w TypeScript.

ROZPOCZĘCIE

Rozpoczęcie zajęć. Powtórzenie podstaw składni TypeScript.

Wejściówka?

UWAGA

Ten dokument aktywnie wykorzystuje niestandardowe właściwości. Podobnie jak w LAB A wejdź do **Plik** -> **Informacje** -> **Właściwości** -> **Właściwości zaawansowane** -> **Niestandardowe** i zaktualizuj pola. Następnie uruchom ten dokument ponownie lub **Ctrl+A** -> **F9**.

WYMAGANIA

W ramach LAB E zmodyfikowany zostanie kod z LAB A:

- w przeciwieństwie do pierwotnej postaci, witryna ma teraz składać się z jednej strony index.html oraz wielu stylów CSS;
- do strony podpięty jest skrypt JS z zewnętrznego pliku budowanego za pomocą webpacka
- w skrypcie napisanym w TS przechowywany jest stan aplikacji:
 - nazwa bieżącego stylu i jego plik;
 - słownik dostępnych stylów i ich plików;
- fragment strony zawierający linki do stron w innym stylu zostają zastąpione przez linki z wywołaniami aplikacji w TS;
- po wywołaniu aplikacji z linka, z DOM usunięte zostaje odwołanie do starego stylu CSS, a dodane zostaje odwołanie do stylu CSS powiązanego w słowniku aplikacji z wybranym stylem

Prowadzący omówi powyższe wymagania. Upewnij się, czy wszystko rozumiesz.

Tu umieść swoje notatki:

...notatki...

INSTALACJA NODE.JS

Wejdź na stronę <https://nodejs.org/en/download/current>. Pobierz wersję Current -> Windows Binary (.zip) -> 64-bit. Rozpakuj archiwum do I:\node. Dodaj ten folder do zmiennej środowiskowej Path użytkownika.

Uruchom nowe okno wiersza poleceń. Wykonaj polecenie `npm -v`. Powinno zadziałać – udało się zainstalować NODE i NPM lokalnie dla użytkownika. Jeśli nie działa w Twoim terminalu, spróbuj w innym (CMD, PowerShell) albo wyloguj się i zaloguj ponownie (<https://ispot.link/hyttiooa>).

KONFIGURACJA PROJEKTU

Powiel projekt z LAB A. Zaktualizuj strukturę projektu:

- pozostaw pojedynczy plik HTML
- pliki CSS umieść w osobnym podkatalogu
- utwórz plik `script.ts` o zawartości:

```
const msg: string = "Hello!";  
alert(msg);
```

Do pliku HTML podłącz plik `dist/script.js`:

```
<head>  
...  
  <script src="dist/script.js" defer></script>  
</head>
```

Zainicjalizuj pakiet NPM i zainstaluj paczkę mix oraz zależności:

```
npm install --save-dev laravel-mix ts-loader typescript  
npx tsc --init
```

Utwórz plik `webpack.mix.js`:

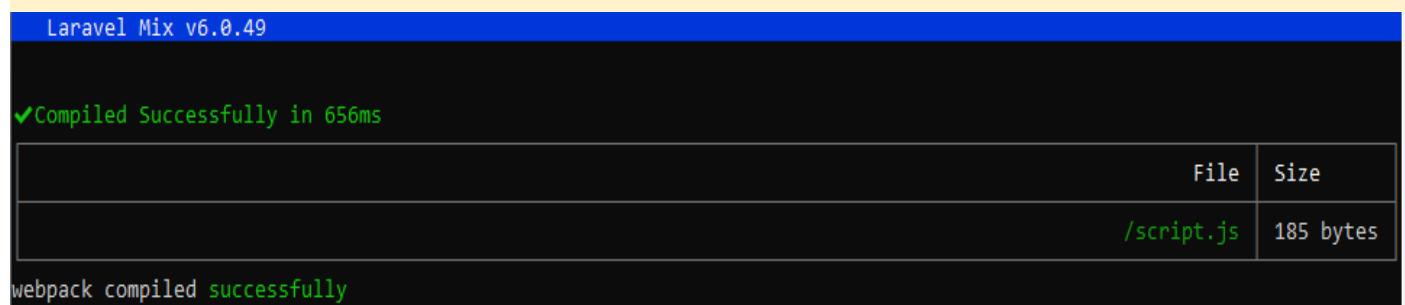
```
let mix = require('laravel-mix');  
mix.ts('script.ts', 'dist').setPublicPath('dist');
```

Włącz kompilację ciągłą:

```
npx mix watch
```

Uruchom stronę w przeglądarce.

Wstaw zrzut ekranu komunikatu o sukcesie kompilacji:



Wstaw zrzut ekranu zbudowanej strony:

Karciane Turnieje Pokémon

Wszystko, co musisz wiedzieć o karcianych turniejach Pokémon

Czym są karciane turnieje Pokémon?

Karciane Turnieje Pokémon to wydarzenia, podczas których gracze rywalizują ze sobą, używając rozmaitych talii. Celem jest zwycięstwo nad innymi graczami i zdobycie tytułu mistrza.

Rodzaje turniejów

W turniejach Pokémon możemy wyróżnić różne rodzaje zawodów, takie jak:

- Turnieje regionalne
- Mistrzostwa krajowe
- Mistrzostwa świata

Więcej informacji o turniejach Pokémon znajdziesz na naszej stronie.

[Styl: styl1.css](#) [Styl: styl2.css](#) [Styl: styl3.css](#) [Styl: styl4.css](#)

Tu przepraszam za tego screena, ale robiłem go jeszcze na zajęciach i nie wiem czemu taki wyszedł brzydki :((((

Punkty:	0	1
---------	---	---

DYNAMICZNIE PODŁĄCZANY STYL

Wstaw zrzut ekranu kodu odpowiedzialnego za dynamiczne podłączanie stylu CSS:

```

42 function changeStyle(newStyle: string): void {
43     const linkElement = document.getElementById("theme-link") as HTMLLinkElement | null;
44
45     // Walidacja nowego stylu
46     if (!appState.styles[newStyle]) {
47         console.error('Nie znaleziono stylu: ${newStyle}');
48         return;
49     }
50
51     if (linkElement) {
52         // Zmiana atrybutu href w istniejącym linku
53         linkElement.href = appState.styles[newStyle];
54         console.log('Zmieniono styl na: ${newStyle}');
55     } else {
56         console.error("Nie znaleziono elementu <link> z ID 'theme-link'.");
57     }
58
59     // Aktualizacja bieżącego stylu w stanie aplikacji
60     appState.currentStyle = newStyle;
61 }

```

Ten już ładny :D

Punkty:	0	1
---------	---	---

DYNAMICZNIE TWORZONY OBSZAR Z LINKAMI

Wstaw zrzut ekranu kodu odpowiedzialnego za dynamiczne generowanie linków do stylów:

```

16 ~ function generateStyleLinks(): void {
17     const container : HTMLElement | null = document.getElementById("style-buttons");
18     ~ if (!container) {
19         console.error("Nie znaleziono kontenera dla przycisków stylu.");
20         return;
21     }
22
23     // Czyszczenie istniejących dzieci
24     container.innerHTML = "";
25
26     // Tworzenie nowych linków dla każdego stylu
27     ~ Object.keys(appState.styles).forEach((styleName : string) : void => {
28         const link : HTMLAnchorElement = document.createElement("a");
29         link.href = "#";
30         link.textContent = `Styl: ${styleName}`;
31         link.style.marginRight = "10px";
32         ~ link.onclick = (event : MouseEvent) : void => {
33             event.preventDefault(); // Zapobieganie przeładowaniu strony
34             changeStyle(styleName);
35         };
36
37         container.appendChild(link);
38     });
39 }

```

Wstaw zrzut ekranu obszaru linków na stronie:

[Styl: styl1.css](#) [Styl: styl2.css](#) [Styl: styl3.css](#)

Wstaw zrzut ekranu obszaru linków po dodaniu kolejnego stylu do tablicy stylów:

[Styl: styl1.css](#) [Styl: styl2.css](#) [Styl: styl3.css](#) [Styl: styl4.css](#)

Punkty:

0

1

Wstaw zrzut ekranu kodu odpowiedzialnego za zmianę stylu po kliknięciu na link:

```

43 function changeStyle(newStyle: string): void {
44     const linkElement = document.getElementById("theme-link") as HTMLLinkElement | null;
45
46     // Walidacja nowego stylu
47     if (!appState.styles[newStyle]) {
48         console.error(`Nie znaleziono stylu: ${newStyle}`);
49         return;
50     }
51
52     if (linkElement) {
53         // Zmiana atrybutu href w istniejącym linku
54         linkElement.href = appState.styles[newStyle];
55         console.log(`Zmieniono styl na: ${newStyle}`);
56     } else {
57         console.error("Nie znaleziono elementu <link> z ID 'theme-link'.");
58     }
59
60     // Aktualizacja bieżącego stylu w stanie aplikacji
61     appState.currentStyle = newStyle;
62 }

```

Wstaw zrzut ekranu strony po kliknięciu na link zmiany stylu:

KARCIANE TURNIEJE POKÉMON

Wszystko, co musisz wiedzieć o karcianych turniejach Pokémon

Czym są karciane turnieje Pokémon?

Karciane Turnieje Pokémon to wydarzenia, podczas których gracze rywalizują ze sobą, używając rozmaitych talii. Celem jest zwycięstwo nad innymi graczami i zdobycie tytułu mistrza.

Rodzaje turniejów

W turniejach Pokémon możemy wyróżnić różne rodzaje zawodów, takie jak:

- Turnieje regionalne
- Mistrzostwa krajowe
- Mistrzostwa świata

Więcej informacji o turniejach Pokémon znajdziesz na naszej stronie.

[Styl: styl1.css](#) [Styl: styl2.css](#) [Styl: styl3.css](#) [Styl: styl4.css](#)

Wstaw rzuty ekranu fragmentu kodu strony (narzędzia developerskie) z podłączonymi linkami przed i po zmianie stylu:

```
<link id="theme-link" rel="stylesheet" href="style_all/styl1.css">
```

Próba zmiany stylu na: styl2.css

Zmieniono styl na: styl2.css

Dodałem console.logi w funkcji changeStyle bo do końca nie wiedziałem o co chodzi.

Punkty:	0	1
---------	---	---

COMMIT PROJEKTU DO GIT

Zacommituj i pushnij swoje rozwiązanie do repozytorium GIT.

Upewnij się, czy wszystko dobrze się wysłało. Jeśli tak, to z poziomu przeglądarki utwórz branch o nazwie `lab-e` na podstawie głównej gałęzi kodu.

Podaj link do brancha `lab-e` w swoim repozytorium:

...link, np. <https://github.com/inazwisko/ai1-lab/tree/lab-e...>

PODSUMOWANIE

W kilku zdaniach podsumuj zdobyte podczas tego laboratorium umiejętności.

...podsumowanie...

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.