AI1 **LAB E**	Jasnos Gracjan Album 50992 Grupa 4	**Wersja 1**

TypeScript

Spis treści

Spis treści 1
Cel zajęć 1
Rozpoczęcie 1
Uwaga 1
Wymagania 2
Badanie API ..... **Error! Bookmark not defined.**
Implementacja ..... **Error! Bookmark not defined.**
Commit projektu do GIT 7
Podsumowanie 7

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 TL2 (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 styli 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.

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"></script>
</head>
```

Zainicjalizuj pakiet NPM i zainstaluj paczkę mix oraz zależności:

```
npm install --save-dev mix ts-loader typescript
```

```
npx tsc -init
```

Utwórz plik `webpack.mix.js`:

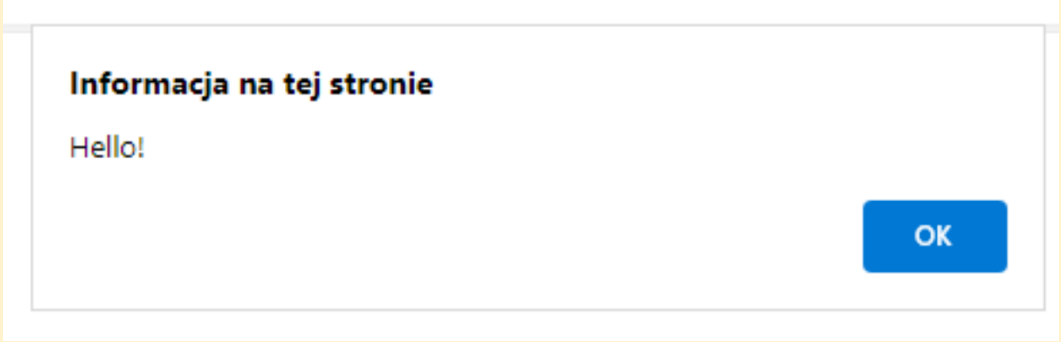
```
let mix = require('laravel-mix');
mix.ts('script.ts', 'dist').setPublicPath('dist');
```

Włącz kompilację ciągłą:

```
npx mix watch
```

Uruchom stronę w przeglądarce.

Wstaw zrzut ekranu komunikatu o sukcesie kompilacji:



Wstaw zrzut ekranu zbudowanej strony:

Punkty:	0	1

DYNAMICZNIE PODŁĄCZANY STYL

Wstaw zrzut ekranu kodu odpowiedzialnego za dynamiczne podłączanie stylu CSS:

```
function changeStyle(styleKey: string) {
    const head = document.head;

    const link = document.createElement('link');
    link.id = 'app-style';
    link.rel = 'stylesheet';
    link.href = appState.styles[styleKey];

    head.appendChild(link);

    appState.currentStyle = styleKey;
}
```

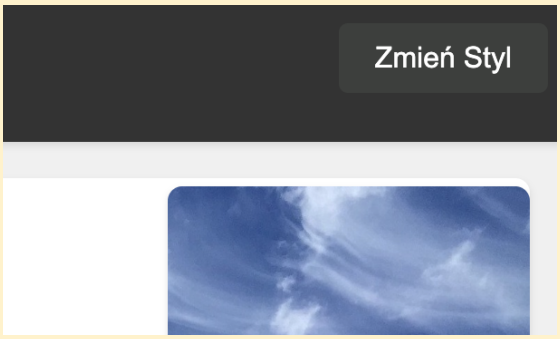
Punkty:	0	1

DYNAMICZNIE TWORZONY OBSZAR Z LINKAMI

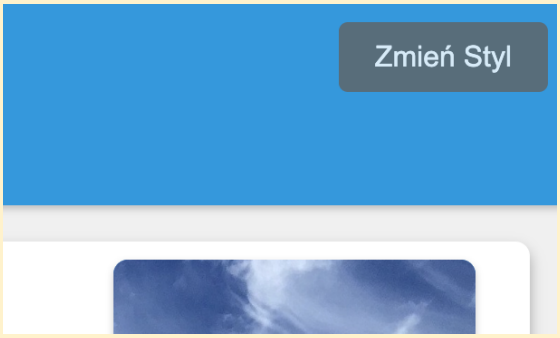
Wstaw zrzut ekranu kodu odpowiedzialnego za dynamiczne generowanie linków do styli:

```
function changeStyle(styleKey: string) {
    const head = document.head;

    const previousStyle = document.getElementById('app-style');
    if (previousStyle) {
        head.removeChild(previousStyle);
    }

    const link = document.createElement('link');
    link.id = 'app-style';
    link.rel = 'stylesheet';
    link.href = appState.styles[styleKey];

    head.appendChild(link);

    appState.currentStyle = styleKey;
}
```

Wstaw zrzut ekranu obszaru linków na stronie:



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



Punkty:	0	1

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

```
function handleChangeStyleButtonClick() {
    console.log('Button clicked!');

    let nextStyle;

    switch (appState.currentStyle) {
        case 'style1':
            nextStyle = 'style2';
            break;
        case 'style2':
            nextStyle = 'style3';
            break;
        case 'style3':
            nextStyle = 'style1';
            break;
        default:
            nextStyle = 'style1';
            break;
    }

    changeStyle(nextStyle);
}
```

Wstaw zrzut ekranu strony po kliknięciu na link zmiany stylu:

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

```
<head> == $0
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Samoloty</title>
  <link rel="stylesheet" href="styles/style1.css">
  <script src="dist/script.js"></script>
  <script defer type="module" src="script.js"></script>
  <script defer type="module" src="script.ts"></script>
  ▶<style>…</style>
  <link id="app-style" rel="stylesheet" href="styles/style1.css">
</head>
```

```
<head> == $0
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Samoloty</title>
  <link rel="stylesheet" href="styles/style1.css">
  <script src="dist/script.js"></script>
  <script defer type="module" src="script.js"></script>
  <script defer type="module" src="script.ts"></script>
  ▶<style>…</style>
  <link id="app-style" rel="stylesheet" href="styles/style2.css">
</head>
```

```
<head> == $0
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Samoloty</title>
  <link rel="stylesheet" href="styles/style1.css">
  <script src="dist/script.js"></script>
  <script defer type="module" src="script.js"></script>
  <script defer type="module" src="script.ts"></script>
  ▶<style>…</style>
  <link id="app-style" rel="stylesheet" href="styles/style3.css">
</head>
```

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-c` na podstawie głównej gałęzi kodu.

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

…link, np. https://github.com/gracjanjasnos/main/tree/main/lab05

PODSUMOWANIE

W kilku zdaniach podsumuj zdobyte podczas tego laboratorium umiejętności.

Podczas laboratorium nauczyłem się modyfikować stronę internetową, aby korzystała z wielu plików stylów CSS, a także z zewnętrznego skryptu napisanego w TypeScript. Zaimplementowałem funkcjonalność przechowywania stanu aplikacji w języku TypeScript, zarządzając bieżącym stylem oraz słownikiem dostępnych stylów i ich plików.

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.