SKRYPT 27

1. Przetestuj kod z prezentacji - strona 18
2. Przetestuj kod z prezentacji - strona 22
3. Przygotuj aplikację TS (zainstaluj TS w projekcie npm install --save-dev typescript- jeśli konieczne)
4. interface Car {
5. readonly id: number;
6. brand: string;
7. }
9. const myCar: Car = { id: 101, brand: "Toyota" };
10. myCar.brand = "Honda"; // OK
11. // myCar.id = 102; // Błąd: id jest readonly
13. Przetestuj kod z prezentacji – strona 58

----------------------------------------------------------------

1. Przygotuj aplikację klient-serwer napisaną w Node.js z użyciem TypeScript, która wystawia endpoint REST-owy i wykorzystuje interfejsy TypeScript w logice serwera.

//////////////////////////// app.ts//////////////////////////

Ma wystawiać trzy endpointy z wykorzystaniem **Express.js.** Zadeklaruj interfejs zawierający:

**id: number; name: string; email: string; isActive: boolean;**

utwórz tablicę 10 obiektów na podstawie interfejsu **User.**

**Endpointy (funkcje do endpointów):**

- pierwszy zwraca wszystkie pola obiektu **user (**function getAllUsers())

- kolejny zwraca użytkownika po ID (function getUserById(id: number))

- ostatni niech wypisuje wszystkie czy użytkownik o danym ID jest aktywny (function getIsActivById(id:number))

**Przykładowa kolejność w pliku:**

//Import modułu(ów)

// Definicja interfejsu

// Przykładowe dane użytkowników (tablica obiektów)

// Funkcja pobierająca użytkownika po ID

// Funkcja zwracająca wszystkich użytkowników

// Tworzenie aplikacji Express

const app: Application = express();

const PORT = 3000;

// Middleware

app.use(express.json());

// Endpointy

// Uruchomienie serwera

///////////////////////////////app.js/////////////////////////

Kompilowany (transpirowany) plik app.ts

////////////////////////////client.html//////////////////////

Wysyła żądania fetch() i wyświetla JSON-a

/////////////////////////// tsconfig.json /////////////////////

Konfiguracja ts

{

    "compilerOptions": {

      "target": "ES6",

      "module": "commonjs",

      "strict": true

    }

  }

/////////////////app.ts //////////////////

#Plik TypeScript serwera

///////////////////////////package.json ///////////

Plik konfiguracyjny Node.js