**Type Script**

<https://www.typescriptlang.org/>

**init TypeScript**

1. Ініціалізуємо ts

npm i –g typescript

npm i –D typescript

tsc --init

**tsc config:**

**general**:

- **target: ‘es6’** – вказуємо тип тип es в який компілюватиметься код

- **allowJs**: true – компілює не лише ts, а і js

* **rootDir** – вказуємо деректорію з якої ts бартиме файли длякомпіляції
* **outDir** – вказуємо деректорію для вигрузки скомпільованого js
* **lib: []** – налантовуємо роботу ts з api (в тому числі і браузерне api – dom)
* **sourceMap**
* **jsx: ‘react’** – для компіляції реакта по правилам реакта
* **removeComments: true** – видаляє в скомпільованому коді всі коментарі
* **noEmitOnError: true** – при будь-якій помилці, новий файд не скомпілюється

**strictConfig:**

* **strict: true** – видаватиме помилки на не відповідність типів
* **noImplicitAny: true** – не довзволяє компілювати код, якщо в нас є невизначений тип в переданих функції параметрах
* **strictNullChecks: true** – виводить помилку, якщо перед використанням змінної в неї не буде перевірки на наявність

**additionChecker:**

* **noUnusedLocals: true** – виводить попередження про невикористання ініціалізованої змінної, локальної, глобально можна
* **noImplicitReturns: true** – якщо не всі умови ведуть в функції до повернення результату і не вказано, що функція може не повертати нічого, то виведеться помилка

**Module Resolution Options:**

* **esModuleInterop**: true – покращує співпрацю із babel
* **allowSyntheticDefaultImport:** true –динамічний імпорт
* **resolveJsonModule:** true –взаємодія із json, як із об’єктом

**tsc command:**

tsc – компіляція

tsc --watch – спостереження і компіляція по змінам

**Libs:**

-- Class-transformer – допомагає об’єднувати поля із json і класи з методами, що працюють із цими полями в екземпляр класу

--class-validator – використання великох к-сті дикораторів заготовок