How to Install typescript in one project

**dependencies:**

npm i typescript -D

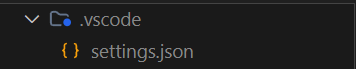
npm i ts-node -D

npx tsc --init

How to configure code runner for run typescript

Create folder “.vscode”

.vscode -> folder with config of project



In “settings.json”

Paste this config…

{

  "code-runner.executorMap": {

    "typescript": "npx ts-node --files"

  }

}

How to configure ESlint with typescript

npm i eslint -d

npm i @typescript-eslint/eslint-plugin @typescript-eslint/parser -D

Create archive “.eslintrc.js”

Paste this code...

module.exports = {

env: {

browser: true,

es6: true,

node: true,

},

extends: [

'eslint:recommended',

'plugin:@typescript-eslint/eslint-recommended',

'plugin:@typescript-eslint/recommended',

'plugin:prettier/recommended',

],

globals: {

Atomics: 'readonly',

SharedArrayBuffer: 'readonly',

},

parser: '@typescript-eslint/parser',

parserOptions: {

ecmaVersion: 11,

sourceType: 'module',

},

plugins: ['@typescript-eslint'],

rules: {},

};

How to configure Prettier with ESlint

npm i prettier eslint-config-prettier eslint-plugin-prettier -D

Create archive “prettierrc.js”

Paste this config...

module.exports = {

  semi: true,

  singleQuote: false,

  trailingComma: "all",

  printWidth: 80,

  tabWidth: 2,

}

Anottations

Tsconfig.json anottations

CompilerOptions: Opções de compilação

**Target:** versão do ecma utilizada.

**Lib:** com --lib você pode especificar uma lista de grupos de declaração de API integrados que você pode optar por incluir em seu projeto. Por exemplo, se você espera que seu tempo de execução tenha suporte para Map, Set e Promise (por exemplo, a maioria dos navegadores atuais), basta incluir --lib es2015.collection,es2015.promise. Da mesma forma, você pode excluir declarações que não deseja incluir em seu projeto, por exemplo, DOM, se estiver trabalhando em um projeto de node usando --lib es5,es6.

[O que a opção "lib" do tsconfig faz?](https://stackoverflow.com/questions/39303385/what-does-the-tsconfig-option-lib-do)

**Module:** tipo de exportação (common ou modules).

**outDir:** pasta de saída do Código javascript (depois de transpilado).

**Strict:** resumidamente força o desenvolvedor a programar direito.

Include: Pasta de entrada dos arquvios TS

TS with webpack for frontend

Install

npm i ts-loader webpack webpack-cli -D

create archive

**webpack.config.js**



Paste this config…

const path = require("path");

module.exports = {

  mode: "development",

  entry: "./src/A0018-webpack/index.ts",

  module: {

    rules: [

      {

        test: /\.tsx?$/,

        use: "ts-loader",

        exclude: /node\_modules/,

      },

    ],

  },

  resolve: {

    extensions: [".tsx", ".ts", ".js"],

  },

  output: {

    filename: "bundle.js",

    path: path.resolve(\_\_dirname, "dist", "assets", "js"),

  },

  devtool: "source-map",

};

[webpack typescript docs](https://webpack.js.org/guides/typescript/)

for run webpack

paste on terminal

npx webpack

for run webpack in time real

npx webpack -w