Configuring TypeScript

CLI Commands

tsc --init: Creates tsconfig.json file which can be used to change different settings.

tsc: Compiles TypeScript files and outputs Javascript with .js extension.

If the file name is not specified in the command, it will compile the whole directory.

tsc -w: Instead of compiling TS once, listenes for changes and automatically recompiles once file is saved.

If the file name is not specified in the command, it will watch the whole directory. (w stands for watch)

tsc -v: Checks version of TS.

tsc --noEmitOnError: Do not emit compiler output files if error is occurred.

tsconfig Settings

Files: Contains a list of files which will be compiled when the tsc command is run.

```
{
 "compilerOptions": {},
 "files": [
   "file1.ts",
   "file2.ts",
 ]
}
```

Include, Exclude

• With Files, it was specified which files were compiled with the tsc command. But adding files to the list one by one can be time consuming.

Include: Specifies an array of filenames or patterns to include in the program. These filenames are resolved relative to the directory containing the tsconfig.json file.

```
{
 "include": ["src/**/*",
"tests/**/*"]
}
```

Configuring TypeScript

· If files are specified then includes option will default to nothing.

Exlude: Works together with include. For example, compile.ts and dontCompile.ts files are in the src directory. We can use the following settings:

```
{
 "include": ["src"]
}
```

```
{
 "exclude":
["src/dontCompile.ts"]
}
```

By default, node_modules is included in the exclude list and we have to add it when we edit the list manually.

Out Dir: Specifies where TS should emit the compiled JS files. By default it emits files alongside TS files.

Common folder name for compiled JS files is dist.

Target: JS version that TS compiles into.

Strict: Enables all strict type-checking options. For example, one of the options is noImplicitAny. Since the use of Any is contrary to the type definition logic in the logic of TS, this results in stronger guarantee of program correctness.

Lib: Specify a set of bundled library declaration files.

Some examples are:

· Document library which is useful for accessing dom elements. Although we can access the document in JavaScript, if we want to be able to access its type-defined form, we need to define it in the lib array.

· To access additional APIs that are available in different versions of ES. By default it will use same version as defined in target.

Type declaration files end with .d.ts. For example name of the document file is lib.dom.d.ts