100 XP

Exercise - Compile modules

5 minutes

Modules import one another using a module loader. At runtime, the module loader locates and executes all dependencies of a module before executing it. Depending on the module target that you specify during compilation, the compiler will generate appropriate code for Node.js (CommonJS), require.js (AMD), UMD , SystemJS , or ECMAScript 2015 native modules (ES6) module-loading systems.

To compile modules, specify a `--module` target on the command line or in the **tsconfig.json** file for the project.

Continue your project from the previous exercise.

1. Open the terminal and compile the **main.ts** module for Node.js by typing the following command:

   Bash

   ```
   tsc --module commonjs main.ts
   ```

2. The compiler follows `import` statements to compile all dependent files. Notice that when **main.ts** is compiled, each module will become a separate **.js** file.

3. Type `node main` to test the file.

Running modules from a web page

If you want to instead compile the TypeScript file for ES6 for use in a web browser, type the following command:

Bash

```
tsc --module es6 main.ts
```

To run a module from a web page, remember to set the `type` option to `"module"`:

HTML

```
<script type="module" src=".\main.js"></script>
```

Exercise solution

To see the solution to this exercise, clone the repository by entering the following at the command prompt.

Bash

```
git clone https://github.com/MicrosoftDocs/mslearn-typescript
cd mslearn-typescript/code/module-07/module07-exercise-01-end
code .
```

Next unit: Exercise - Access external type libraries

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆