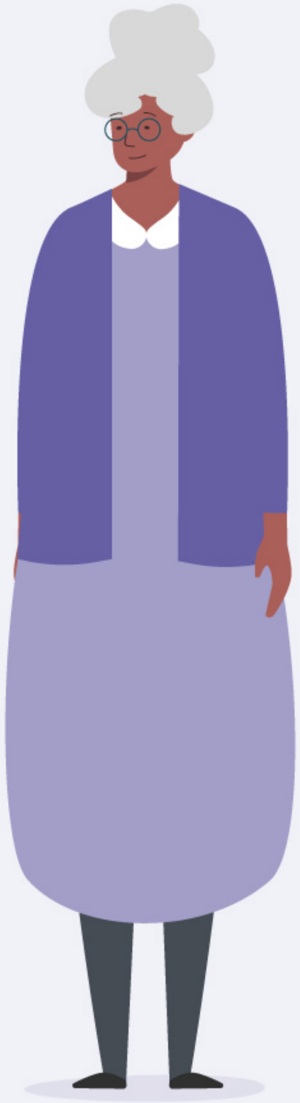
Configuring, Compiling, and Debugging Typescript Projects

COURSE OVERVIEW

Daniel Stern
CODE WHISPERER

@danieljackstern

A Big Enterprise Application Scenario



You are the lead developer at **Helpful Technology Solutions**, a company that makes software which helps retirees manage their finances.

Your manager asks you to create a new web application to replace the ageing desktop version of the product.

Rolling out one feature at a time, management expects dozens of developers to be working on this project.

The online application needs to be reliable, bug-free, and ready for a big team to work on.

What do you?

Why TypeScript?

"Go then, there are other worlds than these."

Stephen King

Why TypeScript?

Compile-time type checking prevents errors

Classes, types and interfaces assist in collaborative development

Numerous time saving features for advanced users (see conclusion for links)

There are myriad reasons why TypeScript might make a good choice for a given project. These are just a few.

A Look at the Completed Application

Demo

Application is a straightforward program for asking and answering questions

Consists of several TypeScript files (server, app) and their JavaScript equivalents

We will build the application over the course of the project

Finished application available at: `https://github.com/danielstern/compiling-typescript`

Course Roadmap

Set up a development environment for TypeScript and get to “Hello World”

Write our own compiler configuration and apply it to our workflow

Install Type Declaration files for critical libraries

Debug our application in Chrome and VS Code

Configuring, Compiling, and Debugging TypeScript Projects

SCAFFOLDING AN ENVIRONMENT FOR TYPESCRIPT COMPILATION

Daniel Stern
CODE WHISPERER

@danieljackstern

Learning Objectives

Select and install appropriate TypeScript compilation software

Create a bare-bones typescript configuration (`tsconfig`) file – more settings will be added in next module

Scaffold and test web application built around TypeScript compilation pipeline (more advanced features added as we go)

Installing and Configuring Visual Studio Code

Why Visual Studio Code?

> “What good is the warmth of summer, without the cold of winter to give it sweetness.”

John Steinbeck

Why Visual Studio Code?

Visual Studio is one of many tools that can help us compile TypeScript. Since it’s a reliable application made by an industry leader, it’s a reasonable choice.

Familiar

Many developers on a large team will already be acquainted with VS

Supported

Microsoft supports VS and has a big incentive to keep it secure and updated

Helps with TypeScript

Code hints, compilation and debugging are all available

Demo

Identify appropriate version of Visual Studio Code to install

Install Visual Studio Code and packages for compiling TypeScript

Create a blank project directory

Executing the TypeScript Compiler

Executing the TypeScript Compiler

Transforms specified TypeScript files into ES5 JavaScript files

Transformation is necessary before app can be used on the web

Can be configured to execute automatically (coming up later)

Demo

Install `tsc` (TypeScript Compiler), verify correct installation and version

- `tsc` included in TypeScript package

Create rudimentary TypeScript file

- Contains only basic code
- More advanced TypeScript code will be added in later modules

Use command line to convert TypeScript file into JavaScript file

Creating a TypeScript Configuration (.tsconfig) File

What Is .tsconfig?

Defines which TypeScript files should be compiled and the name of the resulting JavaScript files

Specifies which TypeScript features to use when compiling this particular project

Demo

Create a .tsconfig file in the project directory

Define compiler options:

- Specify ES5 directory
- Source maps

Scaffolding the Demo Application with Express and Visual Studio

What Are the Basic Components of a Node / Typescript Application?

Even relatively simple applications require many files.

It is easy to maintain files that are separated logically and simple to understand.

`express.ts`

Instructs Node on how to serve the application.

`index.html`

Aggregates references to all JS files, CSS files, etc.

`main.ts`

Contains our application logic, written in TypeScript.

Demo

Install express via `npm`

Create `index.html`

Create basic `express.ts` file and compile

Serve app to browser using express

Module Summary

Summary

VSCode is suitable for TypeScript development by large teams

`tsc` is used to compile TypeScript (development) into JavaScript (production)

Details of compilation specified in `tsconfig.json` configuration file

Coming Up in the Next Module

Take a deep dive into the available compiler options

Advanced `tsconfig.json` setup

Using the compiler in watch mode

Configuring project references