COMPSCI497S-SEC01 ST- Scalable Web Systems Spring 2021

Week 3, Assignment 3, Building a Microservice

Overview

The purpose of this assignment is to have you explore or further explore the following:

- The TypeScript programming language to construct an HTTP server API
- The Express web application framework

As we move towards understanding the design and implementation of a microservice it is important to recognize two fundamental roles a microservice plays:

1. A client requesting information from other services that are either part of the same application or services built as part of other applications that expose an API to the outside world.
2. A server servicing requests from other services (HTTP clients).

The first role was explored in the first assignment. The second role was explored in the second assignment. In this assignment, we want to focus on using a scalable language (e.g., TypeScript) to implement a microservice that will receive incoming HTTP requests, communicate to a 3rd-party service, do something interesting with the results, and respond back to the HTTP client. Your microservice should do something very specific, that is, it should not be a full-blown application. You should imagine it existing in the context of a larger application where the incoming requests are similar to a call to a library. You might find it useful to look at this list of 3rd-party APIs that you can use for this application or use what you had in assignment 1.

Prerequisites

The prerequisites for completing this assignment are the following:

- Install Node.js on your host platform (this should also install NPM). You should have already done this.
- Install TypeScript on your host platform with NPM (use NPM to install TypeScript). You should have already done this.
- Install Express on your host platform with NPM (use NPM to install Express).

The assumption is that you have experience with Node.js and NPM and you are familiar with the basics of this environment. I highly recommend the VSCode editor to complete this and future assignments. You might also want to review the VSCode support for TypeScript. Indeed, VSCode itself is written in TypeScript.

Instructions

In this assignment you are to write a Node.js/Express TypeScript program that will accept incoming requests from an HTTP client, make a request to a 3rd-party API, do something with the response, and then respond back to the original request. This application does not need to be complicated. In fact, we prefer that you deliberately keep it simple and focus on how TypeScript plays a role in writing HTTP clients and servers with the Express framework and submit a polished (and well documented comments in the code) implementation that demonstrates your understanding on how this all works. Leave the complicated implementation to the semester project! To summarize the requirements:

1. Write a TypeScript program that will receive incoming HTTP requests from an HTTP client.
2. The application should contact at least one 3rd-party API over HTTP and receive an HTTP response (likely in JSON format). You may use more than one 3rd-party API end-point.
3. The application should do something with the response from the 3rd-party. It can really be just about anything, but it must do something other than respond with the results of the request.
4. Respond to the client in JSON format.
5. Your application must provide at least 3 HTTP end-points that an HTTP client can call.

You must also include a README.md file in markdown format that briefly describes how your program works and struggles that you encountered.

Part of using TypeScript is becoming familiar with DefinitelyTyped. Please note your experiences with using this with your experience in TypeScript as it relates to the Express framework and any other library that you use to construct your application.

To test your HTTP server you can use an HTTP client of your choice. Here are a few suggestions:

- curl - A command line HTTP client. Simple to use and easy to script.
- PostMan - A more robust and graphical tool. Easy to play around with after you get used to it.
- RestMan - a chrome browser extension that is super simple (between curl and PostMan).
- Or simply a browser.

Additional Notes

Your HTTP server must respond to GET requests only. You can use GET requests to send data through query parameters or URL paths, but for now, do not worry about POST requests - I am deliberate in keeping this straightforward. Your HTTP may use any type of request required by the 3rd-party API (I would still recommend to keep this simple).

Your HTTP server must accept some type of data through query parameters and respond dynamically to those requests by performing some type of computation.

Your HTTP server must maintain data in memory only. Do not use a database or any external 3rd party data storage. The point here is to focus - not elaborate. We are looking to keep everyone grounded and clear on what it means to implement a microservice using TypeScript with Express.

We would prefer to see a clearly constructed application that satisfies the above rather than adding everything under the kitchen sink! If you want to explore more - then do it within these parameters.

Lastly, although it is tempting to move forward quickly it is even more important to understand more deeply the language, frameworks, and APIs that you are using. So, if you need more I suggest that you become more deeply acquainted with TypeScript or a specific aspect of Express so you are more prepared for the semester project. You can investigate additional APIs and how they deliver information. Do not add anything outside of the assignment parameters.

Grading

You will be graded on the following scale:

- 4 - the submitted program demonstrates proper usage of the TypeScript programming language and satisfies the requirements of the assignment.
- 3 - the submitted program demonstrates proper usage of the TypeScript programming language, but not all of the requirements have been satisfied.
- 2 - the submitted program partially demonstrates proper usage of the TypeScript programming language, some or all of the requirements have been satisfied.
- 1 - the submitted program does not demonstrate proper usage of the TypeScript programming language, some or all of the requirements have been satisfied.
- 0 - no submission

Submission

Please upload a zip archive of your work to this assignment in Moodle for review by the due date. Make sure you do not upload your **node_modules** folder as this can exceed the allowable upload limit.

Submission status

Submission status	No attempt
Grading status	Not graded
Due date	Monday, March 1, 2021, 11:59 PM
Time remaining	5 days 7 hours
Last modified	-
Submission comments	Comments (0)

Add submission

You have not made a submission yet

Moodle Help for Students • Moodle Help for Instructors

You are logged in as Timothy Nguyen (Log out)
COMPSCI497S-SEC01 SP21