# Developer Survey and Coding Exercise

## Technical Expertise

What is your level of experience with the following technologies and frameworks?

Please answer each question with **1** for "no experience", **2** for "some experience" or **3** for "considerable experience".

### Languages

* C#: \_\_\_\_
* Javascript: \_\_\_\_
* Typescript: \_\_\_\_
* HTML/SCSS: \_\_\_\_

### Frameworks and Technologies

* .NET: \_\_\_\_
* .NET Core: \_\_\_\_
* Angular: \_\_\_\_
* NgRx: \_\_\_\_
* GraphQL: \_\_\_\_

## Experience and Training

Please list any computer-related training, qualifications and work experience you have.

## Exercise

The goal is to create a simple server-client "To-Do" application using .NET Core as the server and Angular as the client. The data sent between server and client will be in JSON format. In the ZIP file you received you will find the base source code with instructions on how to compile and run it.

For now, we have only created a basic server controller returning a list of to-do items and a simple client app which shows the list to the user. To-do items must be formatted in title case (the first letter of every word should be capitalized).

Feel free to add any comments that could be useful for the reviewer. In addition, unless otherwise indicated, you can use any library of your choice.

Please reply to the questions of this exercise directly in this document to submit it back for review.

### Pre-requisites

Run the following commands to initialize a local Git repo in the root of the project. You would need to have Git installed.

git init

git add .

git commit -m "Initial commit"

For every section of this exercise, please create a new commit after completing it with these commands:

git add .

git commit -m "EXERCISE\_SECTION"

For example: git commit -m "Refactor"

### Code Review

Review the existing code in the following files:

#### Server

* **src/Todos/Controllers/TodosController.cs**
  + Take note of the ToTitleCase function and its unit tests.
* **tests/Todos.Tests/Controllers/TodosControllerTests.cs**

#### Client

* **client/src/app/todo-list/todo-list.component.ts**

Please, have the following points in mind:

* Standards, good practices you know of
* Code re-usability (for example, we may need to use ToTitleCase in other places too)
* The input data could be very large (your program could be run on a machine with limited memory)

**Please list all the things you would change or improve in the code specifying the line number whenever necessary and explain the reasons.**

**What possible enhancements would you consider?**

### Refactor

Refactor the code in Controllers/TodosController.cs and todo-list/todo-list.component.ts applying the changes and improvements you suggested in the code review.

### Add Create and Delete To-Do Items

Add the ability to create and delete to-do items. User input and stored data should always be valid and safe. You may store data in a text file or in a in-memory database.

**What kind of storage would you use in a real application to store the to-do items? Explain why.**

### Add New Fields

Add a couple of new optional fields to the to-do items: due date and notes.

Create a new detail view to edit a single to-do item. From the current list, the user should be able to click one and navigate to the new view containing:

* Title
* Indication of completion
* Due date
* Notes

The user should be able to edit the fields and save or cancel the changes.

Feel free to enhance the user interface.

### Add a Store (Optional)

Using NgRx, implement a store in the client app.

**What are benefits of using a centralized store? Explain.**

**What factors would you consider when designing the data structure? Explain.**

### Additional Questions

**How would you implement translations?**

**What problems would providing internationalization and localization for many languages have?**

**How would you ensure the accessibility of the app?**

**What types of tests would you implement to ensure QA?**

**How would you implement security in your application? Explain.**

**How would you improve the application to support a high number of concurrent users? Explain.**

## Final Steps

* Run in the root folder a git command to remove all build assets: git clean -dfxi
* Zip up the source files (excluding the node\_modules, binaries and build folders)
* Make sure to include in the zip the hidden .git folder
* E-mail it back to us with this document filled out