**DT Code Test Solution Notes**

Prepared by Rohan Pacey, 18/06/2017

**Introduction**

The task was to develop a well structured solution to meet the following provided specifications:

*Develop a web application that performs the following functions:*

1. *Capture a person’s name and a number*
2. *Convert this number into words*
3. *Render this name and number (as words) as a web page.*

*This is similar in functionality to that of a cheque writing program, for example:*

*Input: John Smith  
“123.45”*

*Output: John Smith  
“ONE HUNDRED AND TWENTY-THREE DOLLARS AND FORTY-FIVE CENTS”*

Other specified requirements were that the solution:

1. implements a web service
2. implements a front end application to consume this web service
3. is built within the .NET technology stack
4. contains well documented and readable code
5. implements relevant and comprehensive Unit Tests
6. is logical and easy to understand
7. demonstrates good programming practices & principals

In the development of this solution I have attempted to display the following programming and solution design principles:

* A modular N-tier application structure promoting good separation of concerns
* Object-oriented design and the single responsibility principle
* Extensibility through logical application architecture and the use of interfaces
* Good test coverage via unit tests
* Logical, readable, well-documented code

**Instructions**

To execute the application please open the solution in Visual Studio 2017, ensure that \DT\_CodeTest.Service\index.html is set as the start page, then use the “Start Without Debugging” option from the Debug menu. No other configuration should be required to execute the application.

**Application Architecture**

The solution contains the following projects:

|  |  |
| --- | --- |
| Project | Description |
| DT.CodeTest.Common | A class library project containing supporting helper classes. In this case the CurrencyHelper and IntegerToTextHelper classes which implement the required currency to text function. |
| DT.CodeTest.TestConsole | A simple console project to facilitate testing and debugging of the DT.CodeTest.Common classes. This could be safely removed. |
| DT.CodeTest.Common.Tests | A test project containing unit tests relevant to the DT.CodeTest.Common project. |
| DT\_CodeTest.Service | An ASP.Net Web API 2 project to implement the required web service. This is implemented via \Controllers\CurrencyToTextController. The controller is dependent on the DT.CodeTest.Common.Helpers.CurrencyHelper class to implement the required business logic.  This also contains a simple HTML/jQuery client interface \Index.html to demonstrate consumption of the service from a client. |
| DT\_CodeTest.Service.Tests | A test project containing unit tests relevant to the DT\_CodeTest.Service project. |

**Technologies**

The following technologies and frameworks have been utilised in the solution:

* Visual Studio 2017 Community Edition
* .NET Framework 4.5.2
* ASP.NET Web API 2
* Microsoft Unit Test Framework
* jQuery

**Summary**

Given the limited time available to develop this solution I have endeavoured to implement a simple yet solid application which demonstrates good development practises, a logical architecture, good test coverage and a simple client interface to demonstrate the requisite service functionality and business rules.