# Introduction

Pizza Api is an application to handle pizza orders and your objective is to test as much of it as possible within 3 - 5 hours. It is written in C# 7, ASP.NET Core MVC and Entity Framework 3 and .net Core 3.1.

 Your tests **MAY**use **HttpClient** class to test the API (Already provided to you).

OR You can implement a **RestClient** class.

 There are two projects as follows:

**PizzaApi** - This project contains the API that is to be tested.

Please examine the Controller and Models folders to understand the API and Pizza classes.

**PizzaApi.Tests -** This project is where the tests you will need to implement will be located.

Within the IntegrationTests.cs class there are some test methods added for you that will require implementing as part of this coding challenge.

**You must only** write your tests in this class. It inherits from another class that will start up the API and provide several objects that will assist you in writing your tests.

# Test Objectives

The test is split into two questions.

It is suggested you spend 60% of your time on the first question and 40% on the second question:

1. Implement the 4 happy path tests that have been provided in the PizzaApi.Tests\IntegrationTests.cs class

/// Get a count of all the pizza orders with the order status "Completed" using GET PizzaOrders

/// Ensure that the assertion passes and that 385 orders are returned.

CountCompletedPizzaOrders()

/// Add a pizza order using POST PizzaOrders

/// Assert that the order was added successfully by improving on the existing assertion which should be replaced with something more appropriate.

CreatePizzaOrder()

/// Delete a Pizza Order from the system using DELETE PizzaOrders/{id}

DeletePizzaOrder()

/// The database contains a list of Historical Pizza orders

/// Use the GET PizzaOrders endpoint to retrieve the completed orders from last year.

/// Get the total price of the pizzas, grouped by month.

GetPizzaTotalPriceGroupedByMonthForLastYear()

2. Devise and implement your own set of tests for the API, these can be unhappy or happy path tests, it doesn't matter. This is a chance to show your QA skills off. Points will be awarded for tests with "sudo" code as well as fully implemented tests as long as it’s clear and understandable to persons marking the exercise

 Finally, feel free to add and use any other packages that you feel will help you better implement the tests, a few have been provided for you already:

NewtonSoft.Json

FluentAssertions

Xunit

Nunit

System.Http.Json