IoT Developer Programming Test

Document Details

Author	Matthew Turnbull, Martin Harris
Created	
File	
Distribution	Internal Only

Version	Release Date	Comments
1.1	***21/09/22***	

Copyright Notice

CONTENTS

1 INTRODUCTION

1.1 Purpose

The purpose of this programming test is to allow Integrated System Technologies to get a better understanding of your ability to program, problem solve and creative thinking.

We're focused on the quality of work you produce.

2 JS TEST

2.1 Overview

You will be given access to a project "interview.test", which uses Maven for dependency management and relies on parts of the Spring Framework. This has been setup to test various aspects that you would face during your day to day duties.

The project includes two API endpoints to get temperature data (current and previous) as well as a web view endpoint. It uses Thmyeleaf for server-side rending of views and includes Bulma for styling, Font Awesome, jQuery and Chartjs dependencies.

2.2 Rules

- You have **30 minutes** to complete this test.
- You have access to the internet, feel free to use it.
- Focus on the quality of work and not the quantity of work.
- Everything is in place for you, no new files need to be created.
- If you're unsure on anything, please ask before the test starts.
- If you finish everything before the time is up, feel free to expand the system as you wish.

2.3 Tasks

1) Make API endpoint value return as two decimal places.

   The current API endpoints return a float value to five decimal places. This needs to be changed to return as two decimal places.

   - Implement any logic to return the value as two decimal places.
   - The return value must remain a float.

2) Plot the "current temperature" on a chart using ChartJS.

   Using the web view template "index.html", create a chart, of your choosing, which plots the "current temperature" value, returned from the API endpoint.

   - Data should update a maximum of every 2 seconds
   - Only a maximum of 15 data points can be display on the chart at once.

3) Package as a jar file.

   Before the test is complete, you should try to package the project into an executable jar file.

3 JAVA TEST

3.1 Overview

You will be given access to a project “java test”. This is a pure Java project.

There is one interface called MathematicalOperations, at least one additional class will need to be created but feel free to design this however you deem fit.

3.2 Rules

- You have **30 minutes** to complete this test.
- Focus on the quality of work and not the quantity of work.
- For this please don't use any additional libraries. All code should be your own.
- For any parts you may not complete, please enter comments to show how you would approach it.

3.3 Tasks

1) The interface MathematicalOperations has a method called multiplyAndAdd.

Change this method to take two arrays of type int as its parameters.

Next, create a class, name it however you deem appropriate. Implement the method you just changed; it should perform the following:

- multiply the value in arrayA at index i, with the value in arrayB at index i
- keep a running sum total of these values
- repeat for the entire length of the arrays
- return the total sum

2)

Do the same calculation but this time remove all occurrences of the number 3 before calculating it.

3.4 Hint

A[1] MULTIPLY BY B[1] + A[2] MULTIPLY BY B[2] .....

$$\begin{aligned}[1,3,-5]\cdot[4,-2,-1] &= (1\times 4)+(3\times -2)+(-5\times -1)\\ &= 4-6+5\\ &= 3\end{aligned}$$