

# Technical Test

## Automation QA



## Table of Contents

<b>Introduction .....</b>	<b>3</b>
<b>Technical Tools.....</b>	<b>3</b>
The Tools we use:.....	3
<b>Basic Requirements:.....</b>	<b>3</b>
<b>Section 1) .....</b>	<b>4</b>
General Programming: .....	4
Question 1: Given two strings, check if they're anagrams or not .....	4
Question 2: Whats wrong with the given code below: .....	4
<b>Section 2) .....</b>	<b>5</b>
Front End automation .....	5
API Testing .....	5
<b>Appendix.....</b>	<b>7</b>
<b>Resources.....</b>	<b>8</b>

## Introduction

Welcome to the Hostelworld Technical Test. The aim of this test is to give you a taste of the work we do on a daily basis and to see if your technical abilities match that of the companies expectations.

The test comprises of three sections. Please complete as many sections as possible, keep in mind that we are looking for how you design your solution rather than getting them all completed.

## Technical Tools

For this test you can utilise any tools or technologies you see fit, **however please try and keep to the technologies already in use at Hostelworld**. Also please make sure you document what technology you used and anything we would need to get the code running.

### The Tools we use:

For Front End Testing we write all of our tests using:

- Java or Groovy backed with WebDriver

And for our API tests we use:

- Java with Retrofit

Remember we're looking at how you implement the solution not how fast or how much you've completed....quality over quantity!!!

## Basic Requirements:

For any code you write please abide by the following principles / requirements:

- 1) Code Complexity—Keep all classes and methods simple and readable
- 2) Comments—Comment both at a Class and Method Level(use Java Docs or equivalent)
- 3) Code Standards—Follow industry standards for whatever language you chose
- 4) OOP—Make sure your solutions follow the patterns set out by OOP
- 5) Use Selenium Webdriver for the Website Front-End questions
- 6) Store all of your code in GitHub and make sure the code is runnable

## Section 1)

### General Programming:

Question 1: Given two strings, check if they're anagrams or not

Two strings are anagrams if they are written using the same exact letters. For example, 'Eleven plus two' and 'Twelve plus one' are meaningful anagrams of each other.

Using the least amount of code, prove if the following are anagrams of each other:

- 1) Punishment—>Nine Thumps
- 2) The Morse code — > Here come dots
- 3) Snooze alarms—>Alas! No more Zs
- 4) Halley's Comet—>Shall yet come
- 5) One good turn deserves another.—>Do rogues endorse that? No,never!

Question 2: Whats wrong with the given code below:

```
public class Main {  
  
    public static void main(String[] args) {  
        System.out.println(fixSpelling("hostleworld"));  
    }  
  
    public static String fixSpelling(String name) {  
        String wordToCheck = new String(name);  
        if(wordToCheck == "hostleworld" ) {  
            name = "hostelworld";  
        }  
        else {  
            fixSpelling(name);  
        }  
        return name;  
    }  
}
```

## Section 2)

### Front End automation

**Use case 1:** User Searches for Available properties in ‘Dublin, Ireland’ on the Hostelworld Site and sorts the results by **Name**

**User Journey:** The user goes to [www.hostelworld.com](http://www.hostelworld.com) and from here (the home page) searches for “*Dublin, Ireland*” using the [predictive text search<sup>1</sup>](#) box and clicks the search button.

Once the search is submitted they will be brought to the Property results page where they sort the properties displayed by **Name**.

**Requirement:** Using Webdriver implement the user journey and ensure that once sorted, the properties are displayed in alphabetical order.

Repeat this test for the following dimensions:

1. Full screen

2.412x732

3.768x1024

**Notes:** Class names or ids are available on all elements. Any browser of your choice can be used.

### API Testing

The aim of this section is to demonstrate your ability in creating tests against a Restful API service. For this we will use a subset of the GitHub gist API (<https://developer.github.com/v3/gists/>).

The tests should ensure that the API implementations conform to the github documentation. There should be a single script which launches your test suite, and outputs the pass/fail status for each test. **You can create the tests in any language or tool you wish but please remember we use either Java / Groovy here at Hostelworld.**

If you do use another tool / language please remember to document what we need to get your code running. In order to create your tests you will need an account on GitHub, once done please create a personal access token on your GitHub account settings page

(<https://github.com/settings/applications#personal-access-tokens>), and use that to authorize API calls. **Please DO NOT send us your personal token** in the tests, instead allow us to pass our own token (you can do this however you like i.e. via command line, config file etc)

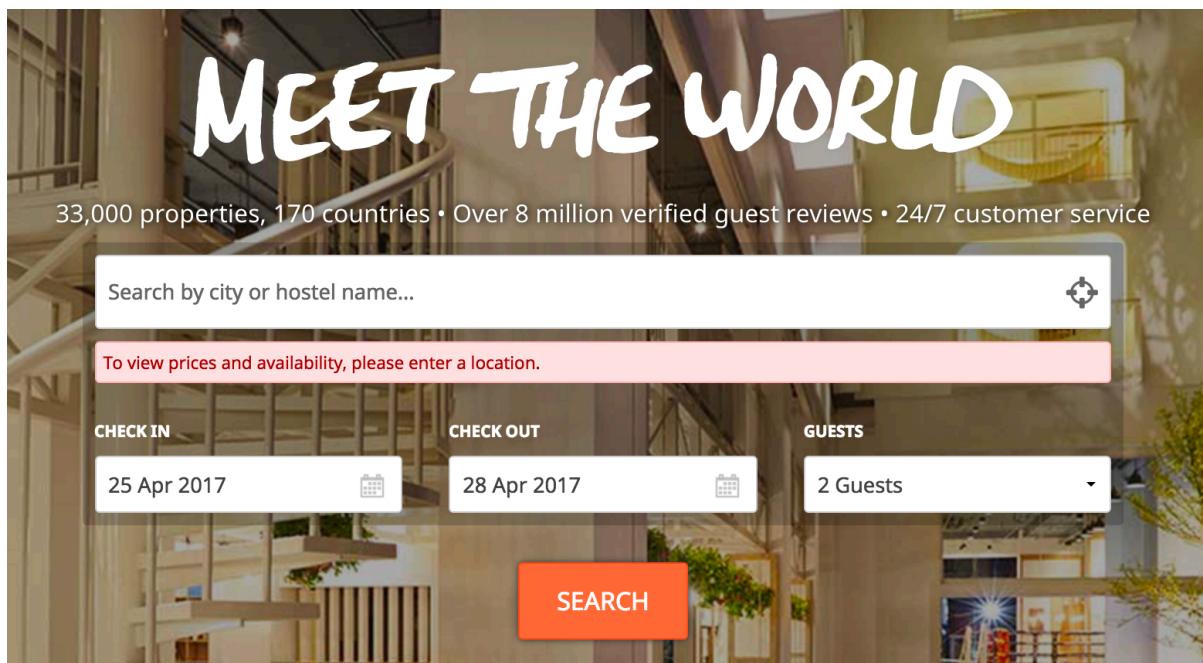
At the very least the tests should cover:

Assert that the create call creates gists. Assert that listing gists returns the fields that the docs specify. Assert that the delete call deletes a gist.

**There is no need to test the OAuth authentication process.**

## Appendix

### 1. Hostelworld Predictive Search Box



### 2. Hostelworld Search Results Page

The screenshot shows the search results page for "Dublin, Ireland" on Hostelworld. At the top, it displays "66 Hostels in Dublin, Ireland - 32 Available". The search filters are identical to the ones on the homepage: "LOCATION" (Dublin, Ireland), "CHECK IN" (25 Apr 2017), "CHECK OUT" (28 Apr 2017), and "GUESTS" (2 Guests). An orange "Search" button is present. Below the filters, there is a "BEST PRICE GUARANTEE" badge with the text "If you find a booking cheaper on any other website, we'll refund you the difference." The main content area features three "FEATURED PROPERTIES": "Abigail's Hostel" (Fabulous rating, From: € 18.00), "Abrahams Hostel" (Fabulous rating, From: € 17.00), and "Sky Backpackers" (Fabulous rating, From: € 31.67). At the bottom of the page, there are buttons for "Filter", "Sort", "Per page", "DISPLAY: List" (selected), "Gallery", and "Map". A call-to-action button "Click here to filter your results." is located near the bottom left.

### 3.Hostelworld Sort Option

The screenshot shows the Hostelworld website interface. At the top, there are three buttons: 'Filter', 'Sort' (with a dropdown arrow), and 'Per page'. A dropdown menu is open under 'Sort', listing 'Price', 'Rating', 'Distance', and 'Name'. To the left of the dropdown is a thumbnail image of a red telephone booth with hanging flower boxes, with a green circular overlay containing a white thumbs-up icon. Below the dropdown is a large image of the same scene. To the right of the dropdown is a listing for 'Isaacs Hostel'. The listing includes the hostel's name, address (2-5 Frenchmans Lane, ...), a distance of 0.5km, a 'Fabulous' rating of 8.6 from 9571 reviews, and icons for Free WiFi and Free Breakfast. Below the listing is a text block stating 'So why choose Isaacs? We have won Best Hostel in Dublin in 2014, 2015 and 2016' and a 'Compare' button.

## Resources

Hostelworld Website: <http://www.hostelworld.com>

WebDriver Homepage: <http://docs.seleniumhq.org/projects/webdriver/>  
WebDriver tutorial:

<https://code.google.com/p/selenium/wiki/GettingStarted>

Retrofit HomePage: <http://square.github.io/retrofit/>  
Retrofit tutorial: <http://www.vogella.com/tutorials/Retrofit/article.html>

GitHub: <https://github.com/>  
GitHub Gist: <http://en.wikipedia.org/wiki/GitHub#Gist>