



Automation Project

Java+Selenium+TestNG+ReadFiles

High Level Automation Steps

- Goto <https://mockaroo.com/>
- Specify Field names, Type, Rows count, format
- Download data in CSV format
- Read all data from the file and load to collection

Detailed Automation Steps

1. Create a Java class MockarooDataValidation.java
2. Navigate to <https://mockaroo.com/>
3. Assert title is correct.
4. Assert *Mockaroo* and *realistic data generator* are displayed



Need some mock data to test your app? Mockaroo lets you generate up to 1,000 rows of realistic test data in CSV, JSON, and XML.
Download data using your browser or sign in and create your own [Mock APIs](#).
Need more data? Plans start at just \$50/year.

use Xpath with tagname
and classname then get
text and assert

```
Elements Console Sources Network Performance Memory Application Security Audits
<!doctype html>
<html class="gr_mockaroo_com">
  <head>
    <meta charset="utf-8">
    <title>Mockaroo - Realistic Data Generator</title>
    <meta name="description" content="Mockaroo lets you generate up to 1,000 rows of realistic test data in CSV, JSON, and XML.">
    <meta name="keywords" content="Mockaroo, realistic data generator, test data, CSV, JSON, XML">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="/css/main.css" rel="stylesheet">
    <script src="/js/main.js"></script>
  </head>
  <body class="mockaroo" data-no-transition-cache="on" data-turbolinks="true">
    <div class="navbar navbar-fixed-top">
      <div class="container">
        <div class="navbar-header">
          <a href="/">
            <div class="brand">Mockaroo</div>
            <div class="tagline">realistic data generator</div>
          </a>
        </div>
      </div>
    </div>
  </body>
</html>
```

5. Remote all existing fields by clicking on x icon link

Note: findElement method in Selenium always find first one if there are multiple matches

The screenshot shows the Mockaroo website interface. At the top, there's a green header with the Mockaroo logo and the text "realistic data generator". Below this is a table with columns: Field Name, Type, and Options. The table lists several fields: 'id' (Row Number), 'first_name' (First Name), 'last_name' (Last Name), and 'email' (Email Address). Each field has a 'blank' checkbox, a percentage input, a 'fx' button, and an 'x' icon. A red arrow points from the 'x' icon in the 'id' row to the browser's developer tools. The developer tools show the HTML structure, with a red arrow pointing to the following element:

```
<a class="close remove-field remove_nested_fields" data-association="columns" href="javascript:void(0)">x</a>
```

**use Xpath with tagname
and classname**

→ Field Name Type Options

→ Add another field

→ # Rows: Format: Line Ending: Include: ☒ header ☐ BOM

6. Assert that 'Field Name', 'Type', 'Options' labels are displayed

use Xpath with tagname,
classname and text

→

```
<div class="column column-header column-name">Field Name</div>  
<div class="column column-header column-type">Type</div> == $0  
<div class="column column-header column-options">Options</div>
```

7. Assert that 'Add another field' button is enabled. Find using xpath with tagname and text. *isEnabled() method in selenium*

8. Assert that default *number of rows is 1000.*

9. Assert that default *format selection is CSV*

10. Assert that *Line Ending is Unix(LF)*

11. Assert that *header checkbox is checked and BOM is unchecked*

Field Name	Type	Options
<input type="text" value="enter name..."/>	<input type="text" value="choose type..."/>	blank: <input type="text" value="0"/> % <input type="text" value="fx"/> ×
<input type="button" value="Add another field"/>		

Choose a Type

All (1)

Location (1)

Commerce

Advanced

Personal

Nature

Car

Basic

IT

Health

Products

City

New York

Berlin

London

12. Click on 'Add another field' and enter name "City"

13. Click on Choose type and assert that Choose a Type dialog box is displayed.

14. Search for "city" and click on City on search results.

15. Repeat steps 12-14 with field name and type "Country"

Field Name	Type	Options
City	City	blank: 0 % fix x
Country	Country	restrict countries... blank: 0 % fix x

Add another field

Rows: 1000 Format: CSV Line Ending: Unix (LF) Include: ☒ header ☐ BOM

Download Data

Preview

More ▾

Want to save this for later? [Sign up for free.](#)

```
1 City,Country
2 Zaandam,Netherlands
3 Culasian,Philippines
4 Eindhoven,Netherlands
5 Wangcun,China
6 Zhapu,China
7 Changning,China
8 Vodinci,Croatia
```

16. Click on Download Data.

17. Open the downloaded file using BufferedReader.

18. Assert that first row is matching with Field names that we selected.

19. Assert that there are 1000 records

```
1 City, Country
2 Zaandam, Netherlands
3 Culasian, Philippines
4 Eindhoven, Netherlands
5 Wangcun, China
6 Zhapu, China
7 Changning, China
8 Vodinci, Croatia
```

20. From file add all Cities to Cities ArrayList

21. Add all countries to Countries ArrayList

22. Sort all cities and find the city with the longest name and shortest name

23. In Countries ArrayList, find how many times each Country is mentioned.

and print out

ex: Indonesia-10

Russia-7 etc

```
1 City, Country
2 Zaandam, Netherlands
3 Culasian, Philippines
4 Eindhoven, Netherlands
5 Wangcun, China
6 Zhapu, China
7 Changning, China
8 Vodinci, Croatia
```

24. From file add all Cities to citiesSet HashSet

25. Count how many unique cities are in Cities ArrayList and assert that it is matching with the count of citiesSet HashSet.

26. Add all Countries to countrySet HashSet

27. Count how many unique cities are in Countries ArrayList and assert that it is matching with the count of countrySet HashSet.

28. Push the code to any GitHub repo that you have and submit the url