

Automation Project

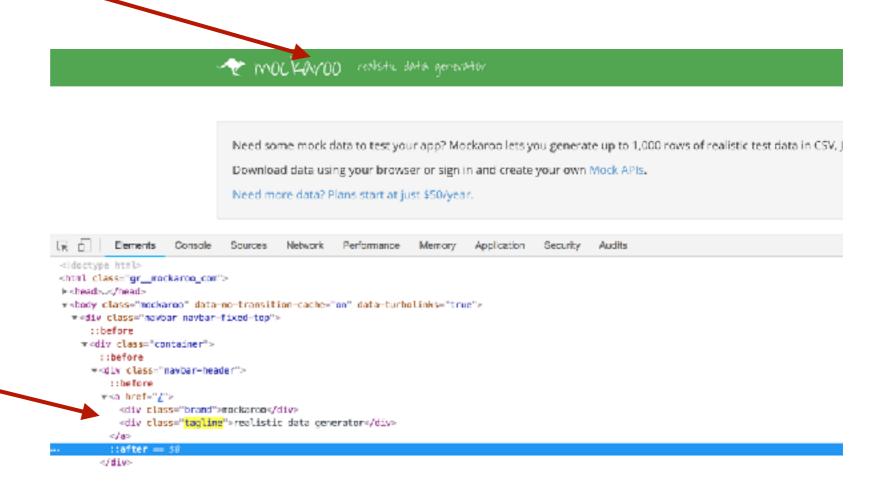
Java+Selenium+TestNG+ReadFiles

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

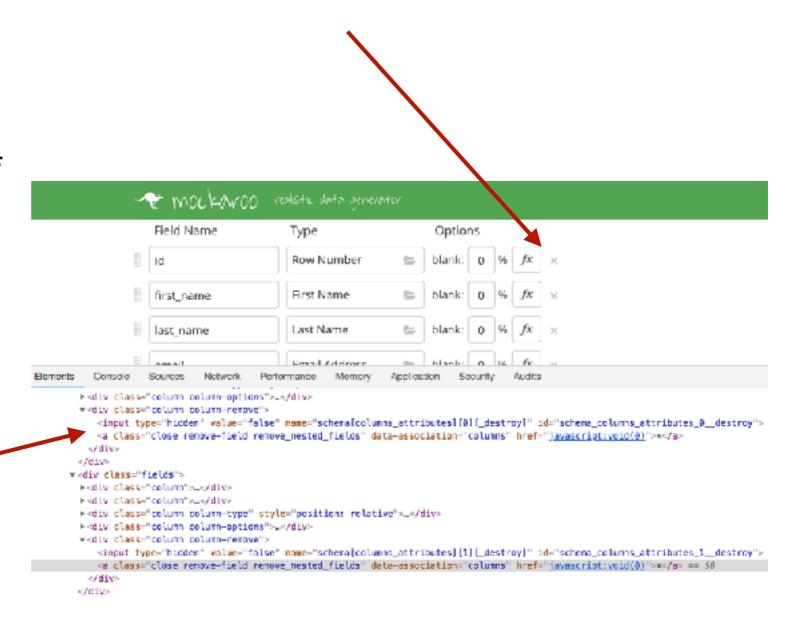


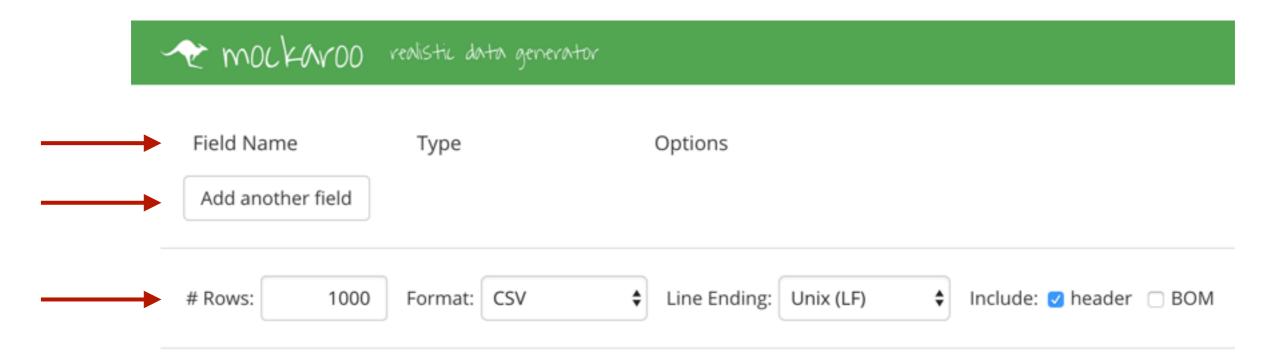
use Xpath with tagname and classname then get text and assert

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

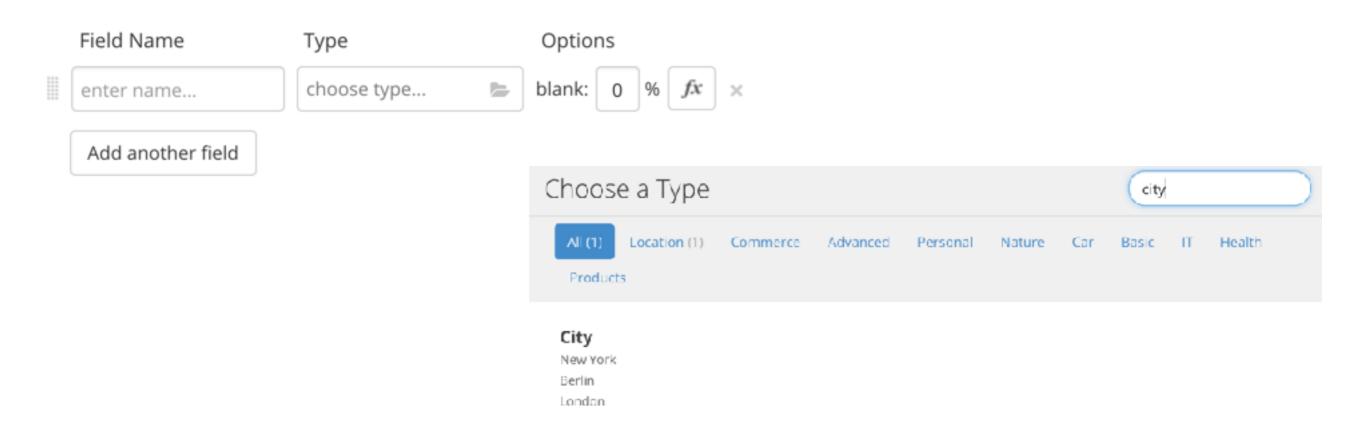
Note: findElement method in Selenium always find first one if there are multiple matches

use Xpath with tagname and classname

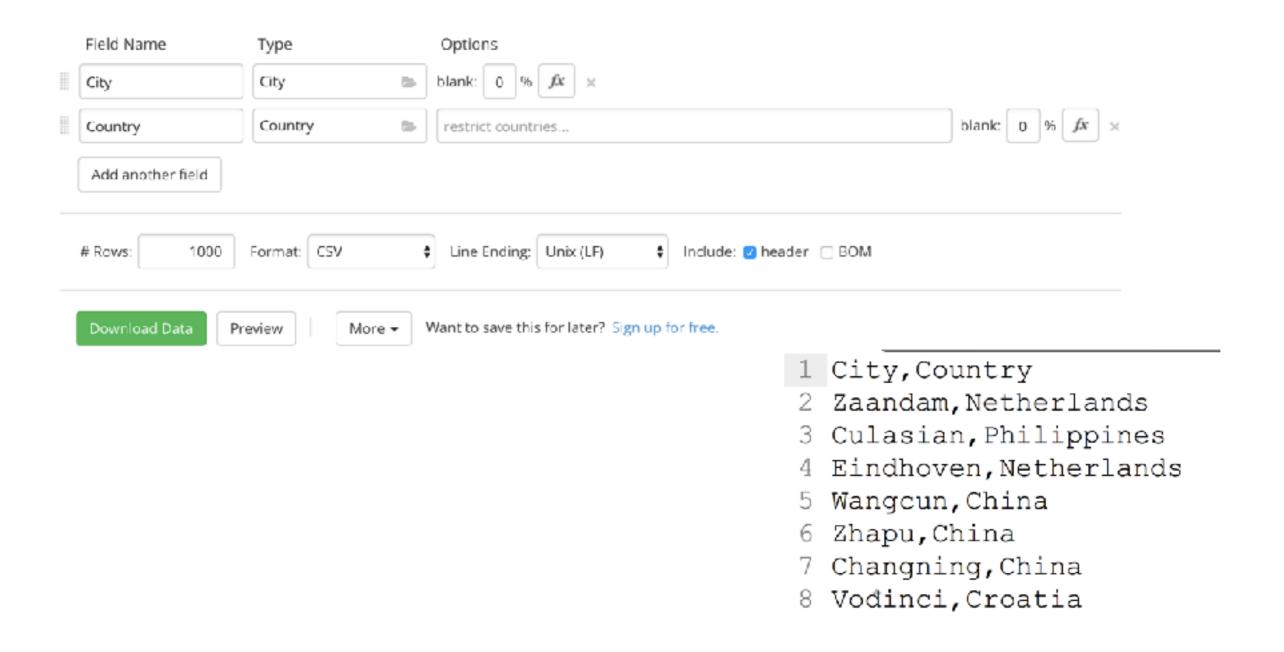




- 6. Assert that 'Field Name', 'Type', 'Options' labels are displayed use Xpath with tagname, class="column column-header column-name">Field Name</div>
 div class="column column-header column-type">Type</div> == \$0
 classname and text
- 7. Assert that 'Add another field' button is enabled. Find using xpath with tagname and text. is Enabled() 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



- 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"



- 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