## Ajax Dropdowns / Fields

This is the next tutorial in selenium-java series. Please go through the previous tutorials before you start this one. In the last tutorial, we learned how to handle web tables. In this tutorial we will see how to handle ajax based fields and dropdowns!

**What you will Learn:**
1. Ajax based search field/auto-suggestive dropdown
2. Ajax dropdown – exercise 1
3. Ajax dropdown – exercise 2

### Auto suggestive dropdowns (Ajax based search field)

See figure 1. We have a search field in which pre-defined dropdown options are not present. Only when you enter a search keyword, the options get displayed. Navigate to https://ksrtc.in/oprs-web/ & type BENG in the 'Leaving From' field. Notice that we get various suggestive options on the fly. This kind of field is what we call as ajax-based field. Ajax based requests dynamically render the html.

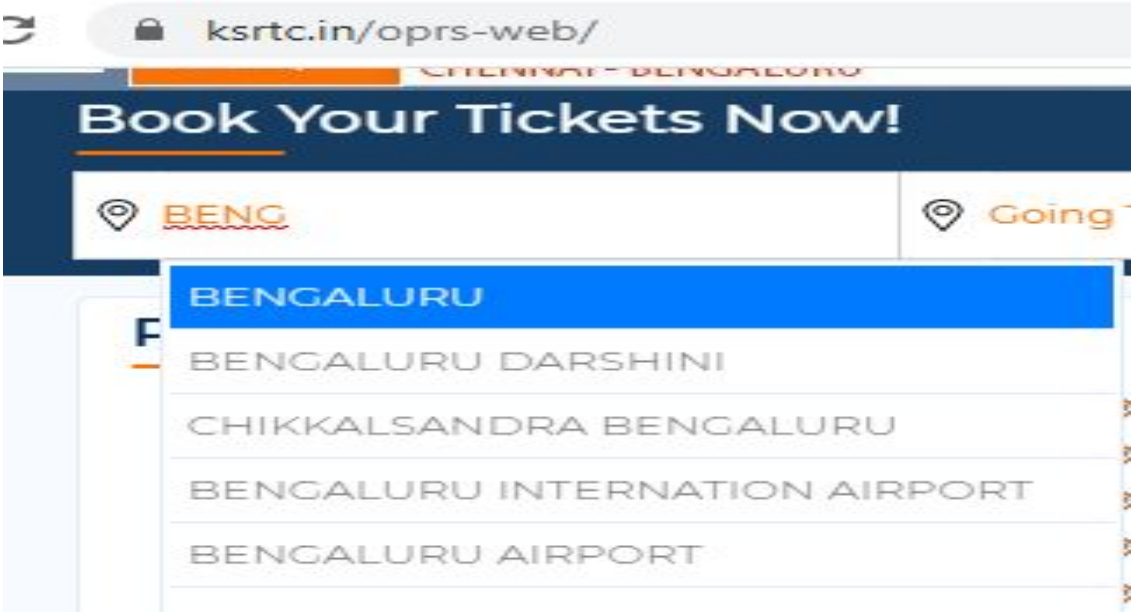


***Figure 1***

The use case which we want to automate here is:

- Type letters BENG or beng in the search field
- Select the option BENGALURU DARSHNI from the dropdown
- Extract the text BENGALURU DARSHNI from the 'From To' field.

See figure 2. Notice that, by pressing the key-down, you can move to different options in the dropdown. Also, wherever the key is pointing to, that option appears in the search field.

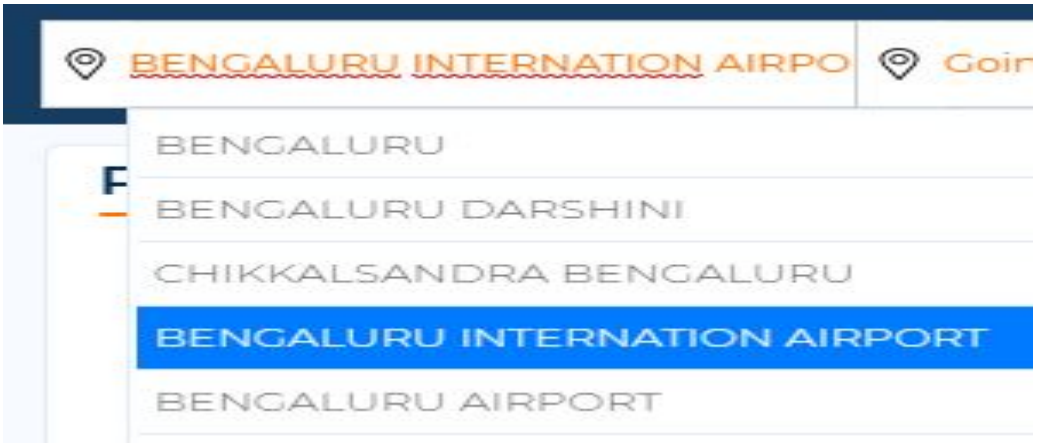


***Figure 2***

Let us inspect 'Leaving From' field. We can use the value of 'id' attribute since that seems to be unique

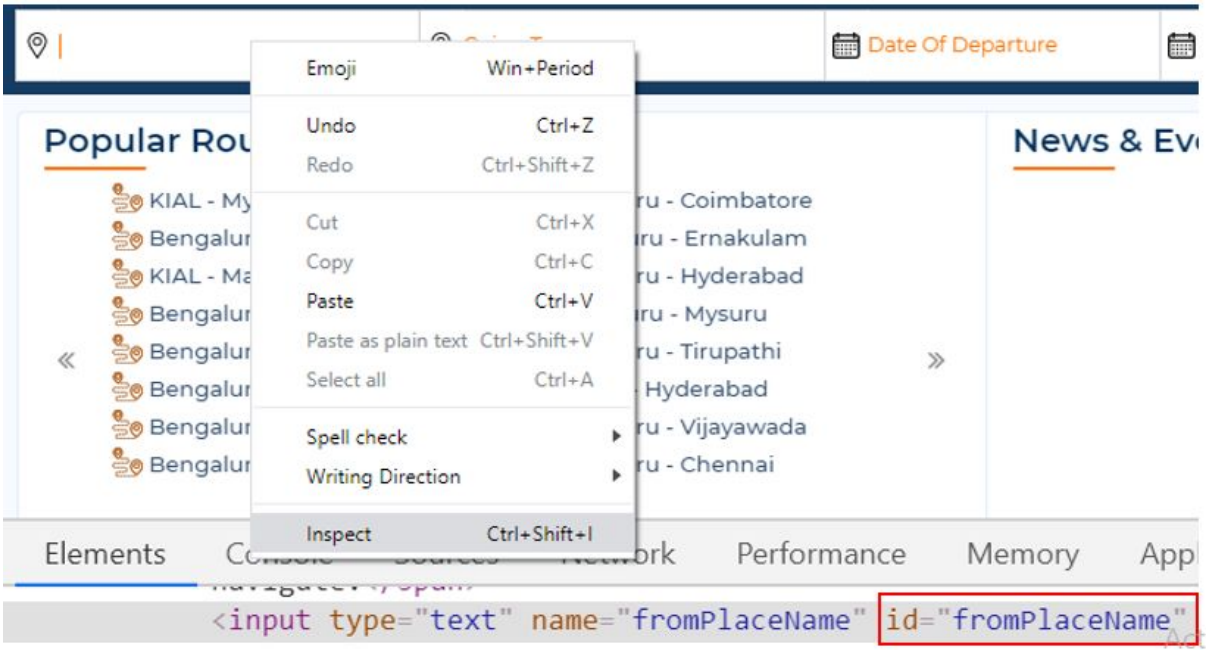


***Figure 3***

See below, line#17 types BENG in 'from to' field, line#18 moves key-down, line#19 moves key-down. In line#20, the attribute "value" fetches the text that is entered during runtime.

```
15        driver.get("https://ksrtc.in/oprs-web/");
16
17        driver.findElement(By.id("fromPlaceName")).sendKeys("BENG");
18        driver.findElement(By.id("fromPlaceName")).sendKeys(Keys.DOWN);
19        driver.findElement(By.id("fromPlaceName")).sendKeys(Keys.DOWN);
20        System.out.println(driver.findElement(By.id("fromPlaceName")).getAttribute("value"));
```

***Figure 4***

When you run the script, notice below that BENGALURU DARSHNI is typed in the 'from to' field

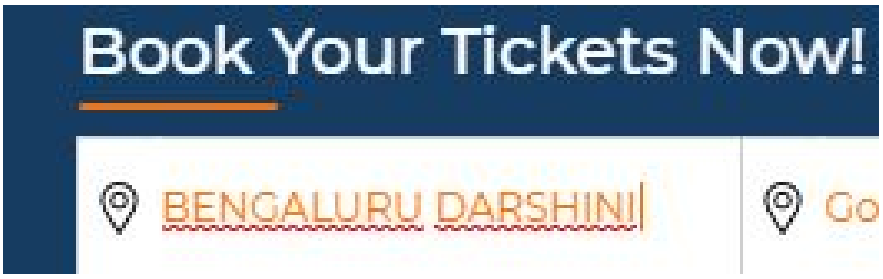


***Figure 5***

See below. BENGALURU DARSHNI gets printed in console.

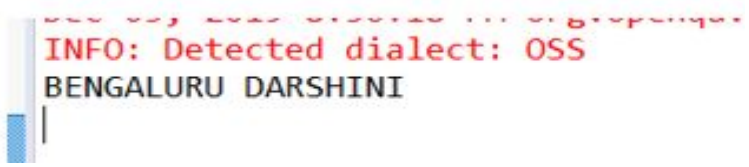


***Figure 6***

**Ajax dropdown – exercise 1**

Navigate to https://www.makemytrip.com/

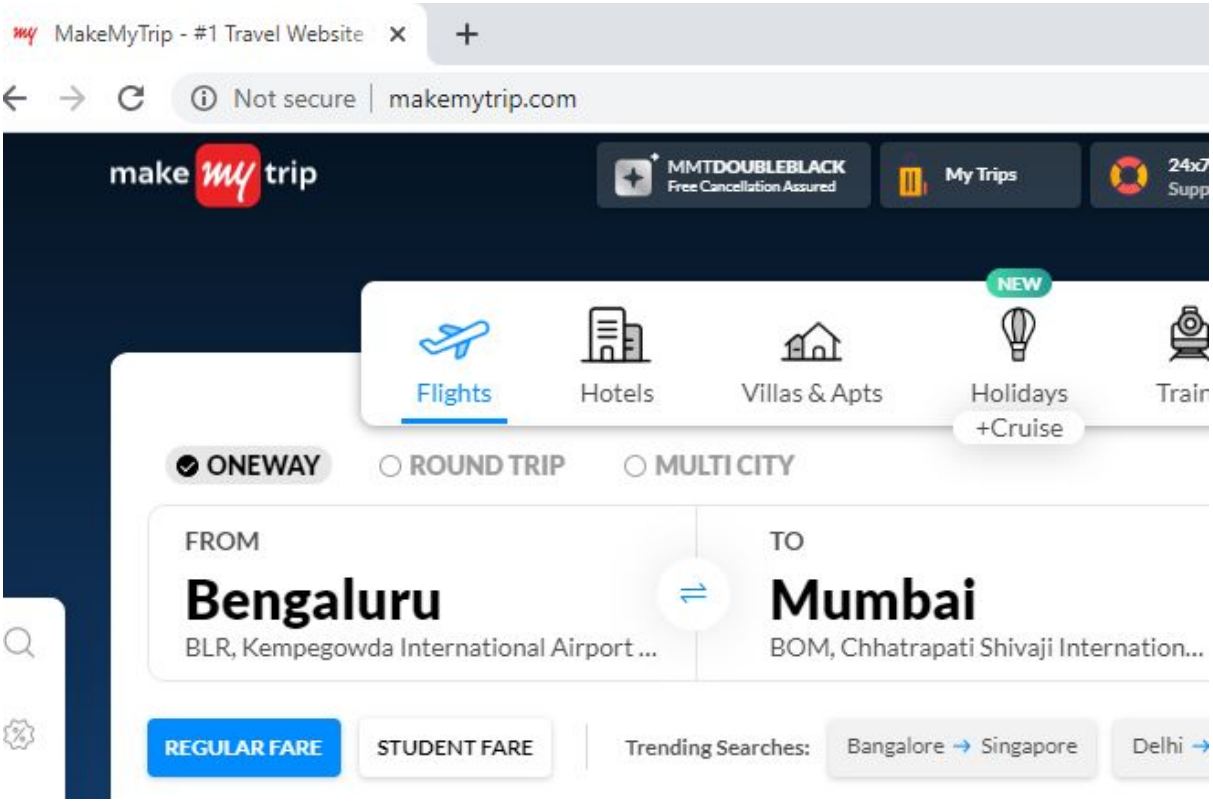


***Figure 7***

We see 2 ajax dropdowns over here: FROM and TO. The task here is to select 'Mumbai' in the FROM dropdown field


***Figure 8***

Now how do we know that these are ajax dropdown fields? When you right click a city in the FROM field and inspect it, you can see 'autosuggest' keywords in the values of few attributes. This autosuggest keyword suggests it to be a ajax field

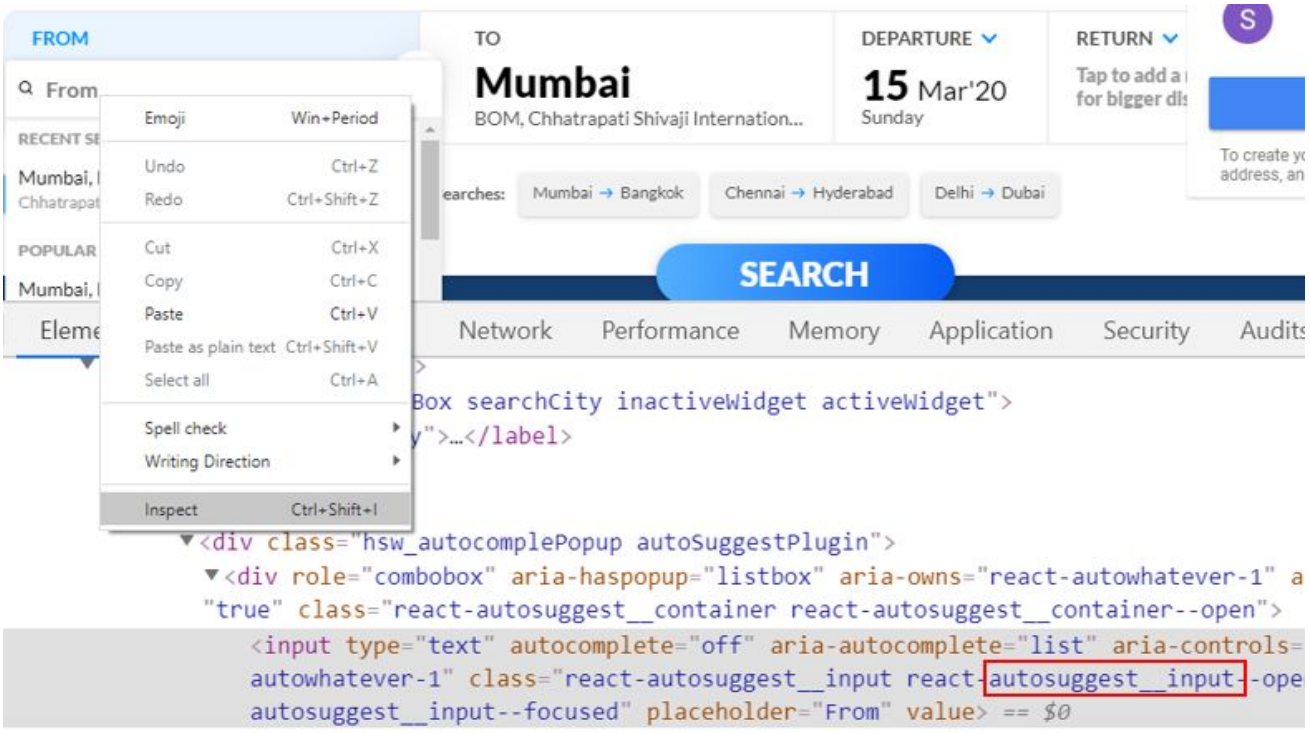


***Figure 9***

Now see below. Bring the mouse cursor on FROM field. Right click FROM field and inspect it. We see that FROM is represented by  tag

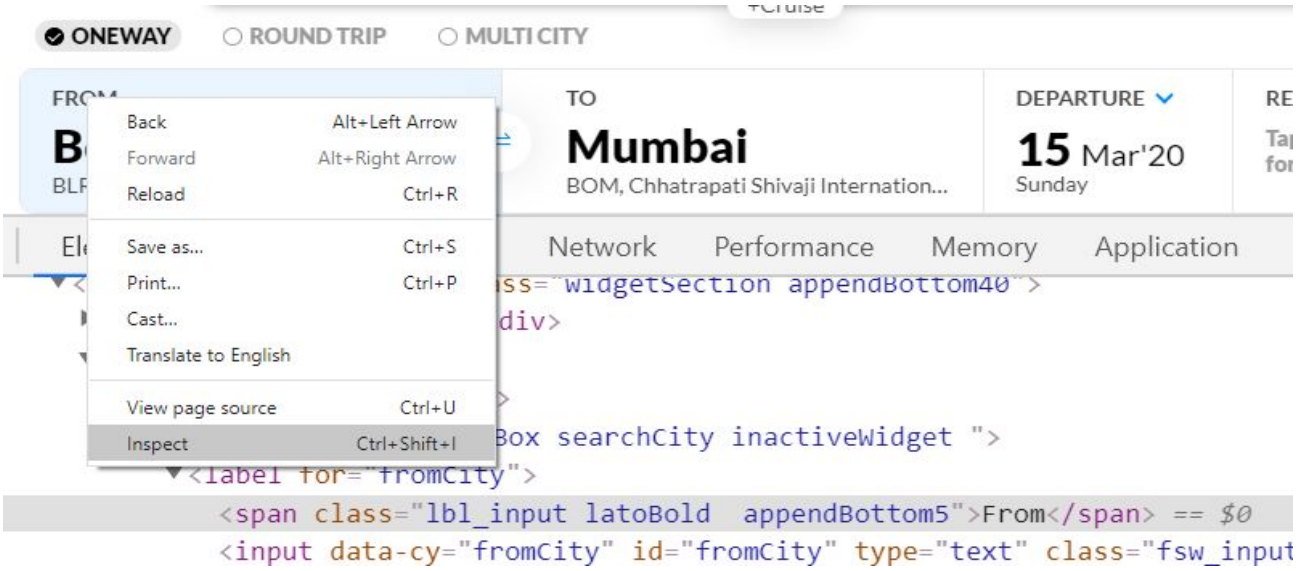


***Figure 10***

If you hover the mouse over this span tag, you can see that FROM field gets highlighted

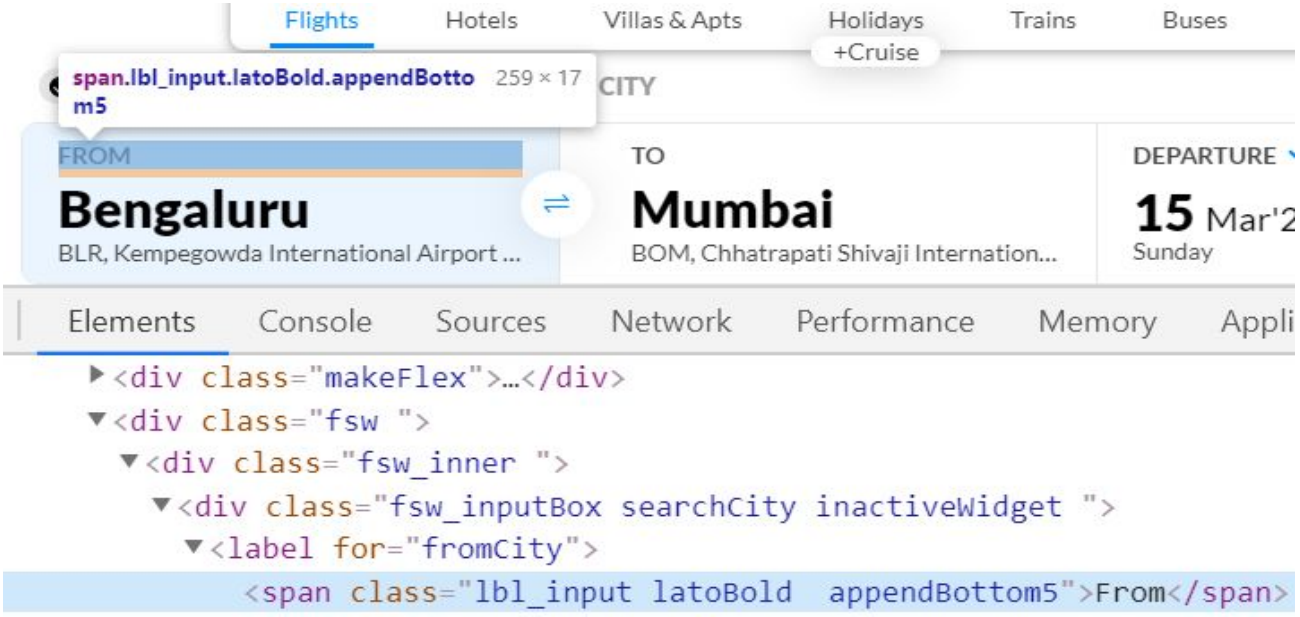


***Figure 11***

We can use cssSelector to create custom path, see below. We can append the classes using dot.

```
15        driver.get("https://www.makemytrip.com/");
16
17        driver.findElement(By.cssSelector(".lbl_input.latoBold.appendBottom5")).click();
```

***Figure 12***

So when we run the above script, the 'FROM' field gets clicked and the search box 'From' appears

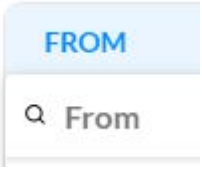


***Figure 13***

Next, we will inspect 'From' search field (because we will be typing the name of city in this field), see figures 14 and 15

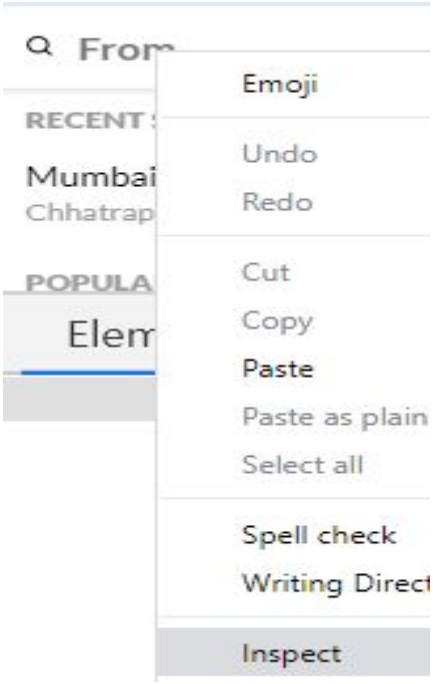


***Figure 14***


***Figure 15***

Our complete script would be as shown below. Lines#20 and 21 would help us in key down and 'Enter' operations

```
15          driver.get("https://www.makemytrip.com/");
16
17          driver.findElement(By.cssSelector(".lbl_input.latoBold.appendBottom5")).click();
18          Thread.sleep(2000);
19          driver.findElement(By.xpath("//input[@aria-autocomplete='list']")).sendKeys("MUM");
20          driver.findElement(By.xpath("//input[@aria-autocomplete='list']")).sendKeys(Keys.DOWN);
21          driver.findElement(By.xpath("//input[@aria-autocomplete='list']")).sendKeys(Keys.ENTER);
```

***Figure 16***

Run the script, notice that 'Mumbai' gets selected in FROM dropdown

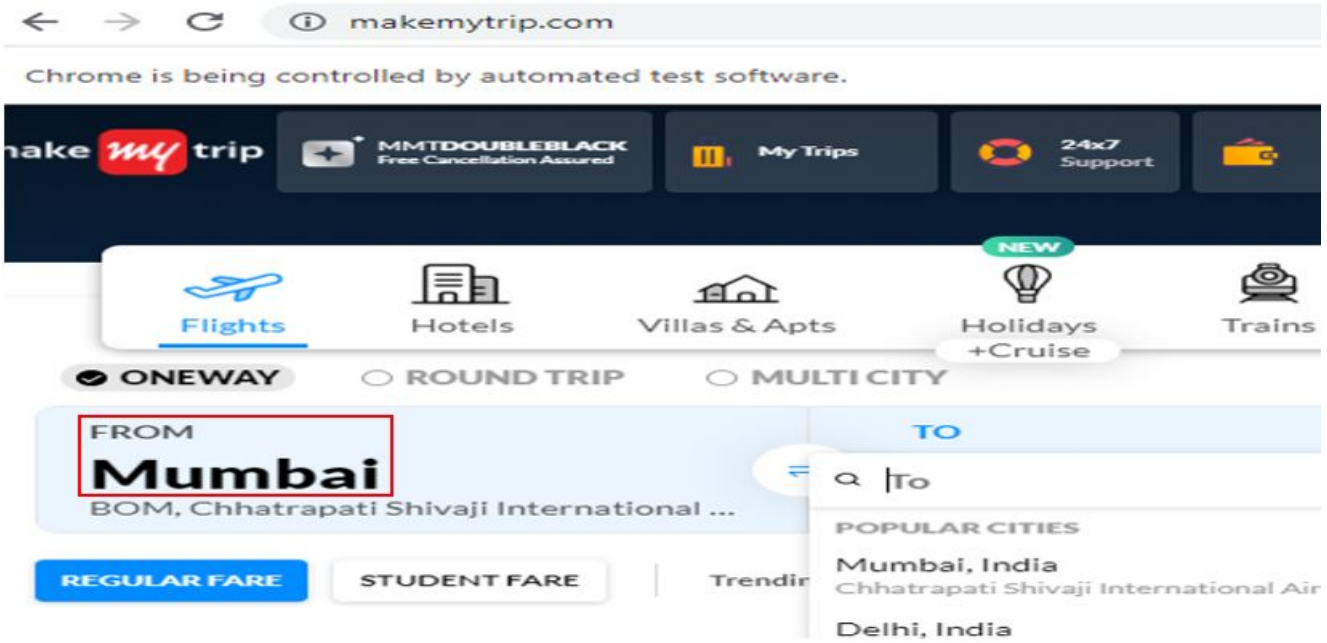


***Figure 17***

**Ajax dropdown – exercise 2**

Navigate to https://in.yahoo.com/?p=us and type 'hello' in the search text field. You will see list of auto-suggestive dropdown values, see below figure

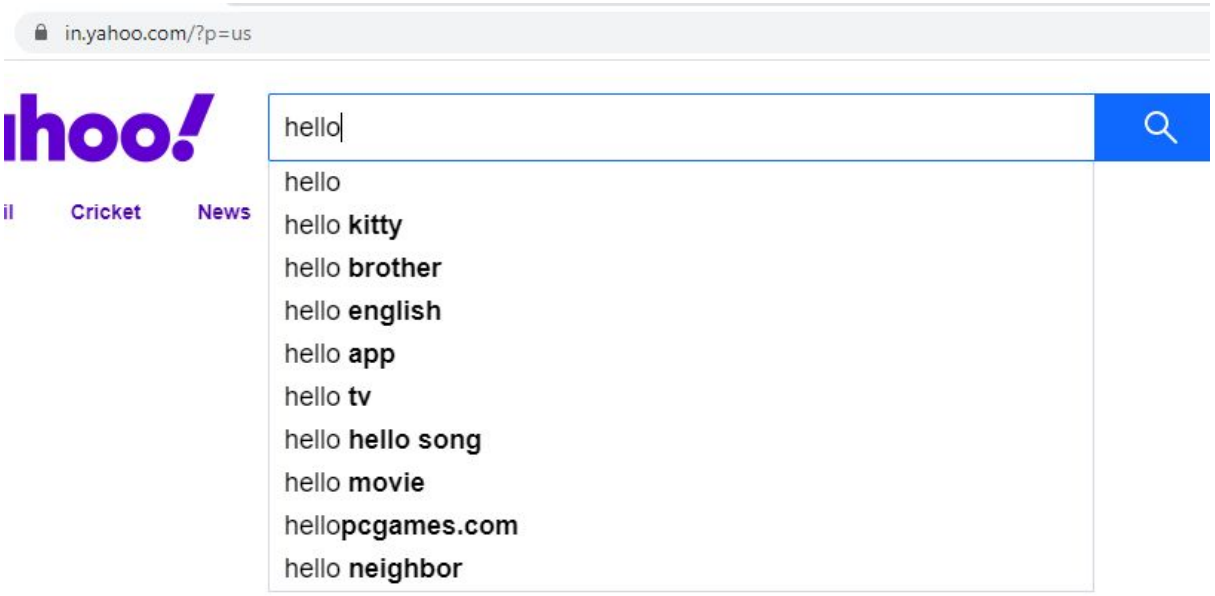


***Figure 18***

Now our requirement is to grab all these values and print them in console. Right click and inspect any value in the dropdown

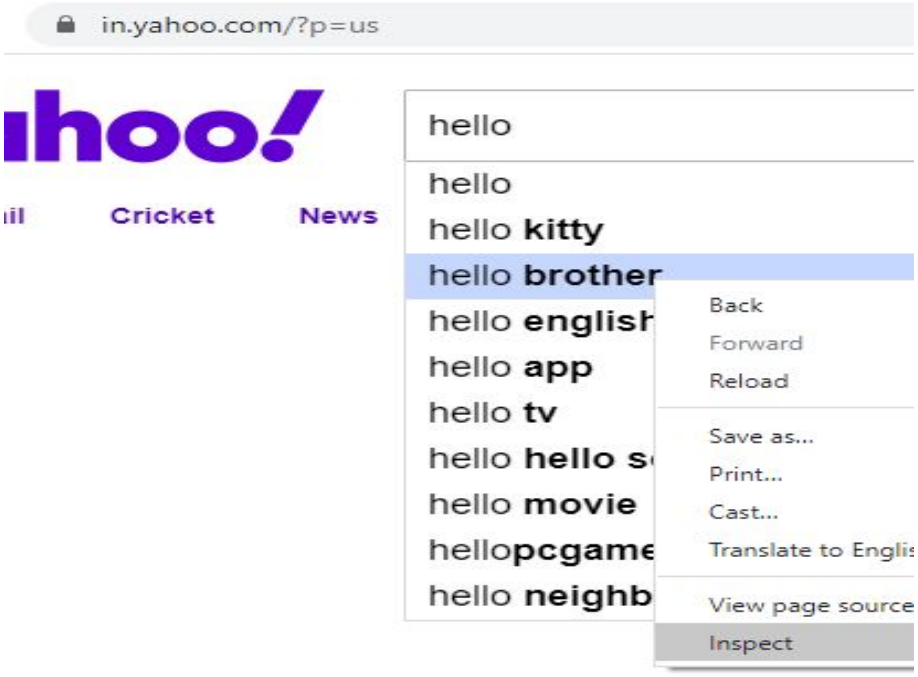


***Figure 19***

If you mouse hover over <div tag, you see that the entire box (containing all the auto-suggestive values) gets highlighted.

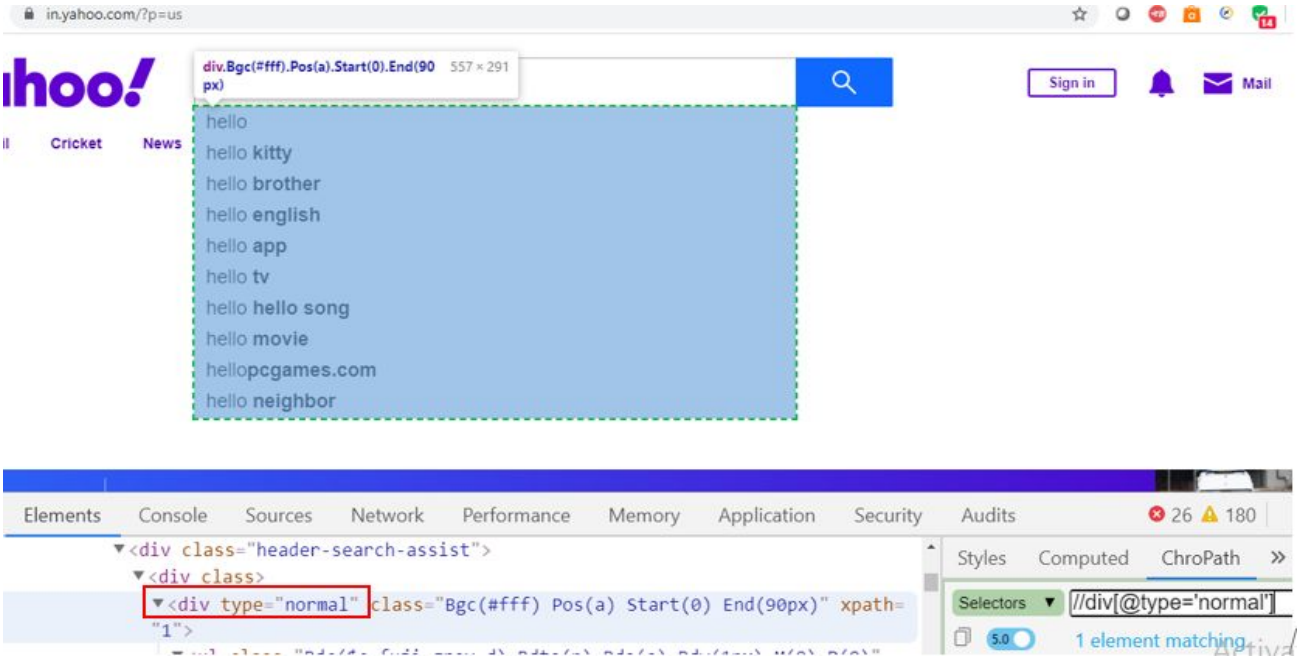


***Figure 20***

So the xpath //div[@type='normal'] will contain a list of all the auto-suggestive values (see line#24 below). We can than use a 'for' loop to iterate over each value and print it in console

```
17          driver = new ChromeDriver();
18
19          driver.get("https://in.yahoo.com/?p=us");
20
21          driver.findElement(By.xpath("//input[@type='text']")).sendKeys("hello");;
22
23          Thread.sleep(5000);
24          List<WebElement> values = driver.findElements(By.xpath("//div[@type='normal']"));
25
26          for(int i=0; i < values.size(); i++)
27          {
28              System.out.println(values.get(i).getText());
29          }
```

***Figure 21***

Run the script, notice below that the console prints all the auto-suggestive values


***Figure 22***

These are some of the ways to handle the ajax auto-suggestive values. Thank you for reading!