Selenium Web Tables Handling

Handling **web tables in Selenium with Java** is a common requirement for automation testing. Web tables are typically built using HTML **`<table>`** tags and can be **static** or **dynamic**.

✅ Topics Covered:

1. What is a Web Table?
2. Types of Web Tables
3. Locating Table Elements
4. Example HTML Table
5. Java Selenium Code to Handle:
    - Row Count
    - Column Count
    - Cell Data Retrieval
    - Loop through All Table Data
6. XPath Strategies
7. Bonus: Table data validation/assertion

---

1. 🔎 What is a Web Table?

A **web table** is an HTML structure used to display data in tabular format using:

```
html

<table>       <!-- Table -->
  <thead>     <!-- Header (optional) -->
  <tbody>     <!-- Body rows -->
  <tr>        <!-- Row -->
  <td>        <!-- Data cell -->
```

---

2. 📘 Types of Web Tables

- **Static Web Tables:** Table structure doesn't change.
- **Dynamic Web Tables:** Rows/columns are populated dynamically (e.g., AJAX).

---

3. ✅ Locating Table Elements (XPath)

Examples:

```java
// All rows inside tbody
List<WebElement> rows = driver.findElements(By.xpath("//table[@id='example']//tbody/tr"));

// All columns of the first row
List<WebElement> columns =
driver.findElements(By.xpath("//table[@id='example']//tbody/tr[1]/td"));

// Specific cell [row 2, column 3]
WebElement cell = driver.findElement(By.xpath("//table[@id='example']//tbody/tr[2]/td[3]"));
```

4. 🔪 Sample HTML Table

```html
<table id="employee">
  <thead>
    <tr>
      <th>Name</th>
      <th>Role</th>
      <th>Location</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>John</td>
      <td>QA</td>
      <td>New York</td>
    </tr>
    <tr>
      <td>Jane</td>
      <td>Dev</td>
      <td>London</td>
    </tr>
  </tbody>
</table>
```

5. 🔧 Selenium Java Code to Handle Web Table

✅ a) Setup & Load Page

```java
WebDriver driver = new ChromeDriver();
driver.get("https://your-url-with-table.com");
driver.manage().window().maximize();
```

✅ b) Count Rows and Columns

```java
List<WebElement> rows = driver.findElements(By.xpath("//table[@id='employee']/tbody/tr"));
System.out.println("Total Rows: " + rows.size());

List<WebElement> cols = driver.findElements(By.xpath("//table[@id='employee']/thead/tr/th"));
System.out.println("Total Columns: " + cols.size());
```

✅ c) Read a Specific Cell

```
java

WebElement cell = driver.findElement(By.xpath("//table[@id='employee']/tbody/tr[2]/td[1]"));
System.out.println("Second row, first column value: " + cell.getText());
```

✅ d) Read All Table Data (Nested Loops)

```
java

for (int i = 1; i <= rows.size(); i++) {
    for (int j = 1; j <= cols.size(); j++) {
        WebElement data = driver.findElement(By.xpath("//table[@id='employee']/tbody/tr[" + i +
"]/td[" + j + "]"));
        System.out.print(data.getText() + "   ");
    }
    System.out.println();
}
```

---

6. 🎯 Dynamic XPath (Alternative for unknown sizes)

You can use:

```
java

//td[contains(text(), 'QA')]
```

To locate a cell by value regardless of position.

---

7. ✅ Example: Assert Cell Value

```
java

String expectedValue = "London";
WebElement actualCell =
driver.findElement(By.xpath("//table[@id='employee']/tbody/tr[2]/td[3]"));
Assert.assertEquals(actualCell.getText(), expectedValue);
```

---

8. 🚗 Best Practices

- Use **`WebDriverWait`** if data is dynamic.
- Handle stale elements in dynamic tables.
- Avoid hardcoded indexes where possible—consider reading headers.

✅ Summary Table

Task	XPath Example
Count rows	`//table[@id='employee']/tbody/tr`
Count columns	`//table[@id='employee']/thead/tr/th`
Specific cell	`//table[@id='employee']/tbody/tr[2]/td[3]`
Loop through all data	Nested loop using XPath with tr[i] and td[j]
Value by content	`//table[@id='employee']//td[contains(text(),'QA')]`

Would you like a **real-time lab demo** site to practice, or need **code using Page Object Model (POM)?**

Great! Here's a **real-time lab demo site**, plus two full working examples:

✅ 1. 🔗 Demo Site for Web Table Practice

Use this site for practicing web table automation with Selenium:

👉 **URL:** https://www.w3schools.com/html/html_tables.asp

This page contains a static table like this:

```
html


<table id="customers">
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
  <!-- More rows... -->
</table>
```

✅ 2. Java + Selenium Code (No POM)

🔧 Example: Print All Table Data

```
java


import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import java.util.List;

public class WebTableExample {
    public static void main(String[] args) {

        WebDriver driver = new ChromeDriver();
        driver.get("https://www.w3schools.com/html/html_tables.asp");
        driver.manage().window().maximize();

        // Get total number of rows
        List<WebElement> rows =
driver.findElements(By.xpath("//table[@id='customers']/tbody/tr"));
        System.out.println("Total Rows: " + rows.size());

        // Get total number of columns (from header row)
        List<WebElement> columns =
driver.findElements(By.xpath("//table[@id='customers']/tbody/tr[1]/th"));
        System.out.println("Total Columns: " + columns.size());

        // Loop through table
        for (int i = 2; i <= rows.size(); i++) {
            for (int j = 1; j <= columns.size(); j++) {
                String cellXpath = "//table[@id='customers']/tbody/tr[" + i + "]/td[" + j + "]";
                WebElement cell = driver.findElement(By.xpath(cellXpath));
                System.out.print(cell.getText() + "  |  ");
            }
            System.out.println();
```

```
        }

        driver.quit();
    }
}
```

✅ 3. Same Example Using Page Object Model (POM)

📁 Project Structure

```
css

WebTableTest/
├── src/
│   ├── pages/
│   │   └── TablePage.java
│   └── tests/
│       └── WebTableTest.java
```

📄 TablePage.java

```
java

package pages;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import java.util.List;

public class TablePage {
    WebDriver driver;

    public TablePage(WebDriver driver) {
        this.driver = driver;
    }

    public int getRowCount() {
        return driver.findElements(By.xpath("//table[@id='customers']/tbody/tr")).size();
    }

    public int getColumnCount() {
        return driver.findElements(By.xpath("//table[@id='customers']/tbody/tr[1]/th")).size();
    }

    public String getCellData(int row, int col) {
        String xpath = "//table[@id='customers']/tbody/tr[" + row + "]/td[" + col + "]";
        return driver.findElement(By.xpath(xpath)).getText();
    }
}
```

📄 WebTableTest.java

```
java

package tests;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import pages.TablePage;
```

```
public class WebTableTest {
    public static void main(String[] args) {

        WebDriver driver = new ChromeDriver();
        driver.get("https://www.w3schools.com/html/html_tables.asp");
        driver.manage().window().maximize();

        TablePage table = new TablePage(driver);

        int rows = table.getRowCount();
        int cols = table.getColumnCount();

        System.out.println("Rows: " + rows + ", Columns: " + cols);
        System.out.println("Printing all table data:");

        for (int i = 2; i <= rows; i++) {
            for (int j = 1; j <= cols; j++) {
                System.out.print(table.getCellData(i, j) + "  |  ");
            }
            System.out.println();
        }

        driver.quit();
    }
}
```

✅ Want to Try Dynamic Tables Next?

If you'd like, I can show you how to handle **dynamic web tables** (e.g., sort, filter, paginated) using:

- Tables from **DataTables.net**
- JavaScript-loaded rows
- WebDriverWait, FluentWait

Let me know!