Aim: A basic overview of writing test cases to automate the process of testing.

Audience: A blackbox tester with the ability to select webpage elements using css or the willingness to learn to do so.

In this context, a **Test Case** comes in two parts.

Commands – using **selectors** and **actions** to drive the **manipulation** of a **website**.

Exceptions – using **predefined rules** and **expected outcomes** to **determine results**.

This document will cover both of these elements as well as formatting for the spread sheet to be imported and all the tools given to you.

**TEST CASE IMPORTING & FORMATTING**

When you are importing test cases using excel, you need to follow the following format:

|  |  |  |
| --- | --- | --- |
| Title | ID | Steps |
| Username without special char | 1234 | 1...2…3… |

This format is very important, and should be followed always for best results.

When writing test cases, there are special characters and formatting you will need to use.

Each test case will take up **one** line.

**In general, test cases should have a variable amount of commands followed by ONE exception.**

**COMMAND AND EXCEPTION SYNTAX**

**As a general template, here is a sample “command”**

#selector::action->(optional)argument;;

**As a general template, here is a sample “exception”**

%%rule::pass/fail;;

**In general, here are all the symbols used in “commands” and “exceptions”**

|  |  |
| --- | --- |
| # | Start of a command |
| :: | Separator between command/exception sections |
| ;; | End of a command/exception |
| -> | Argument for a command |

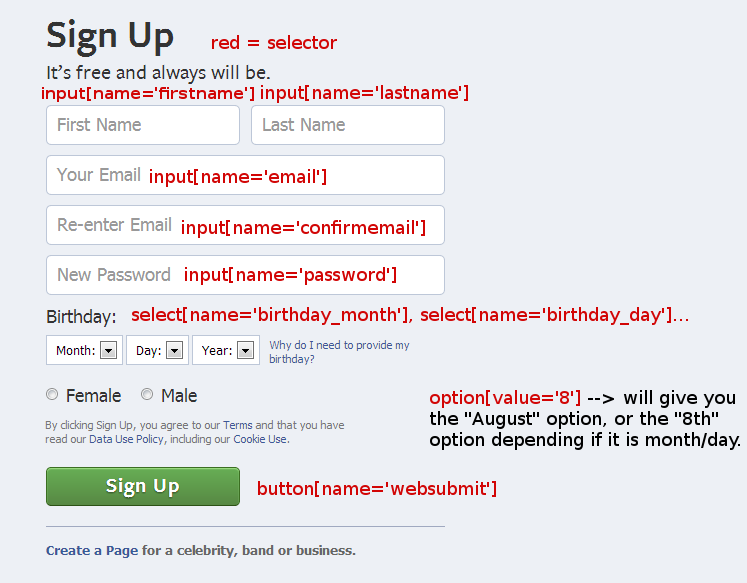
**COMMAND ACTIONS**

There are a set amount of “actions” you can use when manipulating a page with a command.

1. click – Pretty straightforward. This sends a click event to the selected element.
2. submit – This is for submitting forms if you are too lazy to select the submit button and use it. Use at own risk.
3. **sendkeys** – This sends the keys given in the argument.
   1. Ex: ##username::sendkeys->Brandon;;

**Bold** = requires argument

**SELECTOR QUICK-START**



Selectors are driven by the names of **html** elements like **input**, **div**, **h2**, **b** or **p**.

You can use **attributes** that are defined by the developers like **name** or **type** by specifying the **attribute**, a **=** sign and the **value** of that attribute **enclosed** **in** **quotes**.

If we were looking for an **input with a name of “username”** we would use **input[name=’username’]**

There are special attributes that serve no purpose other than to allow the developer to refer to them later.

These would include **ids** and **classes**.

An **id** is often used to identify something **unique** and a **class** is most often used to identify something **used throughout a page** (or even a site).

**WRITING TEST CASES a tutorial**

This will be an **interactive** tutorial. You will need

1. A browser (I will be using chrome)
2. This document.
3. Excel.

THE GUIDE

Step 1:  
open excel and enter this data

|  |  |  |
| --- | --- | --- |
| Firstname\_Empty | #errormsg\_0\_FirstName | You can’t leave this empty. |
| Lastname\_Empty | #errormsg\_0\_LastName | You can’t leave this empty. |
| Email\_spchar | #errormsg\_0\_GmailAddress | Please use only letters (a-z), numbers, and periods. |
| Password\_mismatch | #errormsg\_0\_PasswdAgain | These passwords don’t match. Try again? |
| Birthday\_invalidday | #errormsg\_0\_BirthDay | Hmm, the day doesn’t look right. Be sure to use a 2-digit number that is a day of the month. |
| Birthday\_invalidyear | #errormsg\_0\_BirthYear | Hmm, the year doesn't look right. Be sure to use four digits. |

Save it as rules.xls, and keep it open. We will be referring back to it later.

Step 2:

Open chrome. And navigate to google.com, logout of your google account if you are logged in a get yourself to the account creation screen.

Step 3:

We will now write a couple distinct COMMANDS and EXPECTATIONS to test the previously written rules in step 1

#input[name=’FirstName’]::sendkeys->john;;

#input[name=’LastName’]::sendkeys->doe;;

#input[name=’GmailAddress’]::sendkeys->@#@#@@#@#;; ------------- %%Email\_spch::pass;;

#input[name=’BirthDay’]::sendkeys->fd;; ----------------------- %%Birthday\_invalidday::pass;;

##submitbutton::click;;

These are a few examples that should allow you to write complete cases for this form page. It is very often the case that you write more than one command to each expectation. For a small example, let’s test the email special character rule completely

#input[name=’FirstName’]::sendkeys->john;;

#input[name=’LastName’]::sendkeys->doe;;

#input[name=’GmailAddress’]::sendkeys->@#@#@@#@#;;

#input[name=’BirthDay’]::sendkeys->19;;

#input[name=’BirthYear’]::sendkeys->1900;;