

jazzautomation.com

Page Configuration

September 3, 2013

JAZZ AUTOMATION

ph 800.640.5589
info@jazzautomation.com
www.jazzautomation.com
www.statementcorp.com



Overview

What is Jazz Automation?

Jazz Automation is for companies who want to improve the quality of web applications using Total Quality Testing™.

Jazz Automation allows for flexible, repeatable, and configurable testing of web based user interfaces using a plain English *scripting language*. Jazz Automation can be used with any browser on any platform and is operating system independent. Jazz Automation enables data driven testing (expectation/verification) as well as traditional user interface element detection. With Jazz automation, organizations are free to perform more exploratory and negative testing since Jazz Automation enables the organization to automate 'Happy Path' and regression testing.

Jazz Automation is open source tool built on Java that uses several open source libraries. Users of Jazz Automation are free to contribute to the product using the JazzAutomation GIT repository.

More Configuration Files with Custom Formats?

JazzAutomation is incredibly flexible and easy to use and understand. We have taken a true object oriented approach to configuration. The initial step in using JazzAutomation is defining the pages and page components that will be used during the tests. Each Page should be defined under the \$MyConfigDir/pages directory. Each website page should have a corresponding configuration page. The format used to define a page is industry standard JSON. JSON has gained large popularity over the last few years and is very easy for humans to read and comprehend. Please read more about JSON here <http://www.json.org>. Additionally, users should become familiar with jquery.

this the portal page. Here we could create a new page in our *bestBuyExample/pages* directory called *BestBuyPortalPage.json*. What would our JSON configuration page look like?

```
{
  "pageName": "BestBuyPortalPage",
  "title": "Best Buy: Making Technology Work for You",
  "urlExtension": "www.bestbuy.com",
  "keyDomElementName": "productMenu",
  "domElements": {
    "productMenu": {
      "jquery": "$('#nav li.nav-pro a[data-lid=\"ubr_prd\"]')"
    },
    "serviceMenu": {
      "jquery": "$('#nav li.nav-pro a[data-lid=\"ubr_prd\"]')"
    },
    "tvHomeTheaterSubMenu": {
      "jquery": "$('#nav a[data-lid=\"ubr_tv1\"]')"
    },
    "tvLink": {
      "jquery": "$('#nav a[data-lid=\"ubr_tv_tvs\"]')"
    },
    "cart": {
      "jquery": "$('#cart')"
    },
    "cartItems": {
      "jquery": "$('#cart a span strong')"
    }
  }
}
```

That's it! The **bold** words in the above example are key components that are used by JazzAutomation. The following table details the key components.

Quit stalling and show us!

OK, ok. Let's consider Best Buy. The main best buy site is located at www.bestbuy.com. Let's call



Configuration Information

Key Components

Component	Required	Blank Allowed	Omission Allowed	Notes
pageName	Yes	No	No	By convention, the value should match the file name.
Title	No	Yes	Yes	
urlExtension	No	Yes	No	
keyDomElementName	No	Yes	Yes	Recommended to ensure the page has loaded fully, prior to moving forward with the scenario
switchWindows	No	Yes	Yes	
domElements	Yes	No	No	

Key Components Definitions

pageName:

The pageName is used heavily in 'feature' configurations. A meaningful name is highly recommended.

title:

The title should match or partially match the <title> element on the HTML page. It can be left blank if it is too long or contains any special characters.

urlExtension:

The urlExtension should be the full page URL. A partial value is also acceptable.

keyDomElementName:

The keyWebComponentName is the name of a webComponent which is displayed on a fully loaded/displayed page. A web page is not considered to be fully loaded until this component is visible. This attribute is optional, however highly recommended.

switchWindows:

Optional attribute that ensures that the focus is

changed to deal with child windows, etc. Set this attribute to 'true' if you need to switch from one window to another.

domElements

The most critical elements of the page are the webComponents. These are the DOM objects used during the test runs. WebComponent names should be meaningful so that the test writers understand each component's purpose. The rule of thumb is to use the information shown on the screen. For example, if a button on the screen is shown as "Add To Cart", a good webComponent name is "addToCartButton".

At times, you will find that the different web browsers will interpret jquery, xpath, etc differently, so JazzAutomation has added a mechanism to account for these differences. Page configurators can configure a browser specific webComponent by adding the browser name to the end of the component name. For example an component named 'secondProduct' could be altered to secondProduct-chrome to indicate that it should be used only for Chrome.



More Configuration Information

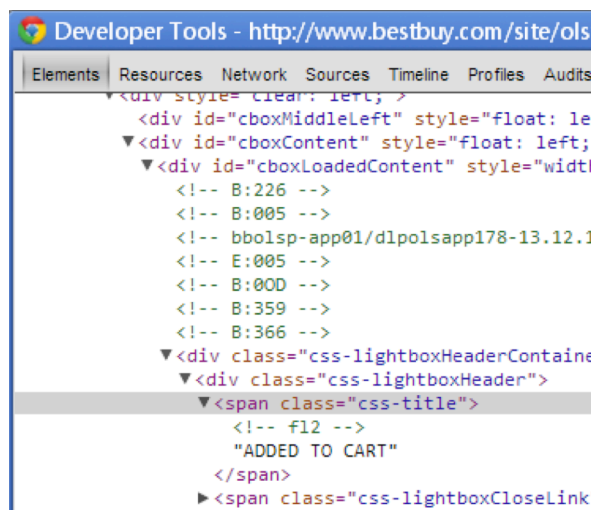
Configuring a Web Component

To configure a web component, a web component must be located. To locate a DOM element on a webpage, use jquery selector to do it (please read the jquery selector document here:

http://www.w3schools.com/jquery/jquery_ref_selectors.asp)

```
"addToCartPopupTitle" :
{
  "jquery": "$('div#cboxContent span.css-
title:first')"
```

A good tool is the Chrome developer tool (Ctrl-Shift-I in windows). You can click the component and examine the html tagging, css and other properties.



After you have the jquery selector, once can verify it by executing the jquery on the developer tool console.


Alternatively, one can use one the selenium built-in approach. JazzAutomation supports the following Selenium approaches: name, linkText, className, xpath, cssSelector, partialLinkText, and tagName.

Examples:

```
"addToCartPopupTitle" :
{
  "cssName": "css-title"
}

"secondProduct-chrome" :
{
  "xpath": "//div[@class='item-pic']/a[1]"
}
```





jazzautomation.com