



---

## **Variables & Common Validation**

**27–28 Apr. 2019**

**4–5 May 2019 @IT MJU**

**Assoc. Rangsit Sirirangsi**

**[www.indythaitester.com](http://www.indythaitester.com)**

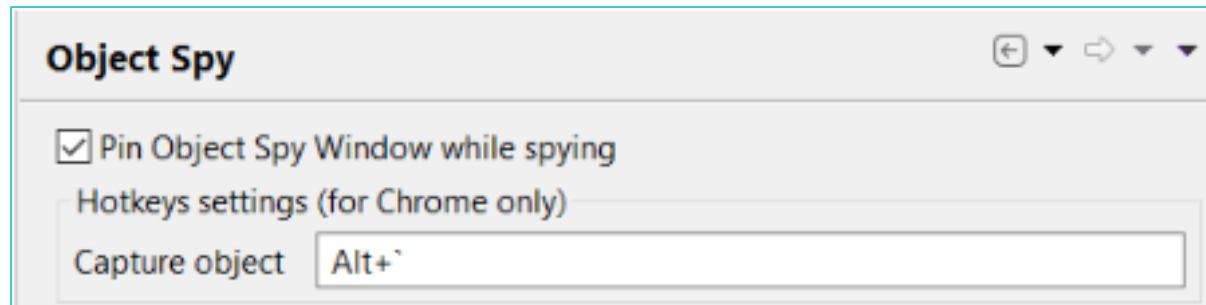
## Add UI elements to Object Repository

---

- กลไกแบบ Record และ Playback มีข้อเสียสองประการได้แก่
  - สคริปต์สำหรับการทดสอบที่ถูกสร้างขึ้นจะมีลักษณะเป็น Hard Code ซึ่งต้องมีการแก้ไขข้อมูลการทดสอบอยู่เสมอ
  - Test Object ภายใน Object Repository มีจำนวนเพิ่มขึ้นตามความซับซ้อนของการทดสอบ
- ดังนั้นในทางปฏิบัติเพื่อลดขนาดของ Object Repository ลงผู้ใช้นักนิยมใช้การสร้าง Test Object สองวิธีดังนี้
  - วิธีแรกใช้กลไกที่เรียกว่า Object Spy ใน Katalon Studio
  - วิธีที่สองเป็นการเพิ่ม Property เข้าสู่ Test Object โดยตรง โดยใช้เครื่องมือค้นหา Locator ของ HTML เช่น ChroPath

# Object Spy

- ผู้ใช้สามารถใช้ Object Spy ช่วยในการตรวจจับและบันทึก Test Object พร้อมกับค่าของ Property ที่กำหนดไว้เข้าสู่ Object Repository ได้โดยอัตโนมัติ
- โดยปกติแล้วค่า Web Spy short cut จะถูกตั้งค่าไว้ดังรูป



- ผู้ใช้สามารถแก้ไขค่าคีย์ดังกล่าวได้ โดยการคลิกเลือกที่เมนู Windows => Katalon Studio Preferences และเลือกที่หัวข้อ Object Spy เพื่อระบุค่าใหม่

# Property : Selection Method

---

- Katalon Studio ยอมให้ผู้ใช้เลือกวิธีการระบุตำแหน่งของ Test Object ได้ดังนี้
  - **XPath** เป็นวิธีแบบ Default ที่ Studio เลือกใช้ โดยเลือกใช้ค่าหลาย ๆ แบบร่วมกับ property ต่าง ๆ เพื่อใช้ระบุตำแหน่งของ Test Object ได้อย่างถูกต้อง
  - **Attribute** เป็นวิธีการระบุตำแหน่งส่วนประกอบต่าง ๆ ของ HTML อาทิ id, name, class, และ value เป็นต้น ซึ่งอยู่ภายใต้ Tag ต่าง ๆ ที่กำหนดไว้ใน HTML
  - **CSS** เป็นการระบุตำแหน่งส่วนประกอบต่าง ๆ ของ HTML อีกแบบหนึ่งที่มีลักษณะใกล้เคียงกับ XPath แต่จะแตกต่างกันตรงรูปแบบและวิธีการเรียกใช้

# Object Spy

## Object Spy with XPath

The screenshot shows the Object Spy tool interface. The URL bar displays `https://www.katalon.com/`. The **CAPTURED OBJECTS** section shows a tree view with `Page_Katalon Studio Best automated` expanded, revealing `h1_Katalon Studio`. In the **OBJECT PROPERTIES** section, the **Object Name** is `h1_Katalon Studio`. The **Selection Method** is set to **XPath** (indicated by a selected radio button). Below this, a table lists properties for the selected object:

Name	Value
xpath:idRelative	//body[@id='page-top']/div[2]/div/div/...
xpath:neighbor	(.//*[normalize-space(text()) and normal...
xpath:neighbor	(.//*[normalize-space(text()) and normal...
xpath:neighbor	(.//*[normalize-space(text()) and normal...
xpath:position	//h1

The **Selector Editor** at the bottom shows the XPath expression: `//body[@id='page-top']/div[2]/div/div/div/div/div/h1`. A **Verify and Highlight** button is located at the bottom right.

## Object Spy with Attribute

The screenshot shows the Object Spy tool interface. The URL bar displays `https://www.katalon.com/`. The **CAPTURED OBJECTS** section shows a tree view with `Page_Katalon Studio Best automated` expanded, revealing `h1_Katalon Studio`. In the **OBJECT PROPERTIES** section, the **Object Name** is `h1_Katalon Studio`. The **Selection Method** is set to **Attributes** (indicated by a selected radio button). Below this, a table lists properties for the selected object:

Name	Condition	Value	
tag	equals	h1	<input checked="" type="checkbox"/>
class	equals	brand-heading	<input type="checkbox"/>
text	equals	Katalon Studio	<input checked="" type="checkbox"/>
xpath	equals	id("page-top")/div[@class="intro-header...	<input type="checkbox"/>

The **Selector Editor** at the bottom shows the Attribute-based XPath expression: `//h1[(text() = 'Katalon Studio' or . = 'Katalon Studio')]`. A **Verify and Highlight** button is located at the bottom right.

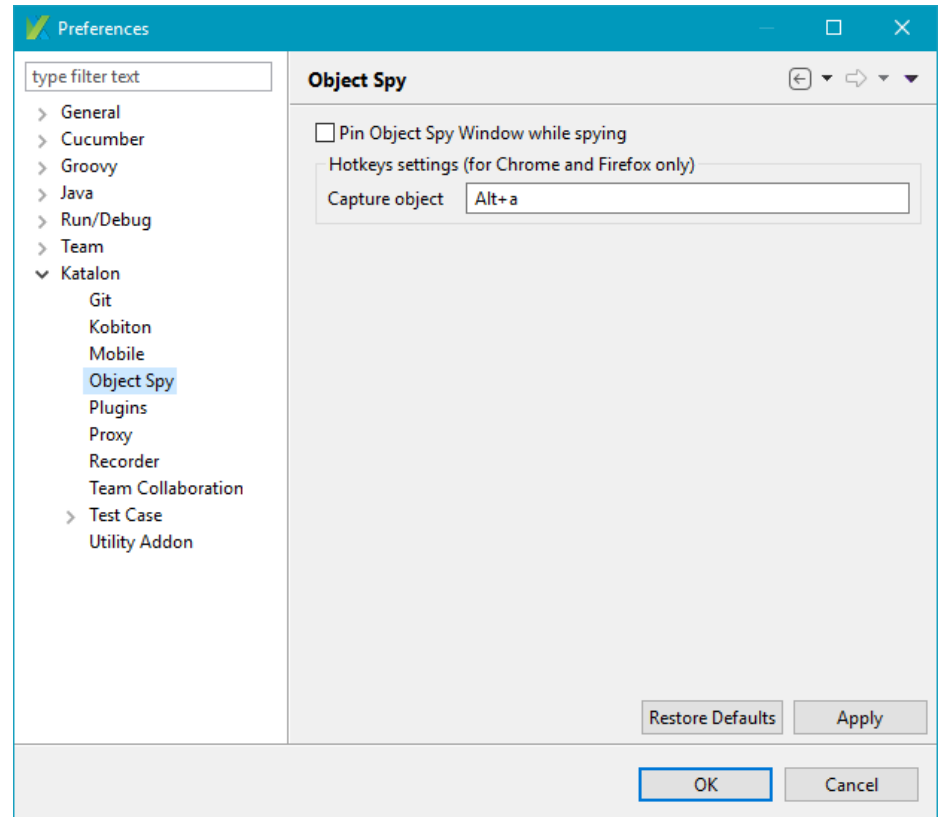
# Object Spy : Locator

---

- **Step 1:** คลิกเลือก **Spy Web** จากเมนูหลักของ Katalon Studio
- **Step 2:** หน้าต่าง Object Spy จะปรากฏขึ้น
- **Step 3:** กรอกข้อมูล URL และเลือกชนิดบราวเซอร์ที่ต้องการ คลิก **Start**
  - จากนั้น Katalon Studio จะเปิดบราวเซอร์และไปยัง Url ที่ถูกระบุไว้
- **Step 4:** การตรวจจับ Test Object สามารถเลื่อนเมาส์ไปยังตำแหน่งที่ต้องการ จะปรากฏ highlight ขึ้น
  - กดคีย์ที่ใช้ในการตรวจจับ Test Object ได้แก่ Alt + ~ จะปรากฏ highlight สีเขียว ซึ่งแสดงถึง Test Object ถูกตรวจจับไว้แล้ว

# Captured Object : HotKeys Setting

- การตั้งค่า HotKeys เพื่อใช้ในการตรวจจับค่า Property ของ Test Object สามารถทำได้โดยการคลิกเลือกเมนู
- Window=> Katalon Studio Preferences => Katalon => Object Spy



## Demo Manual Mode : Ebay

---

- สร้างสคริปต์โดยเริ่มต้นจากการใช้ Object Spy ช่วยในการ Add Property ลงใน Test Object ที่ต้องการใช้งาน
- จากนั้น Add คำสั่งพร้อมทั้งระบุ Test Object ดัง Test Case ต่อไปนี้

TC #	Test Step	Test Data
1	Open Browser	Chrome
2	Go to the Application URL	www.ebay.com
3	Enter Values the search	Iphone X
4	Click Search Button	Yes
5	Get Text No of results	output
6	Print	output
7	Close Browser	Yes





# Locator : Relationship

---

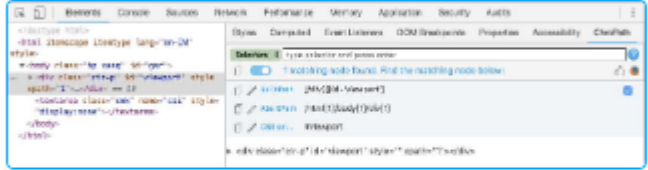
Name	Tag	Attribute	Option
Link	a	href	id, name, class
Text Field	INPUT	Type = "text"	id, name, class
Radio Button	INPUT	Type = "radio"	id, name, class
CheckBox	INPUT	Type = "checkbox"	id, name, class
Button	INPUT	Type = "submit"	id, name, class
Select Box	Select	value	id, name, class
Text	div, span, p, h1, h2, label		id, name, class
Image	Img	src	id, name, class

# Install : ChroPath

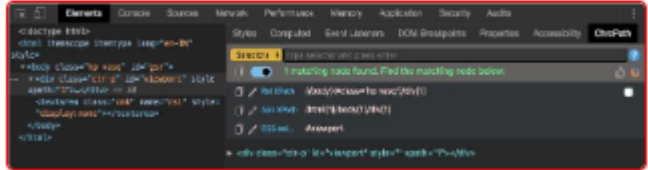
- ChroPath เป็นเครื่องมือช่วยในการพัฒนาเว็บที่ผู้ใช้สามารถตรวจสอบค่า Locator ต่าง ๆ ภายในเว็บเพจ
- การติดตั้งสามารถใช้วิธีการเดียวกับการติดตั้ง add-ons ทั่ว ๆ ไปดังขั้นตอนต่อไปนี้
  - เปิดบราวเซอร์ Chrome และไปที่ WebStore
  - ค้นหา add-ons ชื่อ “ChroPath” และติดตั้ง add-ons ดังกล่าวตามลำดับ
  - รีสตาร์ท Chrome

 **ChroPath** video 


- After adding ChroPath, restart Chrome browser.
- Open devtools panel.
  - Right-click anywhere on the webpage.
  - Click **Inspect**.
- **Default Theme**



- **Dark Theme**



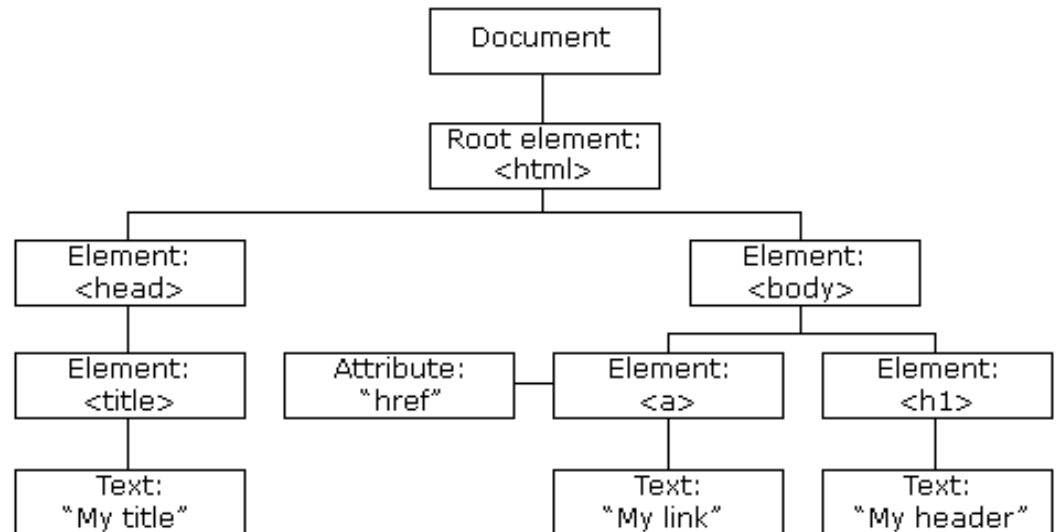
Please click to join [ChroPath Community Design Team](#)



# DOM, XPath และ CSS

- DOM เป็นการอธิบายถึงการจัดเอกสาร HTML ให้อยู่ในรูปของโครงสร้างต้นไม้ เพื่อแสดงความสัมพันธ์ระหว่างส่วนประกอบต่างๆ ตามลำดับชั้นที่ปรากฏ ผู้ใช้สามารถเข้าถึงได้โดยการระบุตำแหน่งอย่างถูกต้อง

```
<html>
<head>
<title>My title</title>
</head>
<body>
<h1>My header</h1>
<a href="test.html">My link</a>
</body>
</html>
```



# Xpath : Absolute & Relative

---

- XPath ย่อมาจาก XML Path เป็นภาษาที่ใช้ในการค้นหาส่วนประกอบต่าง ๆ ภายในเอกสาร XML
- XPath ถูกนำมาใช้ในการระบุตัวตนของเว็บออบเจกต์โดยขึ้นอยู่กับตำแหน่งภายในส่วนประกอบต่าง ๆ ของ HTML ในเว็บเพจได้
- การระบุ XPath ใน Selenium สามารถทำได้ดังนี้
  - **Absolute Xpath** : เริ่มต้นจาก root โหนด โดยใช้สัญลักษณ์ forward slash (/)
    - /html/head/body/table/tbody/tr/th
  - **Relative Xpath** : เป็นการค้นหาเส้นทางที่เริ่มต้นจากโหนดที่ผู้ใช้กำหนด โดยใช้สัญลักษณ์ Double forward slash(//)
    - //table/tbody/tr/th

# Xpath Terminology

---

- Tag, Node , Root Node, Parent Node
- Subtag, Subnode, Childnode

```
<div class="_58mf">  
  id="reg_box" class="registration_resign">  
    <form id="reg">  
      <input type="hidden" value="AVo6KlyY" name="Lsd">  
        <div id="reg_form_box" class="Large_form">  
        </div>  
      </input>  
    </form>  
  </div>
```

# Xpath Selecting Node

---

- `//<node>` :- ใช้สำหรับค้นหาและเลือกทุก ๆ node ภายในเพจปัจจุบัน
- `/<node>` :- ใช้สำหรับเลือก immediate child
- `//<node>[index]` :- ใช้สำหรับเลือกโหนดจาก index
- `//<node>/.` :- ใช้สำหรับเลือก node ปัจจุบัน
- `//<node>/..` :- ใช้สำหรับเลือก parent node จากโหนดปัจจุบัน

`www.facebook.com`

`//div[@id='reg_form_box']/div[1]`

# ChroPath : Locator

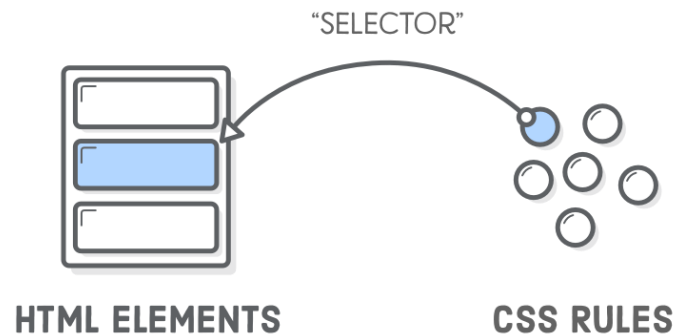
---

- การค้นหา Locator ของส่วนประกอบต่าง ๆ ของ HTML
  - **Id**
  - **Name**
  - **Tag**
  - **Class**
  - **CSS**
- XPath สามารถใช้ร่วมกับค่าแอททริบิวต์ต่าง ๆ ได้ เช่น ID, Name, Value, Type เป็นต้น โดยมีรูปแบบดังนี้

**`//*[@attribute='attribute value']`**

# CSS Locator

- CSS ใช้การระบุ element ในรูปของ tag, id, class, และ attributes ร่วมกับ Attribute ในการค้นหาส่วนประกอบของ HTML ภายในเพจ ซึ่งการทำงานในลักษณะนี้จะถูกใช้เป็น Pattern สำหรับการค้นหา
- CSS ใช้แพตเทิร์นดังกล่าวในการค้นหาส่วนประกอบของ HTML
- การค้นหาในลักษณะนี้จะซับซ้อนกว่าแบบปกติ แต่จะเหมาะสมอย่างยิ่งในกรณีที่ Locator ที่ถูกระบุไม่สามารถค้นหาได้จาก ID หรือ name





# CSS vs. Xpath

- เปรียบเทียบรูปแบบการระบุ Element ด้วยวิธีแบบ CSS และ Xpath ดังนี้
  - **CSS:** tag[attribute = Value]
  - **Xpath:** //tag[@attribute='Value']

Attribute	CSS	Xpath
id	input#email	//input[@id='email']
Class name	input.inputtext	//input[@class='inputtext']
Attribute	input[name=lastName]	//input[@name='lastName']
Multiple attribute	input[id='signIn'][value='Sign in']	//input[@id='signIn'][@value='Sign in']
Contains	input[value*='n i']	//input[contains(value,'n i')]
Starts with	input[id^='Em']	//input[starts-with(@id, 'Em')]
Ends with	input[id\$='wd']	//input[ends-with(@id, 'wd')]

# CSS Locator

---

- `<input id="txtName" class="textboxcss" tabindex="1" type="text">`
  - Tag and ID : syntax `css=tag#id` (เครื่องหมาย Hash(#) อ้างอิง id)
    - `input#txtName`
  - Tag and Class: `css=tag.classname` (เครื่องหมาย dot(.) ใช้อ้างอิง class)
    - `<span class="gb_8a gbii"></span> => css=span.gb_8a.gbii`
- ใช้เครื่องหมาย > แทน / หรือ Direct Child
  - Xpath = `//div/a`      Css = `div>a`
- `<input value="Reading" type="checkbox">`
  - Tag and Attribute : `css=tag[attribute='value']`
    - `input[type='checkbox']`    หรือ `input[value='Reading']`

# Add Property Manually

- ใน Test Explorer เริ่มต้นจาก Object Repository ด้วยการคลิกเลือก New => Test Object พร้อมกับการระบุชื่อ Test Object
- จากนั้นจะปรากฏหน้าต่าง **Add property** เพื่อให้ผู้ใช้กำหนดค่า Property

▼ Settings

Have parent object?

☒ No

☐ Parent iframe

☐ Shadow Root Parent

☐ Use relative path for image

Image

Selection Method ? ☐ XPath ☒ Attributes ☐ CSS

Object's Properties

+

 Add

✖

 Delete

🧹

 Clear

Name	Condition	Value	<input checked="" type="checkbox"/> Detect object by?	

## Add Property : XPath

- ชื่อของ Property จะต้องไม่ซ้ำกันภายใน Test Object เดียวกัน
- ในกรณีที่ผู้ใช้เลือก Selection Method เป็น XPath หรือ CSS สามารถระบุค่า Locator ตามรูปแบบที่กำหนดไว้ได้โดยตรงในส่วน Selector Editor เช่น

Selection Method ? ☒ XPath ☐ Attributes ☐ CSS

Object's Xpaths

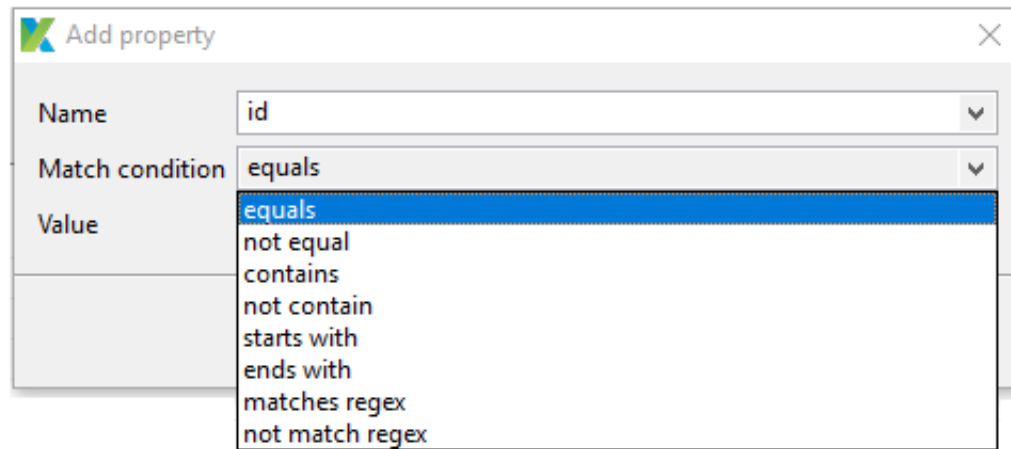
Name	Value

Selector Editor

```
//div[@id='username']
```

## Add Property : Attribute

- ส่วนในกรณีที่เลือกกำหนดค่า Property แบบ Attribute จะปรากฏหน้าต่าง Add property ดังนี้



- Name : เป็นชื่อของ Property ที่กำหนดไว้เพื่อให้ผู้ใช้เลือก ได้แก่ xpath, css, class, id, title และ name
- Match condition : เป็นเงื่อนไขที่ใช้ในการค้นหา Test Object ภายใน AUT

# Object Property : Attribute id

- ในกรณีที่เลือกกำหนดค่า Object Property ด้วยค่า Attribute ผู้ใช้จะต้องเลือกคลิกเลือก checkbox ส่วนที่เป็น Detect object by?
- หลังจากการเพิ่มค่า Object Property เสร็จสิ้นแล้ว ผู้ใช้ต้องบันทึกข้อมูลก่อนการเรียกใช้เสมอ

**Object's Properties**

Selection Method ? ☒ Basic ☐ CSS ☐ XPath

+ Add 🗑 Delete 🧹 Clear

Name	Condition	Value	<input checked="" type="checkbox"/> Detect object by?
id	equals	testControl	<input checked="" type="checkbox"/>

## Demo : wp-login

- สร้างสคริปต์ใน Manual Mode โดยใช้ ChroPath ในการ Add Property เข้าสู่ Test Object ที่ต้องการใช้งานดัง Test Case ที่มีรายละเอียดต่อไปนี้

TC #	Test Step	Test Data
1	Open Browser	Chrome
2	Go to the Application URL	demosite.center/wordpress/wp-login.php
3	Enter Username	admin
4	Enter Password	demo123
5	Click Login Button	yes
6	Click comment menu	yes
7	Get text 'in response to'	output
8	Print	output
8	Close Browser	Yes

# Katalon Studio : Wait

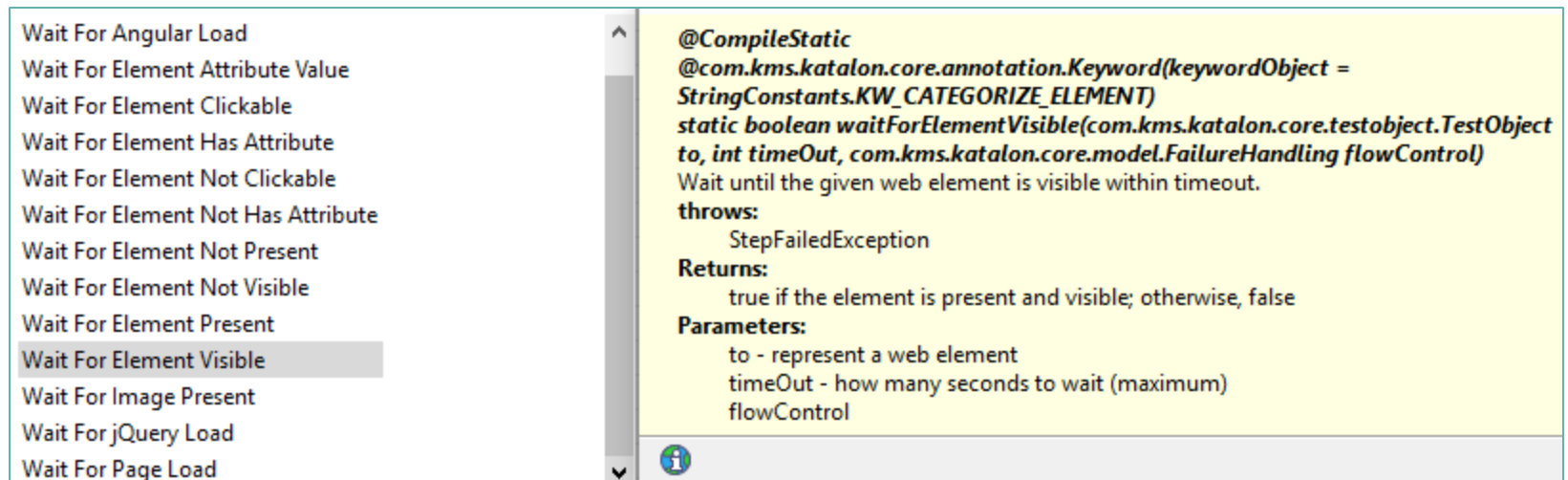
---

- ความล้มเหลวของการทดสอบอัตโนมัติที่เกิดจากเวลาหยุดรอไม่เพียงพอส่วนใหญ่มาจาก 2 สาเหตุ ได้แก่
  - เกิดจากความเร็วของเน็ตเวิร์ค ความล่าช้าของฐานข้อมูลที่มีการร้องขอหรือระบบต้องการเวลามากขึ้นในการโปรเซสและตอบสนอง
  - เกิดจากการหยุดรวบรวมเซอร์โพลบางส่วน ของ HTML ยังไม่เสร็จส่งผลให้สคริปต์ไม่สามารถทดสอบต่อได้
- ดังนั้นเพื่อป้องกันปัญหาดังกล่าว Katalon Studio จึงได้ออกแบบคำสั่งสำหรับการหยุดการทำงานไว้ 2 รูปแบบ ดังนี้
  - `WebUI.delay()` เป็นเวลาหยุดรอแบบคงที่ตามจำนวนที่กำหนดไว้
  - `Wait For ...` เป็นเวลาหยุดรอแบบมีเงื่อนไข



# Wait for ...

- บ่อยครั้งที่การประมวลผลสคริปต์ล้มเหลวและเกิด exceptions อาทิ ‘NoSuchElementException’, ‘ElementNotFoundException’ เป็นต้น
- Katalon Studio จึงได้สร้างคีย์เวิร์ดประเภทที่ขึ้นต้นด้วย “Wait for...” เพื่อใช้แก้ปัญหาดังกล่าวดังรูป



The screenshot displays the Katalon Studio interface. On the left, a list of keywords is shown, with 'Wait for Element Visible' selected and highlighted. On the right, the details for this keyword are displayed on a yellow background. The details include the keyword's annotation, its static method signature, a description of its function, the exceptions it throws, its return value, and its parameters.

**Wait For Angular Load**  
**Wait For Element Attribute Value**  
**Wait For Element Clickable**  
**Wait For Element Has Attribute**  
**Wait For Element Not Clickable**  
**Wait For Element Not Has Attribute**  
**Wait For Element Not Present**  
**Wait For Element Not Visible**  
**Wait For Element Present**  
**Wait For Element Visible**  
**Wait For Image Present**  
**Wait For jQuery Load**  
**Wait For Page Load**

**@CompileStatic**  
**@com.kms.katalon.core.annotation.Keyword(keywordObject = StringConstants.KW\_CATEGORIZE\_ELEMENT)**  
**static boolean waitForelementVisible(com.kms.katalon.core.testobject.TestObject to, int timeout, com.kms.katalon.core.model.FailureHandling flowControl)**  
Wait until the given web element is visible within timeout.  
**throws:**  
    StepFailedException  
**Returns:**  
    true if the element is present and visible; otherwise, false  
**Parameters:**  
    to - represent a web element  
    timeout - how many seconds to wait (maximum)  
    flowControl

## Wait for Page Load

- เป็นคำสั่งในรูปแบบของเมธอดเพื่อหยุดรอเพจโหลดภายในเบราว์เซอร์ภายในเวลาที่กำหนดไว้ในหน่วยวินาที

**static void waitForPageLoad(int seconds, [FailureHandling](#) flowControl)**

Param	Param Type	Mandatory	Description
second	int	Required	The number of seconds to wait for page load.
flowControl	FailureHandling	Optional	Specify <a href="#">failure handling</a> schema to determine whether the execution should be allowed to continue or stop.

**WebUI.openBrowser('http://demoaut.katalon.com/')**

**WebUI.waitForPageLoad(5)**

# Wait For Element Present

- เป็นคำสั่งในรูปของเมธอดเพื่อหยุดรอนกว่าส่วนประกอบของ HTML จะถูกโหลดขึ้นภายใน DOM ภายในเวลาที่กำหนดไว้ในหน่วยวินาที

**static boolean waitForElementPresent([TestObject](#) to, int timeOut)**

Param	Param Type	Mandatory	Description
to	TestObject	Required	Represent a web element.
timeOut	int	Required	Maximum period of time (in seconds) that system will wait to return a result.
flowControl	FailureHandling	Optional	Specify <a href="#">failure handling</a> schema

**'Wait for btn\_Login to be present in 10s'**

**WebUI.waitForElementPresent(findTestObject('Page\_Login/btn\_Login'), 10)**

# Wait For Element Visible

- เป็นคำสั่งในรูปแบบของเมธอดเพื่อหยุดรอนกว่าส่วนประกอบของ HTML จะถูกโหลดขึ้นภายใน DOM และมองเห็นได้ใน UI ภายในเวลาที่กำหนด

**static boolean waitForElementVisible([TestObject](#) to, int timeOut)**

Param	Param Type	Mandatory	Description
to	TestObject	Required	Represent a web element.
timeOut	int	Required	Maximum period of time (in seconds) that system will wait to return the result.
flowControl	FailureHandling	Optional	Specify <a href="#">failure handling</a> schema

**'Wait for btn\_Login to be visible in 10s'**

**WebUI.waitForElementVisible(findTestObject('Page\_Login/btn\_Login'), 10)**

# Wait For Element Not Visible

- เป็นคำสั่งในรูปของเมธอดเพื่อหยุดรอนกว่าส่วนประกอบของ HTML จะหายไปในช่วงเวลาที่กำหนด

**static boolean waitForElementNotVisible([TestObject](#) to, int timeOut)**

Param	Param Type	Mandatory	Description
to	TestObject	Required	Represent a web element.
timeOut	int	Required	Maximum period of time (in seconds) that system will wait to return the result.
flowControl	FailureHandling	Optional	Specify <a href="#">failure handling</a> schema

**'Wait for \'Make Appoint\' button NOT visible in 20 seconds'**

**WebUI.waitForElementAttributeValue(findTestObject('Page\_CuraHomepage/btn\_MakeAppointment'), 'id', 'btnMakeAppointment', 20)**

# Wait For Element Clickable

- เป็นคำสั่งในรูปของเมธอดเพื่อหยุดรอนกว่าผู้ใช้สามารถคลิกบนส่วนประกอบของ HTML ที่ถูกระบุได้ในช่วงเวลาที่กำหนด

**static boolean waitForElementClickable([TestObject](#) to, int timeOut)**

Param	Param Type	Mandatory	Description
to	TestObject	Required	Represent a web element.
timeOut	int	Required	Maximum period of time (in seconds) that system will wait to return the result.
flowControl	FailureHandling	Optional	Specify <a href="#">failure handling</a> schema

**'Wait for 'Make Appoint\' button has id : \'btnMakeAppointment\' in 20 seconds'**  
**WebUI.waitForElementClickable(findTestObject('Page\_CuraHomepage/btn\_MakeAppointment'), 20)**

## Demo : OrangeHrm

- จาก Test Script จาก Test Plan ที่กำหนด โดยใช้ ChroPath ช่วยในการระบุตำแหน่งของ Element จากรายละเอียดของ Test Case ดังต่อไปนี้

TC #	Test Step	Test Data
1	Open Browser	Firefox
2	Go to the Application URL	<a href="https://opensource-demo.orangehrmlive.com/">https://opensource-demo.orangehrmlive.com/</a>
3	Enter username	Admin
4	Enter Password	admin123
5	Click Login Button	Yes
6	Click	Welcome Admin
7	Click Logout	Yes
8	Close Browser	Yes

# Variable

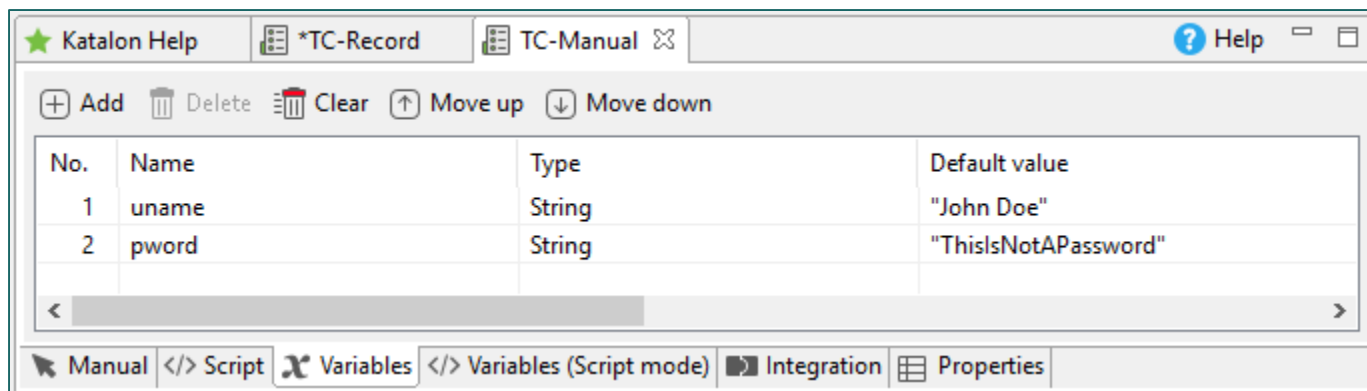
---

- Variable เป็นชื่อของตัวแปรที่ใช้สำหรับการจัดเก็บค่าข้อมูลที่สามารถเปลี่ยนแปลงได้ในช่วงของการประมวลผล ซึ่งจะเป็นประโยชน์อย่างยิ่งในการโปรแกรม
- ดังนั้น Katalon Studio จึงสนับสนุนการทำงานของ Variable 3 รูปแบบ ได้แก่
  - Private เป็นตัวแปรที่ประกาศไว้ภายใน Script View
  - Local เป็นตัวแปรที่กำหนดไว้ใน Test Case Editor สามารถเรียกใช้ในรูปแบบของพารามิเตอร์ผ่าน Manual หรือ Script View
  - Global เป็นตัวแปรที่กำหนดไว้ใน Profile เพื่อใช้งานสำหรับทุก ๆ Test Case ใน Project เดียวกัน



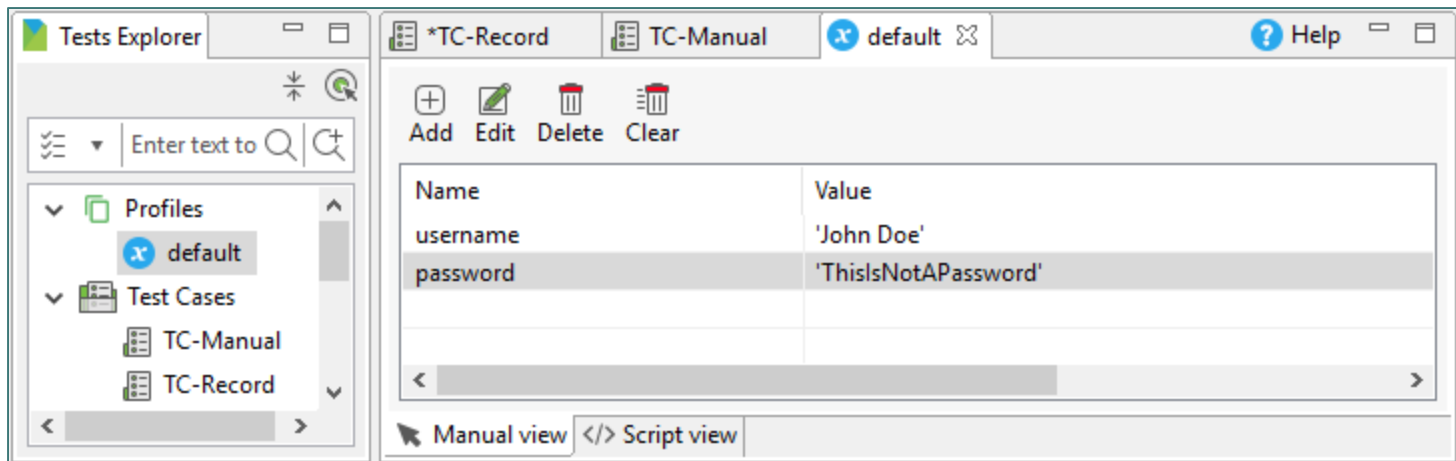
# Local Variables

- เป็นตัวแปรที่กำหนดไว้ใน Variables Tab ที่สามารถนำไปใช้ในรูปของพารามิเตอร์ใน Manual View หรือ Script View โดยการระบุชื่อตัวแปรที่กำหนดไว้
- ผู้ใช้สามารถเพิ่มตัวแปรได้โดยกดปุ่ม add รวมทั้งระบุ Name, Type และ Default Value ตามลำดับ



# Global Variables

- เป็น Variable ที่สามารถใช้งานได้ในทุก ๆ Test Case ภายใน Project เดียวกัน ส่วนการเรียกใช้สามารถทำได้โดยการระบุ GlobalVariable. นำหน้าตัวแปรที่ต้องการเสมอ
- ใน Katalon Studio รุ่นใหม่ ๆ Global Variable จะถูกกำหนดไว้ที่ default ภายใน Profiles



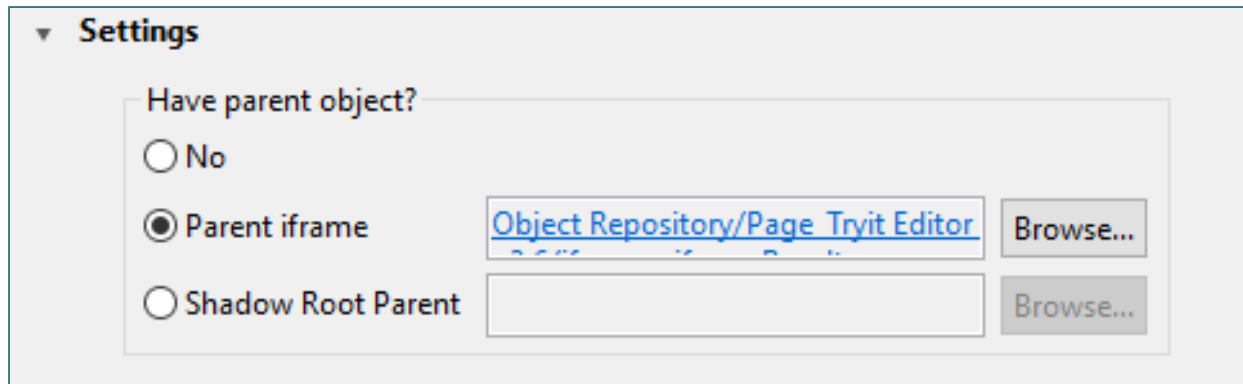
# What is Iframe?

---

- เป็น HTML document ที่ถูกฝังตัวไว้ใน HTML อื่น โดยปกติมักใช้ในการแทรกเนื้อหาจากแหล่งอื่นๆ เช่น โฆษณาลงในเว็บเพจ
- ในทางปฏิบัติแล้ว IFrame จะถูกกำหนดไว้ใน **<iframe>** tag ดังนั้นการเข้าถึงส่วนประกอบ HTML ที่อยู่ภายใน IFrame ไม่สามารถทำได้โดยตรง
- การจัดการ Iframe สามารถทำได้จากขั้นตอนดังนี้
  - Switch to the frame
  - Interact with web Element
  - Switch back to default content

## Settings: Parent iframe

- ดังที่ได้กล่าวมาแล้วว่าการเข้าถึงส่วนประกอบของ HTML ที่อยู่ภายใน IFrame โดยตรงไม่สามารถทำได้
- ดังนั้นจึงจำเป็นต้องตั้งค่า Test Object ในส่วนของ Parent IFrame ดังรูป



The screenshot shows a 'Settings' dialog box with a section titled 'Have parent object?'. It contains three radio button options: 'No', 'Parent iframe' (which is selected), and 'Shadow Root Parent'. To the right of the 'Parent iframe' option is a text input field containing the text 'Object Repository/Page Tryit Editor' and a 'Browse...' button. To the right of the 'Shadow Root Parent' option is an empty text input field and another 'Browse...' button.

- หากไม่มีการตั้งค่าแบบนี้ Test Case จะไม่สามารถทำงานได้
- Katalon จัดการเรื่อง Iframe โดยอัตโนมัติ

## Demo : IFrame

---

- สร้าง Test Script แบบ Manual โดยอาศัย ChroPath ช่วยในการ Add Property ของ Test Object จากรายละเอียดของ Test Case ดังต่อไปนี้

TC #	Test Step	Test Data
1	Open Browser	Chrome
2	Go to the Application URL	<a href="http://jqueryui.com/selectable/">http://jqueryui.com/selectable/</a>
3	Click Item 4	Yes
4	Get Text & save to	output
5	display Result	comment
6	Close Browser	Yes