

KELVERION AUTOMATION

INTEGRATION MODULE FOR WEB AUTOMATION

*FOR KELVERION RUNBOOK STUDIO AND
AZURE AUTOMATION*

User's Guide

Version 2.2

Kelverion Integration Module for Web Automation

Copyright © Kelverion Automation Ltd. All rights reserved.

Published: October 2019

FEEDBACK

Send suggestions and comments about this document to support@kelverion.com

CONTENTS

Introduction	4
Installation and Configuration	5
System Requirements	5
Installing the Integration Module from PowerShell Gallery	5
Installing the Integration Module Manually	6
Licensing the Integration Module	7
Web Automation Cmdlets	8
Cmdlet Reference	9
Add-WebCookie	9
Clear-WebCookie	12
Clear-WebElement	14
Clear-WebElementSelection	16
Close-WebAlert	19
Close-WebPage	21
Get-WebCookie	23
Get-WebElement	25
Get-WebElementOption	29
Get-WebElementSelection	30
Invoke-WebElementClick	32
Invoke-WebScript	34
New-WebBrowser	36
New-WebScreenShot	38
Open-WebPage	40
Read-WebAlertText	42
Read-WebElementAttribute	43
Read-WebElementCssProperty	45
Remove-WebBrowser	47
Remove-WebCookie	48
Send-WebAlertCredential	50
Send-WebAlertKeys	52
Send-WebElementKeys	54
Set-WebElementSearchTimeout	56
Set-WebElementSelection	59
Set-WebPageLoadTimeout	62
Set-WebTarget	65
Set-WebFrame	69
Step-WebPageBack	72
Step-WebPageForward	74
Submit-WebElement	76
Update-WebPage	78

INTRODUCTION

The Integration Module for Web Automation provides a set of ready built activities for interacting with web pages from within Azure Automation and the Keverion Runbook Studio. These are forms driven activities which enable you to automate operations within a Web Browser that simulate human interaction with the web application.

The Integration Module for Web Automation enables Azure Automation or any PowerShell script to interact with any web page or web application and is ideal for situations such as:

- In-house or commercial applications whose interface is via web browser
- B2B vendor sites to place orders for new devices, services, support tickets etc.
- Public or Vendor website containing data you want to integrate into an automated process i.e. patch downloading
- Publishing data to web sites via page creation forms i.e. intranet updates, wiki, service announcements, dashboard updates etc.

The Integration Module for Web Automation enables Azure Automation to become a rich Robotic Process Automation platform by delivering full application-to-application data extract, transfer and entry. This enables Azure Automation to deliver key RPA task automations such as;

- Contract setup
- Statement processing approval
- Report staging
- Auditing
- Reconciliations

INSTALLATION AND CONFIGURATION

The following sections outline how to deploy and configure the Keverion Integration Module for Web Automation.

SYSTEM REQUIREMENTS

The Integration Module for Web Automation requires the following software to be installed and configured prior to implementing the integration.

- Keverion Runbook Studio 3.3
- Microsoft .NET Framework 4.6.2
- Azure Automation
- Hybrid Worker Group with at least one Hybrid Worker
- Microsoft Internet Explorer or Google Chrome web browsers

INSTALLING THE INTEGRATION MODULE FROM POWERSHELL GALLERY

You can install the Integration module directly from the PowerShell Gallery to your Azure Automation Account and Hybrid Workers. If you want to build runbooks in Keverion Runbook Studio using the integration module you must also install it on your Runbook Studio host system.

To install the Integration Module for Runbook Studio:

1. Confirm you have the latest PowerShellGet module. See online documentation for more information: <https://www.powershellgallery.com/>
2. Start a PowerShell window as Administrator and run the command:
`Install-Module -Name Keverion.WebAutomation`

To install the Integration Module in Azure Automation:

1. Go to the PowerShell Gallery URL:
<https://www.powershellgallery.com/packages/Keverion.WebAutomation>
2. Click the button “**Deploy to Azure Automation**” and you will be redirected to the Microsoft Azure portal.
3. Select your Automation account and click **OK** to upload the module.

To install the Integration Module on a Hybrid Worker:

1. Confirm you have the latest PowerShellGet module on your Hybrid Worker computer.
See online documentation for more information: <https://www.powershellgallery.com/>
2. Start a PowerShell window as Administrator and run the command:
`Install-Module -Name Keverion.WebAutomation`

INSTALLING THE INTEGRATION MODULE MANUALLY

After download the integration module, you must upload it to your Portal and Worker hosts. If you want to build runbooks in Keverion Runbook Studio using the integration module you must also install it on your Runbook Studio host system.

To install the Integration Module for Runbook Studio:

1. Copy the **Keverion.WebAutomation.zip** to your computer.
2. Right click on the file and select properties. Unblock the file if it is blocked.
3. Unzip the **Keverion.WebAutomation.zip** file.
4. Copy the **Keverion.WebAutomation** folder to a location in the %PsModulePath% path.

To install the Integration Module in Azure Automation Portal:

1. Go to your automation account and select **Assets**.
2. In **Assets** select **Modules**.
3. In **Modules** select **Add a module**.
4. In **Upload File** field, select the **Keverion.WebAutomation.zip**.
5. Select **OK** to upload the module.

To install the Integration Module on a Hybrid Worker:

1. In the Keverion Runbook Studio or Microsoft Azure portal:
 - a. Create an Runbook with the following PowerShell: `$Env:PSModulePath`.
 - b. Test this runbook on the Hybrid Worker and record the output for use in Step 5.
2. Copy the **Keverion.WebAutomation.zip** to your Hybrid Worker computer.
3. Right click on the file and select properties. Unblock the file if it is blocked.
4. Unzip the **Keverion.WebAutomation.zip** file.
5. Copy the **Keverion.WebAutomation** folder to a location in the path returned in Step 1b.
Note: The Hybrid Worker uses its own unique value for %PSModulePath% and will differ from that of the system.

To install the Integration Module in SMA 2016:

1. Go to your Service Management Portal and select **Automation**.
2. In Automation select **Assets**, and then at the bottom select **Import Module**.
3. In the Import Integration Module dialog, select **Browse for file...** and select the **Kelverion. WebAutomation.zip**.
4. Select the Complete **checkmark** to upload the module.

To install the Integration Module on an SMA Automation Worker:

1. In the Service Management Portal:
 - a. Create an Runbook with the following PowerShell: `$Env:PSModulePath`.
 - b. Test this runbook on an Automation Worker and record the output for use in Step 5.
2. Copy the **Kelverion. WebAutomation.zip** to your Automation Worker computer.
3. Right click on the file and select properties. Unblock the file if it is blocked.
4. Unzip the **Kelverion. WebAutomation.zip** file.
5. Copy the **Kelverion. WebAutomation** folder to a location in the path returned in Step 1b.
Note: The Automation Worker uses its own unique value for `%PSModulePath%` and may differ from that of the system.

LICENSING THE INTEGRATION MODULE

After you install the integration module you must provide a valid Kelverion license before running any runbooks that contain activities from the integration module.

To deploy the integration module license file:

1. On the Hybrid Worker host system(s) where you deployed the Integration Module, open Windows Explorer.
2. Navigate to `%ProgramFiles(x86)%`
3. Create the folder 'Kelverion Automation\Licenses'.
4. Copy the .KAL license file to the 'Kelverion Automation\Licenses' folder.
5. Repeat for each Hybrid Worker where you deployed the Integration Module.

WEB AUTOMATION CMDLETS

This integration module adds the **Kelverion.WebAutomation** module to the **Cmdlets** item in the On-Premises panel in the Runbook Studio. This module contains the following Cmdlets:

- Add-WebCookie
- Clear-WebCookie
- Clear-WebElement
- Clear-WebElementSelection
- Close-WebAlert
- Close-WebPage
- Get-WebCookie
- Get-WebElement
- Get-WebElementOption
- Get-WebElementSelection
- Invoke-WebElementClick
- Invoke-WebScript
- New-WebBrowser
- New-WebScreenShot
- Open-WebPage
- Read-WebAlertText
- Read-WebElementAttribute
- Read-WebElementCssProperty
- Remove-WebBrowser
- Remove-WebCookie
- Send-WebAlertCredential
- Send-WebAlertKeys
- Send-WebElementKeys
- Set-WebElementSearchTimeout
- Set-WebElementSelection
- Set-WebFrame
- Set-WebPageLoadTimeout
- Set-WebTarget
- Step-WebPageBack
- Step-WebPageForward
- Submit-WebElement
- Update-WebPage

Add-WebCookie

Adds a cookie to the current page.

```
Add-WebCookie
  [-InputObject] <IWebBrowser[]>
  [-Cookie] <Cookie[]>
  [-PassThru]
  [<CommonParameters>]
```

```
Add-WebCookie
  [-InputObject] <IWebBrowser[]>
  [-Name] <String>
  [-Value] <String>
  [-Domain <String>]
  [-Path <String>]
  [-Expires <DateTime>]
  [-PassThru]
  [<CommonParameters>]
```

REQUIRED PARAMETERS

-Cookie

Specifies Cookie objects that represent the cookies to be added to the current web page.

Type:	Cookie
Position:	1
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-InputObject

Specifies IWebBrowser objects that represent the web browsers to act on. Enter a variable that contains the objects, or type a command or expression that gets the objects.

Type	IWebBrowser[]
Position	0

Default value	None
Accept pipeline input?	True (ByValue)
Accept wildcard characters?	False

-Name

Specifies the name of the cookie.

Type	String
Position	1
Default value	None
Accept pipeline input?	False
Accept wildcard characters?	False

-Value

Specifies the value to assign to the cookie.

Type	String
Position	2
Default value	None
Accept pipeline input?	False
Accept wildcard characters?	False

OPTIONAL PARAMETERS

-Domain

Specifies the URI for which the cookie is valid.

Type	String
Position	named
Default value	None
Accept pipeline input?	False
Accept wildcard characters?	False

-Expires

Specifies the expiration date and time for the cookie.

Type	Nullable<DateTime>
Position	named
Default value	None
Accept pipeline input?	False
Accept wildcard characters?	False

-PassThru

Returns an object that represents the web browser. By default, this cmdlet does not generate any output.

Type:	SwitchParameter
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-Path

Specifies the URIs to which the cookie applies.

Type	String
Position	named
Default value	None
Accept pipeline input?	false
Accept wildcard characters?	false

INPUTS

Kelverion.WebAutomation.WebBrowser

You can pipe web browser objects to the cmdlet.

OUTPUTS

Kelverion.WebAutomation.WebBrowser

This cmdlet generates a `Kelverion.WebAutomation.IWebBrowser` object that represents the web browser, if you use the `PassThru` parameter. Otherwise, this cmdlet does not generate any output.

Clear-WebCookie

Removes all cookies from the current web browser page.

```
Clear-WebCookie
  [-InputObject] <IWebBrowser[]>
  [-PassThru]
  [CommonParameters]
```

EXAMPLES

Example 1: Removes all cookies from the current web browser page

```
C:\PS> Clear-WebCookie -InputObject $Browser
```

REQUIRED PARAMETERS

-InputObject

Specifies IWebBrowser objects that represent the web browsers to act on. Enter a variable that contains the objects or type a command or expression that gets the objects.

Type:	IWebBrowser[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

OPTIONAL PARAMETERS

-Passthru

Returns an object that represents the web browser. By default, this cmdlet does not generate any output.

Type:	SwitchParameter
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

INPUTS

Kelverion.WebAutomation.WebBrowser

You can pipe web browser objects to the cmdlet.

OUTPUTS

Kelverion.WebAutomation.WebBrowser

This cmdlet generates a `Kelverion.WebAutomation.IWebBrowser` object that represents the web browser, if you use the `PassThru` parameter. Otherwise, this cmdlet does not generate any output.

Clear-WebElement

Clears the value of text entry elements, which are elements defined with INPUT or TEXTAREA tags. The cmdlet has no effect on other element types.

```
Clear-WebElement  
  [-InputObject] <IWebElement[]>  
  [-PassThru]  
  [<CommonParameters>]
```

REQUIRED PARAMETERS

-InputObject

Specifies IWebElement objects that represent the web page elements to act on. Enter a variable that contains the objects or type a command or expression that gets the objects

Type:	IWebElement[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

OPTIONAL PARAMETERS

-Passthru

Returns an object that represents the web browser. By default, this cmdlet does not generate any output.

Type:	SwitchParameter
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

INPUTS

Kelverion.WebAutomation.IWebElement

You can pipe web page elements to the cmdlet

OUTPUTS

Kelverion.WebAutomation.IWebElement

This cmdlet generates a `Kelverion.WebAutomation.IWebElement` object that represents web page elements, if you use the `PassThru` parameter. Otherwise, this cmdlet does not generate any output.

Clear-WebElementSelection

Clears the specified option from a SELECT element. Options can be identified by their index, displayed text or value.

```
Clear-WebElementSelection
  [-InputObject] <IWebElement[]>
  -Index <Int32>
  [-PassThru]
  [<CommonParameters>]
```

```
Clear-WebElementSelection
  [-InputObject] <IWebElement[]>
  -Text <String>
  [-PassThru]
  [<CommonParameters>]
```

```
Clear-WebElementSelection
  [-InputObject] <IWebElement[]>
  -Value <String>
  [-PassThru]
  [<CommonParameters>]
```

EXAMPLES

Example 1: Clear all web options that start with the keyword 'Test'

```
C:\PS>Get-WebElementOption $element | Where-Object {$_.Text -like 'Test*'} |
ForEach-Object {Set-WebElementSelection $element -Text $_.Text}
```

REQUIRED PARAMETERS

-Index

Specifies the index of the option to clear.

Type:	Int32
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-InputObject

Specifies IWebElement objects that represent the web page elements to act on. Enter a variable that contains the objects or type a command or expression that gets the objects

Type:	IWebElement[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

-Text

Specifies the text of the option to clear.

Type:	String
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

OPTIONAL PARAMETERS

-Passthru

Returns an object that represents the web browser. By default, this cmdlet does not generate any output.

Type:	SwitchParameter
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

INPUTS

Kelverion.WebAutomation.IWebElement

You can pipe web page elements to the cmdlet

OUTPUTS

Kelverion.WebAutomation.IWebElement

This cmdlet generates a `Kelverion.WebAutomation.IWebElement` object that represents web page elements, if you use the `PassThru` parameter. Otherwise, this cmdlet does not generate any output.

Close-WebAlert

Accepts or dismisses the active web alert window.

```
Close-WebAlert
  [-InputObject] <IWebBrowser[]>
  [-Dismiss]
  [-PassThru]
  [<CommonParameters>]
```

EXAMPLES

Example 1: Accept the active alert

```
C:\PS> Close-WebAlert -InputObject $Browser
```

Example 2: Dismiss the active alert

```
C:\PS> Close-WebAlert -InputObject $Browser -Dismiss
```

REQUIRED PARAMETERS

-InputObject

Specifies IWebBrowser objects that represent the web browsers to act on. Enter a variable that contains the objects, or type a command or expression that gets the objects.

Type:	IWebBrowser[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

OPTIONAL PARAMETERS

-Dismiss

Specifies that the alert should be dismissed. The default behavior is to accept the alert.

Type:	SwitchParameter
Position:	Named
Default value:	None

Accept pipeline input:	False
Accept wildcard characters:	False

-Passthru

Returns an object that represents the web browser. By default, this cmdlet does not generate any output.

Type:	SwitchParameter
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

INPUTS

Kelverion.WebAutomation.WebBrowser

You can pipe web browser objects to the cmdlet.

OUTPUTS

Kelverion.WebAutomation.WebBrowser

This cmdlet generates a `Kelverion.WebAutomation.IWebBrowser` object that represents the web browser, if you use the `PassThru` parameter. Otherwise, this cmdlet does not generate any output.

Close-WebPage

Closes the web page that is currently open in the web browser.

```
Close-WebPage
  [-InputObject] <IWebBrowser[]>
  [-PassThru]
  [<CommonParameters>]
```

REQUIRED PARAMETERS

-InputObject

Specifies IWebBrowser objects that represent the web browsers to act on. Enter a variable that contains the objects or get type a command or expression that gets the objects.

Type:	IWebBrowser[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

OPTIONAL PARAMETERS

-Passthru

Returns an object that represents the web browser. By default, this cmdlet does not generate any output.

Type:	SwitchParameter
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

INPUTS

Kelverion.WebAutomation.WebBrowser

You can pipe web browser objects to the cmdlet.

OUTPUTS

Kelverion.WebAutomation.WebBrowser

This cmdlet generates a `Kelverion.WebAutomation.IWebBrowser` object that represents the web browser, if you use the `PassThru` parameter. Otherwise, this cmdlet does not generate any output.

Get-WebCookie

Gets the cookie with the specified name.

```
Get-WebCookie
  [-InputObject] <IWebBrowser[]>
  [-Name] <String>
  [<CommonParameters>]
```

EXAMPLES

Example 1: Retrieve cookie with the name CustomerID

```
C:\PS> Get-WebCookie -InputObject $Browser -Name 'CustomerID'
```

REQUIRED PARAMETERS

-InputObject

Specifies IWebBrowser objects that represent the web browsers to act on. Enter a variable that contains the objects or type a command or expression that gets the objects.

Type:	IWebBrowser[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

-Name

Specifies the name of the cookie to retrieve.

Type:	String
Position:	1
Default value:	False
Accept pipeline input:	False
Accept wildcard characters:	False

OPTIONAL PARAMETERS

None

INPUTS

Kelverion.WebAutomation.WebBrowser

You can pipe web browser objects to the cmdlet.

OUTPUTS

System.Net.Cookie

Returns System.Net.Cookie objects that represent the cookie that was found

Get-WebElement

Searches the currently loaded page for web elements. The cmdlet supports searching by CSS class name, CSS selector, ID, link text, name, tag or XPATH.

```
Get-WebElement
  [-InputObject] <IWebSearchContext>
  -ClassName <String>
  [-First]
  [<CommonParameters>]
```

```
Get-WebElement
  [-InputObject] <IWebSearchContext>
  -CssSelector <String>
  [-First] [
  <CommonParameters>]
```

```
Get-WebElement
  [-InputObject] <IWebSearchContext>
  [-First]
  -Id <String>
  [<CommonParameters>]
```

```
Get-WebElement
  [-InputObject] <IWebSearchContext>
  [-First]
  -LinkText <String>
  [-Partial]
  [<CommonParameters>]
```

```
Get-WebElement
  [-InputObject] <IWebSearchContext>
  [-First]
  -Name <String>
  [<CommonParameters>]
```

```
Get-WebElement
  [-InputObject] <IWebSearchContext>
  [-First]
  -TagName <String>
  [<CommonParameters>]
```

```
Get-WebElement
  [-InputObject] <IWebSearchContext>
  [-First]
```

```
-Xpath <String>  
[<CommonParameters>]
```

REQUIRED PARAMETERS

-ClassName

Specifies the name of the CSS selector to use when searching for elements.

Type:	String
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-CssSelector

Specifies the name of the CSS selector to use when searching for elements.

Type:	String
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-Id

Specifies the element ID to use when searching for elements.

Type:	String
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-InputObject

Specifies the IWebBrowser or IWebElement objects to search.

Type:	IWebSearchContext[]
-------	---------------------

Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

-LinkText

Specifies the link text to use when searching for elements.

Type:	String
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-Name

Specifies the element name to use when searching for elements.

Type:	String
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-TagName

Specifies tag name to use when searching for elements.

Type:	String
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-Xpath

Specifies the XPATH expression to use when searching for elements.

Type:	String
-------	--------

Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

OPTIONAL PARAMETERS

-Partial

Specifies that a partial match can be used when searching by link text.

Type:	SwitchParameter
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

INPUTS

Kelverion.WebAutomation.IWebSearchContext

You can pipe IWebSearchContext objects, representing web browser pages or web elements, into the cmdlet.

OUTPUTS

Kelverion.WebAutomation.IWebElement

The cmdlet returns IWebElement objects that represent the web page elements that were found.

Get-WebElementOption

Gets the selections options from the SELECT element.

```
Get-WebElementOption  
  [-InputObject] <IWebElement[]>  
  [<CommonParameters>]
```

EXAMPLES

Example1: Select all of the element options that start with 'Test'

```
C:\PS>Get-WebElementOption $element -All | Where-Object {$_ .Text -like  
'Test*'} | ForEach-Object { Set-WebElementSelection $element -Text $_.Text }
```

REQUIRED PARAMETERS

-InputObject

Specifies IWebElement objects that represent the web page elements to act on. Enter a variable that contains the objects or type a command or expression that gets the objects

Type:	IWebElement[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

OPTIONAL PARAMETERS

INPUTS

Kelverion.WebAutomation.IWebElement

You can pipe web page elements to the cmdlet

OUTPUTS

Kelverion.WebAutomation.IWebElement

This cmdlet generates a Kelverion.WebAutomation.IWebElement object that represents selection options.

Get-WebElementSelection

Gets the selected options from the element.

```
Get-WebElementSelection
  [-InputObject] <IWebElement[]>
  [-All]
  [<CommonParameters>]
```

EXAMPLES

Example1: Get the first element that is selected.

```
C:\PS>$title = Get-WebElement $Browser -Name 'title' | Get-
WebElementSelection
```

In this example, Get-WebElement is used to find the 'title' SELECT element and then Get-WebElementSelection is used to determine which option is selected.

REQUIRED PARAMETERS

-InputObject

Specifies IWebElement objects that represent the web page elements to act on. Enter a variable that contains the objects or type a command or expression that gets the objects

Type:	IWebElement[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

OPTIONAL PARAMETERS

-All

Returns all the web element options that are selected. By default, the cmdlet only returns the first option that is selected.

Type:	SwitchParameter
Position:	Named

Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

INPUTS

Kelverion.WebAutomation.IWebElement

You can pipe web page elements to the cmdlet

OUTPUTS

Kelverion.WebAutomation.IWebElement

This cmdlet generates a `Kelverion.WebAutomation.IWebElement` objects that represents web page elements that were found.

Invoke-WebElementClick

Clicks the web page element. If the click causes a new page to load, the cmdlet will attempt to block until the page has loaded. After calling the cmdlet the web element that it was called with should be discarded unless you know that the element and the page will still be present.

```
Invoke-WebElementClick
  [-InputObject] <IWebElement[]>
  [-PassThru]
  [<CommonParameters>]
```

EXAMPLES

Example 1: Find the 'reset' button and click it.

```
C:\PS>Get-WebElement $Browser -Name 'reset' | Invoke-WebElementClick
```

In this example, the Get-WebElement cmdlet is used to retrieve the 'reset' button, which is then passed to the Invoke-WebElementClick to be clicked.

REQUIRED PARAMETERS

-InputObject

Specifies IWebElement objects that represent the web page elements to act on. Enter a variable that contains the objects or type a command or expression that gets the objects

Type:	IWebElement[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

OPTIONAL PARAMETERS

-Passthru

Returns an object that represents the web browser. By default, this cmdlet does not generate any output.

Type:	SwitchParameter
Position:	Named

Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

INPUTS

Kelverion.WebAutomation.IWebElement

You can pipe web page elements to the cmdlet

OUTPUTS

Kelverion.WebAutomation.IWebElement

This cmdlet generates a `Kelverion.WebAutomation.IWebElement` object that represents web page elements, if you use the `PassThru` parameter. Otherwise, this cmdlet does not generate any output.

Invoke-WebScript

Executes JavaScript code in the context of the currently selected frame or window.

```
Invoke-WebScript
  [-InputObject] <IWebBrowser[]>
  -Script <String>
  [-Arguments <Object[]>]
  [-PassThru]
  [<CommonParameters>]
```

REQUIRED PARAMETERS

-InputObject

Specifies IWebBrowser objects that represent the web browsers to act on. Enter a variable that contains the objects or type a command or expression that gets the objects.

Type:	IWebBrowser[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

-Script

Type:	String
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

OPTIONAL PARAMETERS

-Arguments

Type:	Object[]
Position:	Named
Default value:	None

Accept pipeline input:	False
Accept wildcard characters:	False

-Passthru

Returns an object that represents the web browser. By default, this cmdlet does not generate any output.

Type:	SwitchParameter
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

INPUTS

Kelverion.WebAutomation.WebBrowser

You can pipe web browser objects to the cmdlet.

OUTPUTS

Kelverion.WebAutomation.WebBrowser

This cmdlet generates a `Kelverion.WebAutomation.IWebBrowser` object that represents the web browser, if you use the `PassThru` parameter. Otherwise, this cmdlet does not generate any output.

New-WebBrowser

Creates a new web browser instance.

```
New-WebBrowser
  [-BrowserType <Kelveiron.WebAutomation.BrowserType>]
  [-PageLoadTimeout <Nullable[TimeSpan]>]
  [-SearchTimeout <Nullable[TimeSpan]>]
  [-Url] <Uri>]
  [<CommonParameters>]
```

EXAMPLES

Example 1: Open Chrome

```
C:\PS> $Browser = New-WebBrowser -BrowserType Chrome
```

Example 2: Open default browser and navigate to a web page.

```
C:\PS> $Browser = New-WebBrowser -Url 'https://acme.com'
```

REQUIRED PARAMETERS

None.

OPTIONAL PARAMETERS

-BrowserType

Specifies the type of browser window to open. If not specified, the cmdlet will open a new browser of whatever type is currently the default.

Type:	Kelverion.WebAutomation.BrowserType
Position:	Named
Default value:	Default
Accept pipeline input:	False
Accept wildcard characters:	False

-PageLoadTimeout

Specifies the amount of time to wait for pages to load.

Type:	TimeSpan
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-SearchTimeout

Specifies the amount of time to wait when searching for an element if it is not immediately present.

Type:	TimeSpan
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-Uri

Specifies the URL of the web page to navigate to when the browser opens.

Type:	Uri
Position:	0
Default value:	None
Accept pipeline input:	True
Accept wildcard characters:	False

INPUTS

System.String

You can pipe URIs into the cmdlet.

OUTPUTS

Kelversion.WebAutomation.WebBrowser

An object that represents the browser that was opened. The object should be stored in a variable to be used with other web automation cmdlets.

New-WebScreenShot

Takes a screenshot of the web page. The default image format is .png.

```
New-WebScreenShot
  [-InputObject] <IWebBrowser[]>
  [-FileName] <String>
  [-Format <ImageFormat>]
  [-PassThru]
  [<CommonParameters>]
```

EXAMPLES

Example 1: Take a screen shot and save it as a JPEG file.

```
C:\PS>New-WebScreenShot $Browser '.\ScreenShots\ScreenShot1.jpg' -Format
[System.Drawing.Imaging.ImageFormat]::Jpeg
```

REQUIRED PARAMETERS

-Filename

The path to the file used to save the screenshot.

Type:	String
Position:	1
Default value:	Non
Accept pipeline input:	False
Accept wildcard characters:	False

-InputObject

Specifies IWebBrowser objects that represent the web browsers to act on. Enter a variable that contains the objects or type a command or expression that gets the objects.

Type:	IWebBrowser[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

OPTIONAL PARAMETERS

-Format

The image format used to save the screenshot

Type:	System.Drawing.Imaging.ImageFormat
Position:	Named
Default value:	PNG
Accept pipeline input:	False
Accept wildcard characters:	False

-Passthru

Returns an object that represents the web browser. By default, this cmdlet does not generate any output.

Type:	SwitchParameter
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

INPUTS

Kelverion.WebAutomation.IWebBrowser

You can pipe web browser objects to the cmdlet.

OUTPUTS

Kelverion.WebAutomation.IWebBrowser

This cmdlet generates a Kelverion.WebAutomation.IWebBrowser object that represents the web browser, if you use the PassThru parameter. Otherwise, this cmdlet does not generate any output.

Open-WebPage

Loads a web page into the web browser.

```
Open-WebPage
  [-InputObject] <IWebBrowser[]>
  [-Url] <Uri>
  [-PassThru]
  [<CommonParameters>]
```

REQUIRED PARAMETERS

-InputObject

Specifies IWebBrowser objects that represent the web browsers to act on. Enter a variable that contains the objects or type a command or expression that gets the objects.

Type:	IWebBrowser[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

-Url

The URL of the web page to load.

Type:	Uri
Position:	1
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

OPTIONAL PARAMETERS

-Passthru

Returns an object that represents the web browser. By default, this cmdlet does not generate any output.

Type:	SwitchParameter
-------	-----------------

Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

INPUTS

Kelverion.WebAutomation.IWebBrowser

You can pipe web browser objects to the cmdlet.

OUTPUTS

Kelverion.WebAutomation.IWebBrowser

This cmdlet generates a `Kelverion.WebAutomation.IWebBrowser` object that represents the web browser, if you use the `PassThru` parameter. Otherwise, this cmdlet does not generate any output.

Read-WebAlertText

Reads the window text from the active alert window.

```
Read-WebAlertText  
  [-InputObject] <IWebBrowser[]>  
  [<CommonParameters>]
```

EXAMPLES

Example 1: Store alert text as a variable

```
C:\PS>$alertText = Read-WebAlertText -InputObject $Browser
```

REQUIRED PARAMETERS

-InputObject

Specifies IWebBrowser objects that represent the web browsers to act on. Enter a variable that contains the objects or type a command or expression that gets the objects.

Type:	IWebBrowser[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

OPTIONAL PARAMETERS

None.

INPUTS

Kelverion.WebAutomation.IWebBrowser

You can pipe web browser objects to the cmdlet.

OUTPUTS

System.String

The text that was read from the browser's alert window.

Read-WebElementAttribute

Reads the value of an attribute associated with a web page element.

```
Read-WebElementAttribute
  [-InputObject] <IWebElement[]>
  [-Name] <String>
  [<CommonParameters>]
```

REQUIRED PARAMETERS

-InputObject

Specifies IWebElement objects that represent the web page elements to act on. Enter a variable that contains the objects or type a command or expression that gets the objects

Type:	IWebElement[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

-Name

The name of the attribute to read.

Type:	String
Position:	1
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

OPTIONAL PARAMETERS

None

INPUTS

Kelverion.WebAutomation.IWebElement

You can pipe web page elements to the cmdlet

OUTPUTS

System.String

The value of the attribute that was read.

Read-WebElementCssProperty

Reads the value of an element's CSS property.

```
Read-WebElementCssProperty
  [-InputObject] <IWebElement[]>
  [-Name] <String>
  [<CommonParameters>]
```

REQUIRED PARAMETERS

-InputObject

Specifies IWebElement objects that represent the web page elements to act on. Enter a variable that contains the objects or type a command or expression that gets the objects

Type:	IWebElement[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

-Name

The name of the CSS property to read.

Type:	String
Position:	1
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

OPTIONAL PARAMETERS

None

INPUTS

Kelverion.WebAutomation.IWebElement

You can pipe web page elements to the cmdlet

OUTPUTS

System.String

The value of the CSS property that was read.

Remove-WebBrowser

Closes the web browser and releases any resources that were used.

```
Remove-WebBrowser  
  [-InputObject] <IWebBrowser[]>  
  [<CommonParameters>]
```

EXAMPLES

Example 1: Close a web browser

```
C:\PS>Remove-WebBrowser -InputObject $Browser
```

REQUIRED PARAMETERS

-InputObject

Specifies IWebBrowser objects that represent the web browsers to act on. Enter a variable that contains the objects, or type a command or expression that gets the objects.

Type:	IWebBrowser[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

OPTIONAL PARAMETERS

None

INPUTS

Kelverion.WebAutomation.IWebBrowser

You can pipe web browser objects to the cmdlet.

OUTPUTS

None

Remove-WebCookie

Removes the cookie with the specified name.

```
Remove-WebCookie
  [-InputObject] <IWebBrowser[]>
  [-Cookie] <Cookie[]>
  [-PassThru]
  [<CommonParameters>]
```

```
Remove-WebCookie
  [-InputObject] <IWebBrowser[]>
  [-Name] <String[]>
  [-PassThru]
  [<CommonParameters>]
```

REQUIRED PARAMETERS

-Cookie

Specifies System.Net.Cookie objects that represent the cookies to be removed from the web browser page.

Type:	System.Net.Cookie[]
Position:	1
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-InputObject

Specifies IWebBrowser objects that represent the web browsers to act on. Enter a variable that contains the objects or type a command or expression that gets the objects.

Type:	IWebBrowser[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

-Name

Species the names of the cookies to be removed.

Type:	String[]
Position:	1
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

OPTIONAL PARAMETERS

-Passthru

Returns an object that represents the web browser. By default, this cmdlet does not generate any output.

Type:	SwitchParameter
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

INPUTS

Kelverion.WebAutomation.IWebBrowser

You can pipe web browser objects to the cmdlet.

OUTPUTS

Kelverion.WebAutomation.IWebBrowser

This cmdlet generates a Kelverion.WebAutomation.IWebBrowser object that represents the web browser, if you use the PassThru parameter. Otherwise, this cmdlet does not generate any output.

Send-WebAlertCredential

Sends credentials to the active alert dialog.

```
Send-WebAlertCredential
  [-InputObject] <IWebBrowser[]>
  [-Credential] <PSCredential>
  [-PassThru]
  [<CommonParameters>]
```

EXAMPLES

Example 1: Get credentials for named user and submit to alert window.

```
C:\PS>Send-WebAlertCredential $Browser (Get-Credential Domain01\User64)
```

REQUIRED PARAMETERS

-Credential

Specifies the credentials to send to the alert window.

Type:	PSCredential
Position:	1
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-InputObject

Specifies IWebBrowser objects that represent the web browsers to act on. Enter a variable that contains the objects, or type a command or expression that gets the objects.

Type:	IWebBrowser[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

OPTIONAL PARAMETERS

-Passthru

Returns an object that represents the web browser. By default, this cmdlet does not generate any output.

Type:	SwitchParameter
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

INPUTS

Kelverion.WebAutomation.IWebBrowser

You can pipe web browser objects to the cmdlet.

OUTPUTS

Kelverion.WebAutomation.IWebBrowser

This cmdlet generates a `Kelverion.WebAutomation.IWebBrowser` object that represents the web browser, if you use the `PassThru` parameter. Otherwise, this cmdlet does not generate any output.

Send-WebAlertKeys

Send text to the active web browser alert window.

```
Send-WebAlertKeys
  [-InputObject] <IWebBrowser[] >
  [-Keys] <String>
  [-PassThru]
  [<CommonParameters>]
```

EXAMPLES

Example 1:

```
C:\PS>Set-WebTarget $Browser -Alert -PassThru | Send-WebAlertKeys -Keys
'112358'
```

REQUIRED PARAMETERS

-InputObject

Specifies IWebBrowser objects that represent the web browsers to act on. Enter a variable that contains the objects, or type a command or expression that gets the objects.

Type:	IWebBrowser[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

-Keys

The text to send to the alert window.

Type:	String
Position:	1
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

OPTIONAL PARAMETERS

-Passthru

Returns an object that represents the web browser. By default, this cmdlet does not generate any output.

Type:	SwitchParameter
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

INPUTS

Kelverion.WebAutomation.IWebBrowser

You can pipe web browser objects to the cmdlet.

OUTPUTS

Kelverion.WebAutomation.IWebBrowser

This cmdlet generates a `Kelverion.WebAutomation.IWebBrowser` object that represents the web browser, if you use the `PassThru` parameter. Otherwise, this cmdlet does not generate any output.

Send-WebElementKeys

Simulates typing text into the web page element.

```
Send-WebElementKeys
  [-InputObject] <IWebElement[]>
  [-Keys] <String>
  [-PassThru]
  [<CommonParameters>]
```

EXAMPLES

Example 1:

```
C:\PS>Get-WebElement $Browser -Name 'email' | Send-WebElementKeys
'joe@acme.com'
```

In this example the Get-WebElement cmdlet is used to find the 'email' text box element which is then piped to the Send-WebElementKeys cmdlet which types in the text 'joe@acme.com'

REQUIRED PARAMETERS

-InputObject

Specifies IWebElement objects that represent the web page elements to act on. Enter a variable that contains the objects or type a command or expression that gets the objects

Type:	IWebElement[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

-Keys

Type:	String
Position:	1
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

OPTIONAL PARAMETERS

-Passthru

Returns an object that represents the web browser. By default, this cmdlet does not generate any output.

Type:	SwitchParameter
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

INPUTS

Kelverion.WebAutomation.IWebElement

You can pipe web page elements to the cmdlet

OUTPUTS

Kelverion.WebAutomation.IWebElement

This cmdlet generates a `Kelverion.WebAutomation.IWebElement` object that represents web page elements, if you use the `PassThru` parameter. Otherwise, this cmdlet does not generate any output.

Set-WebElementSearchTimeout

Specifies the time to wait when searching for an element when the element is not immediately available.

```
Set-WebElementSearchTimeout
  [-InputObject] <IWebBrowser>
  [-TimeSpan] <TimeSpan>
  [-PassThru]
  [<CommonParameters>]
```

```
Set-WebElementSearchTimeout
  [-InputObject] <IWebBrowser>
  -Seconds <Int32>
  [-PassThru]
  [<CommonParameters>]
```

EXAMPLES

Example 1: Set the web element search timeout to 20 seconds.

```
C:\PS>Set-WebElementSearchTimeout $Browser -Seconds 20
```

Example 2: Set the web element search timeout using a timespan

```
C:\PS>Set-WebElementSearchTimeout $Browser
[TimeSpan]::FromSeconds($timeToWait)
```

REQUIRED PARAMETERS

-InputObject

Specifies IWebBrowser objects that represent the web browsers to act on. Enter a variable that contains the objects or type a command or expression that gets the objects.

Type:	IWebBrowser[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

-Seconds

Specifies the number of seconds to wait.

Type:	Int32
Alias	s
Position:	1
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-TimeSpan

Specifies the timespan to wait.

Type:	TimeSpan
Position:	1
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

OPTIONAL PARAMETERS

-Passthru

Returns an object that represents the web browser. By default, this cmdlet does not generate any output.

Type:	SwitchParameter
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

INPUTS

Kelverion.WebAutomation.IWebBrowser

You can pipe web browser objects to the cmdlet.

OUTPUTS

Kelverion.WebAutomation.IWebBrowser

This cmdlet generates a `Kelverion.WebAutomation.IWebBrowser` object that represents the web browser, if you use the `PassThru` parameter. Otherwise, this cmdlet does not generate any output.

Set-WebElementSelection

Selects an option of a SELECT element by index, text or value.

```
Set-WebElementSelection
  [-InputObject] <IWebElement[]>
  -Index Int32
  -PassThru
  [<CommonParameters>]
```

```
Set-WebElementSelection
  [-InputObject] <IWebElement[]>
  -Text String
  -PassThru
  [<CommonParameters>]
```

```
Set-WebElementSelection
  [-InputObject] <IWebElement[]>
  -Value <String>
  -PassThru
  [<CommonParameters>]
```

EXAMPLES

Example 1: Select web element option by text

```
C:\PS>Get-WebElement $Browser -Id 'gender' | Set-WebElementSelection -Text 'Female'
```

Example 2: Select web element option by value

```
C:\PS>Get-WebElement $Browser -Id 'title' | Set-WebElementSelection -Value 1
```

REQUIRED PARAMETERS

-Index

Specifies the index of the option to select, as determined by the elements index attribute.

Type:	String
Position:	Named
Default value:	None

Accept pipeline input:	False
Accept wildcard characters:	False

-InputObject

Specifies IWebElement objects that represent the web page elements to act on. Enter a variable that contains the objects or type a command or expression that gets the objects

Type:	IWebElement[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

-Text

Specifies the text of the element to select.

Type:	String
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-Value

Specifies the value of the element to select.

Type:	String
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

OPTIONAL PARAMETERS

-Passthru

Returns an object that represents the web browser. By default, this cmdlet does not generate any output.

Type:	SwitchParameter
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

INPUTS

Kelverion.WebAutomation.IWebElement

You can pipe web page elements to the cmdlet

OUTPUTS

Kelverion.WebAutomation.IWebElement

This cmdlet generates a `Kelverion.WebAutomation.IWebElement` object that represents web page elements, if you use the `PassThru` parameter. Otherwise, this cmdlet does not generate any output.

Set-WebPageLoadTimeout

Specified time to wait for web pages to load.

```
Set-WebPageLoadTimeout
  [-InputObject] <IWebBrowser<>>
  [-TimeSpan] <TimeSpan>
  [-PassThru]
  [<CommonParameters>]
```

```
Set-WebPageLoadTimeout
  [-InputObject] <IWebBrowser<>>
  -Seconds <Int32>
  [-PassThru]
  [<CommonParameters>]
```

EXAMPLES

Example 1: Set the web page load timeout to 20 seconds.

```
C:\PS>Set-WebPageLoadTimeout $Browser -Seconds 20
```

Example 2: Set the web page load timeout using a timespan

```
C:\PS>Set-WebPageLoadTimeout $Browser [TimeSpan]::FromSeconds($timeToWait)
```

REQUIRED PARAMETERS

-InputObject

Specifies IWebBrowser objects that represent the web browsers to act on. Enter a variable that contains the objects, or type a command or expression that gets the objects.

Type:	IWebBrowser[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

-Seconds

Specifies the number of seconds to wait.

Type:	Int32
Alias	s
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-TimeSpan

Specifies the timespan to wait.

Type:	TimeSpan
Position:	1
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

OPTIONAL PARAMETERS

-Passthru

Returns an object that represents the web browser. By default, this cmdlet does not generate any output.

Type:	SwitchParameter
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

INPUTS

Kelverion.WebAutomation.IWebBrowser

You can pipe web browser objects to the cmdlet.

OUTPUTS

Kelverion.WebAutomation.IWebBrowser

This cmdlet generates a `Kelverion.WebAutomation.IWebBrowser` object that represents the web browser, if you use the `PassThru` parameter. Otherwise, this cmdlet does not generate any output.

Set-WebTarget

Sets the active web page search context.

```
Set-WebTarget
  [-InputObject] <IWebBrowser[]>
  -ActiveElement
  [-PassThru]
  [<CommonParameters>]
```

```
Set-WebTarget
  [-InputObject] <IWebBrowser[]>
  -Alert
  [-PassThru]
  [<CommonParameters>]
```

```
Set-WebTarget
  [-InputObject] <IWebBrowser[]>
  -Frame <Object>
  [-PassThru]
  [<CommonParameters>]
```

```
Set-WebTarget
  [-InputObject] <IWebBrowser[]>
  -ParentFrame
  [-PassThru]
  [<CommonParameters>]
```

```
Set-WebTarget
  [-InputObject] <IWebBrowser[]>
  -Window <String>
  [-PassThru]
  [<CommonParameters>]
```

```
Set-WebTarget
  [-InputObject] <IWebBrowser[]>
  -DefaultContent
  [-PassThru]
  [<CommonParameters>]
```

EXAMPLES

Example 1: Select the frame with the name iFrameHelper

```
C:\PS>Set-Target $Browser -Frame iFrameHelper
```

Example 2: Select the outermost frame

```
C:\PS>Set-Target -DefaultContent
```

In this example, the target is being set to the default web content. This is often necessary after setting the target to an internal frame so that you can search again from the outermost context.

REQUIRED PARAMETERS

-ActiveElement

Indicates that the cmdlet should select the element that currently has the focus or the body element if no element with focus can be detected.

Type:	SwitchParameter
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-Alert

Indicates that the cmdlet should select the currently active modal alert window.

Type:	SwitchParameter
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-DefaultContent

Indicates that the cmdlet should select the first frame on the page of the main document when a page contains no iFrames.

Type:	SwitchParameter
Position:	Named
Default value:	None

Accept pipeline input:	False
Accept wildcard characters:	False

-Frame

Indicates that the cmdlet should select the frame with the specified name or zero-based index within the currently selected frame.

Type:	Object
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-InputObject

Specifies IWebBrowser objects that represent the web browsers to act on. Enter a variable that contains the objects, or type a command or expression that gets the objects.

Type:	IWebBrowser[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

-ParentFrame

Indicates that the cmdlet should select the parent frame of the currently selected frame.

Type:	SwitchParameter
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-Window

Indicates that the cmdlet should select the window with the specified name.

Type:	String
Position:	Named

Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

OPTIONAL PARAMETERS

-Passthru

Returns an object that represents the web browser. By default, this cmdlet does not generate any output.

Type:	SwitchParameter
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

INPUTS

Kelverion.WebAutomation.IWebBrowser

You can pipe web browser objects to the cmdlet.

OUTPUTS

Kelverion.WebAutomation.IWebBrowser

This cmdlet generates a Kelverion.WebAutomation.IWebBrowser object that represents the web browser, if you use the PassThru parameter. Otherwise, this cmdlet does not generate any output.

Set-WebFrame

Selects the target frame by web element, name or zero-based index.

```
Set-WebFrame
  [-InputObject] <IWebBrowser<>>
  -Element <IWebElement>
  [-PassThru]
  [<CommonParameters>]
```

```
Set-WebFrame
  [-InputObject] <IWebBrowser<>>
  -Index <Int32>
  [-PassThru]
  [<CommonParameters>]
```

```
Set-WebFrame
  [-InputObject] <IWebBrowser<>>
  -Name <String>
  [-PassThru]
  [<CommonParameters>]
```

REQUIRED PARAMETERS

-Element

Specifies an IWebElement object that represents the frame to select.

Type:	IWebElement
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-InputObject

Specifies IWebBrowser objects that represent the web browsers to act on. Enter a variable that contains the objects, or type a command or expression that gets the objects.

Type:	IWebBrowser[]
Position:	0
Default value:	None

Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

-Index

Specifies the zero-based index of the frame to select.

Type:	Int32
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-Name

Specifies the name of the frame to select.

Type:	String
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

OPTIONAL PARAMETERS

-Passthru

Returns an object that represents the web browser. By default, this cmdlet does not generate any output.

Type:	SwitchParameter
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

INPUTS

Kelverion.WebAutomation.IWebBrowser

You can pipe web browser objects to the cmdlet.

OUTPUTS

Kelverion.WebAutomation.IWebBrowser

This cmdlet generates a `Kelverion.WebAutomation.IWebBrowser` object that represents the web browser, if you use the `PassThru` parameter. Otherwise, this cmdlet does not generate any output.

Step-WebPageBack

Move back a single entry in the browser's history.

```
Step-WebPageBack
  [-InputObject] <IWebBrowser[]>
  [-PassThru]
  [<CommonParameters>]
```

REQUIRED PARAMETERS

-InputObject

Specifies IWebBrowser objects that represent the web browsers to act on. Enter a variable that contains the objects or type a command or expression that gets the objects.

Type:	IWebBrowser[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

OPTIONAL PARAMETERS

-Passthru

Returns an object that represents the web browser. By default, this cmdlet does not generate any output.

Type:	SwitchParameter
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

INPUTS

Kelverion.WebAutomation.IWebBrowser

You can pipe web browser objects to the cmdlet.

OUTPUTS

Kelverion.WebAutomation.IWebBrowser

This cmdlet generates a `Kelverion.WebAutomation.IWebBrowser` object that represents the web browser, if you use the `PassThru` parameter. Otherwise, this cmdlet does not generate any output.

Step-WebPageForward

Move forward a single entry in the browser's history.

```
Step-WebPageForward
  [-InputObject] <IWebBrowser[]>
  [-PassThru]
  [<CommonParameters>]
```

REQUIRED PARAMETERS

-InputObject

Specifies IWebBrowser objects that represent the web browsers to act on. Enter a variable that contains the objects or type a command or expression that gets the objects.

Type:	IWebBrowser[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

OPTIONAL PARAMETERS

-Passthru

Returns an object that represents the web browser. By default, this cmdlet does not generate any output.

Type:	SwitchParameter
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

INPUTS

Kelverion.WebAutomation.IWebBrowser

You can pipe web browser objects to the cmdlet.

OUTPUTS

Kelverion.WebAutomation.IWebBrowser

This cmdlet generates a `Kelverion.WebAutomation.IWebBrowser` object that represents the web browser, if you use the `PassThru` parameter. Otherwise, this cmdlet does not generate any output.

Submit-WebElement

Submits the web element to the web server.

```
Submit-WebElement  
  [-InputObject] <IWebElement[]>  
  [-PassThru]  
  [<CommonParameters>]
```

EXAMPLES

Example 1: Find and submit the 'submit' button

```
C:\PS>Submit-WebElement (Get-WebElement $Browser -Name 'submit')
```

REQUIRED PARAMETERS

-InputObject

Specifies IWebElement objects that represent the web page elements to act on. Enter a variable that contains the objects or type a command or expression that gets the objects

Type:	IWebElement[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

OPTIONAL PARAMETERS

-Passthru

Returns an object that represents the web browser. By default, this cmdlet does not generate any output.

Type:	SwitchParameter
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

INPUTS

Kelverion.WebAutomation.IWebElement

You can pipe web page elements to the cmdlet

OUTPUTS

Kelverion.WebAutomation.IWebElement

This cmdlet generates a `Kelverion.WebAutomation.IWebElement` object that represents web page elements, if you use the `PassThru` parameter. Otherwise, this cmdlet does not generate any output.

Update-WebPage

Refreshes the current web page.

```
Update-WebPage
  [-InputObject] <IWebBrowser[]>
  [-PassThru]
  [<CommonParameters>]
>
```

REQUIRED PARAMETERS

-InputObject

Specifies IWebBrowser objects that represent the web browsers to act on. Enter a variable that contains the objects, or type a command or expression that gets the objects.

Type:	IWebBrowser[]
Position:	0
Default value:	None
Accept pipeline input:	True (ByValue)
Accept wildcard characters:	False

OPTIONAL PARAMETERS

-Passthru

Returns an object that represents the web browser. By default, this cmdlet does not generate any output.

Type:	SwitchParameter
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

INPUTS

Kelverion.WebAutomation.IWebBrowser

You can pipe web browser objects to the cmdlet.

OUTPUTS

Kelverion.WebAutomation.IWebBrowser

This cmdlet generates a `Kelverion.WebAutomation.IWebBrowser` object that represents the web browser, if you use the `PassThru` parameter. Otherwise, this cmdlet does not generate any output.