How To Monitor Websites In Nagios XI 2024

Purpose

This document describes how to monitor websites effectively with Nagios XI. There are three different wizards to help you monitor the health of websites and to be notified when unexpected changes occur on the website or processes are not working as expected.

- Website Wizard is used to monitor the steady-state aspects of a website
- Website URL wizard is similar and allows you to monitor the status and content of a specific URL
- Web Transaction Wizard monitors transactions and other interactive activities on your website

Considerations

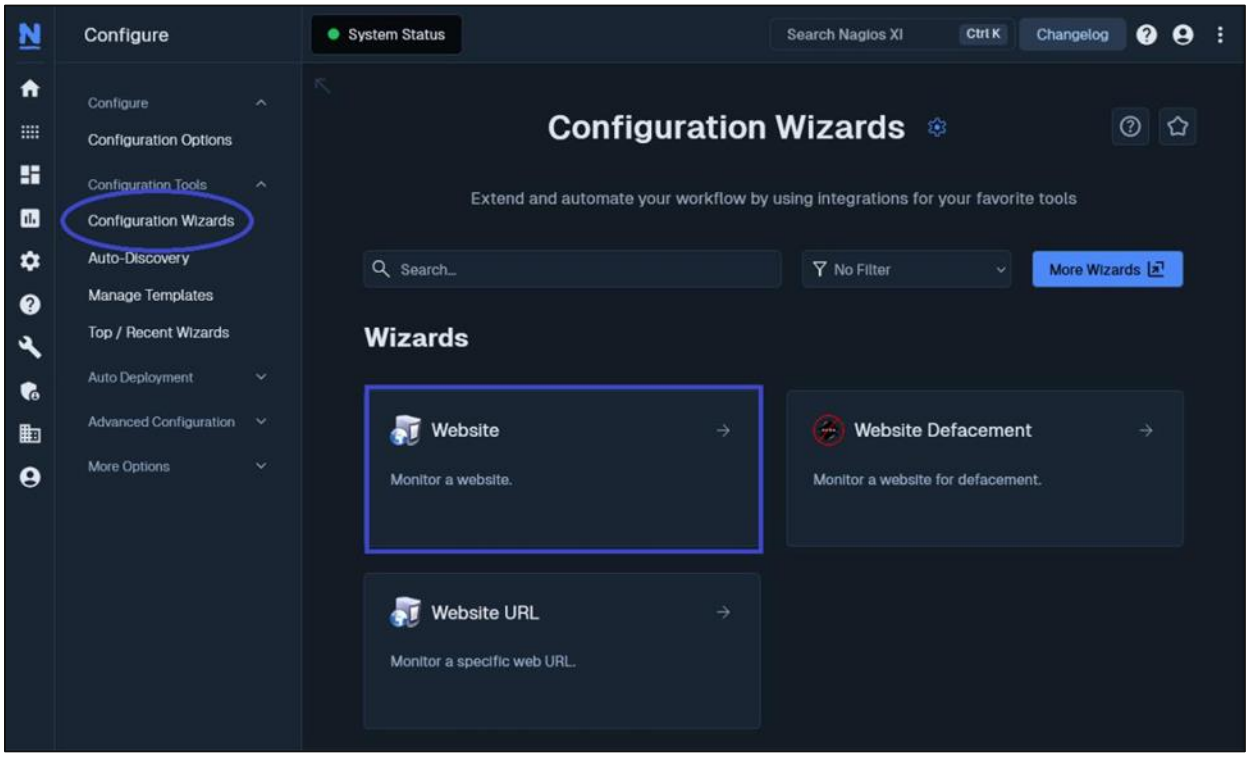
When monitoring websites, it is often recommended to check the operational status of several key metrics, including:

- HTTP response validity and load time
- DNS resolution and IP address match
- Website content
- SSL certificates
- Web transaction success and run time

Your monitoring needs will vary depending on the complexity of your website, its purpose, and its intended end-user.

Running A Wizard

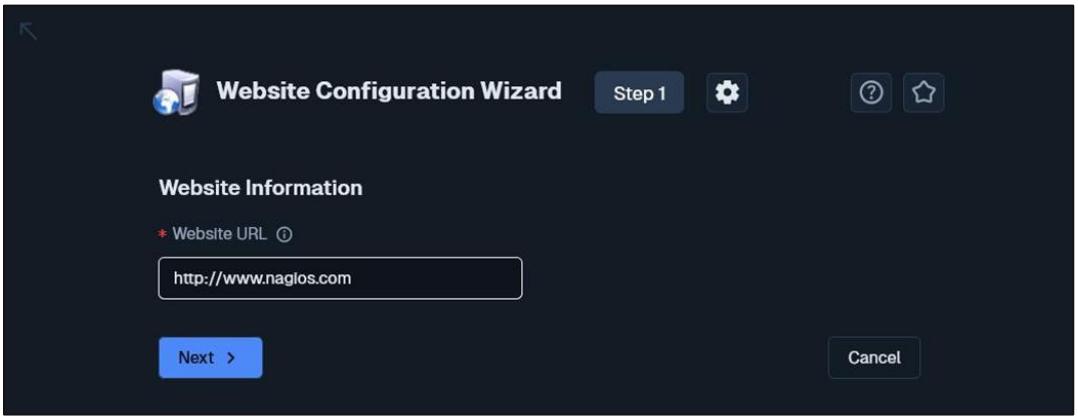
To begin using one of the wizards, navigate via the top menu bar to **Configure > Run a configuration wizard** and select the required wizard from the list. In the following screenshot you can see how the search field allows you to quickly find a wizard.



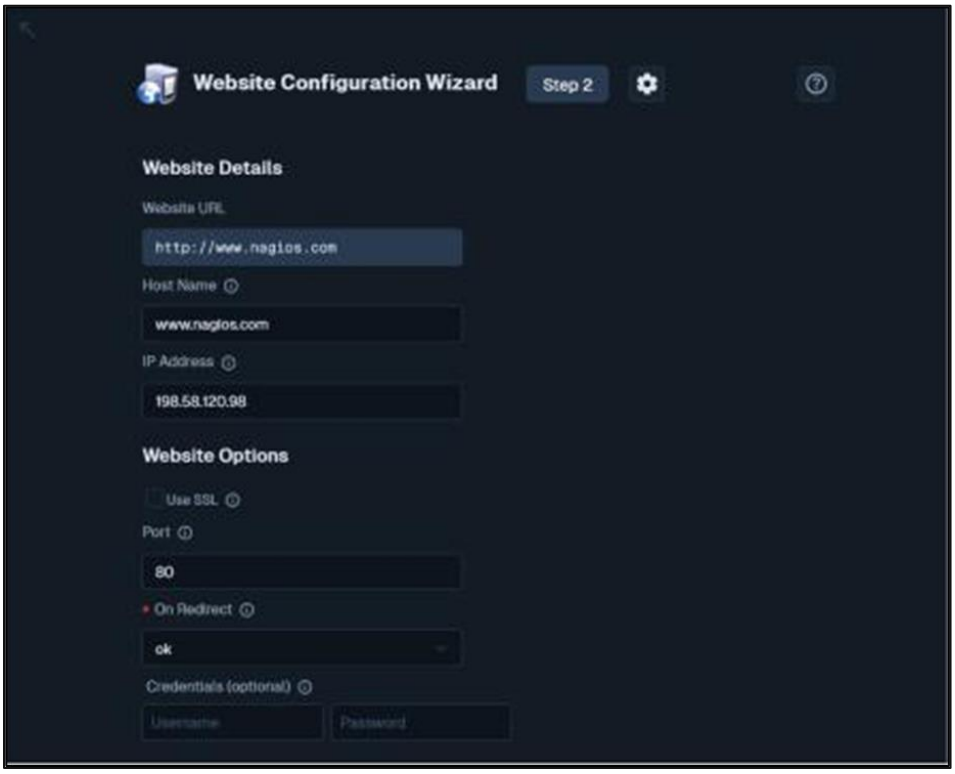
The Website Wizard

This is the wizard you will use for most types of sites, where you are checking common server/site metrics. The best way to understand its capabilities is to see them, so a walkthrough of using this wizard follows. Select the **Website** wizard from the list of wizards:

1. **Step 1** enter the URL to your website. This can be either the front page of the domain or any sub-page. The latter will only have a purpose of checking of existence of that page and content monitoring on it.
    a. Click **Next** once you've entered a URL

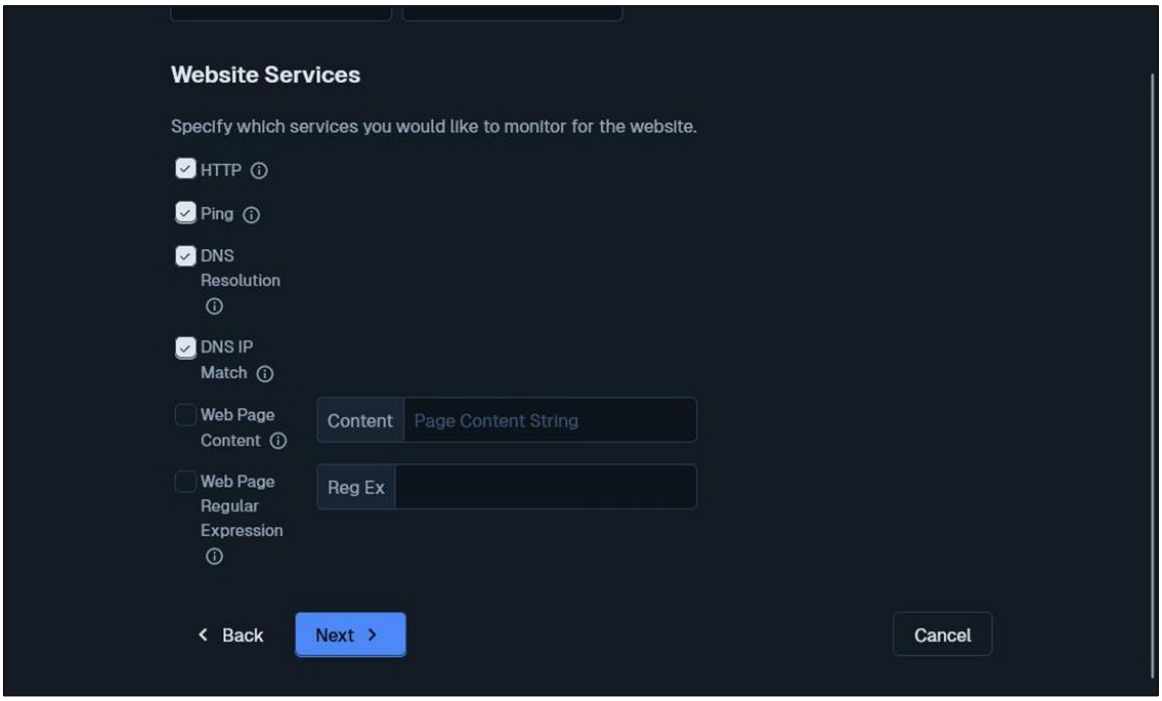


2. **Step 2** is where you define the monitoring options. Here you can define which services you want to add for this site, including whether:
   - To use SSL (HTTPS) and what port to use
   - What to do if a redirect response is received, in this example follow was chosen
   - Credentials if required



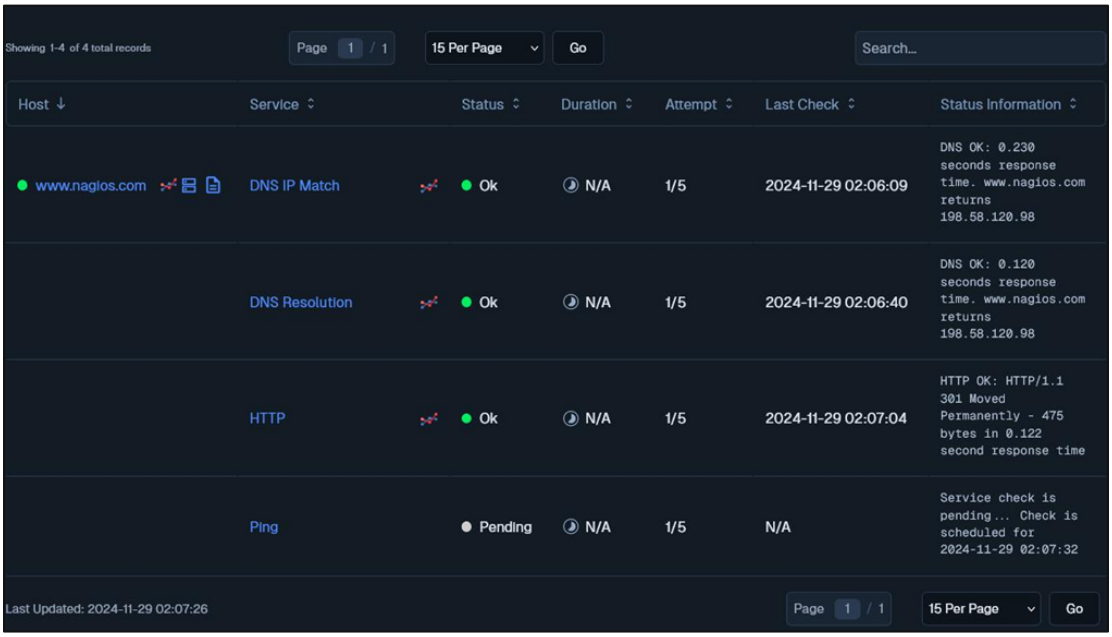
- Test responds with a valid HTTP response
- The option of a ping check
- DNS resolution check
- The DNS response matches the IP address that was resolved when you ran the wizard
- A particular string is found on the page (either a literal string 192.168.172.18 or as a regular expression such as:
  `(^192\.168)|(^10\.)|(^172\.1[6-9])|(^172\.2[0-9])| (^172\.3[0-1]))`
- The SSL certificate's expiry date is sufficiently far away

a. The Use SSL option and SSL Certificate check will only be available if the URL you gave in **Step 2** begin with https.



b. Once you've finished making your selections click **Next** and then complete the wizard by choosing the required options in **Step 3 – Step 5**.

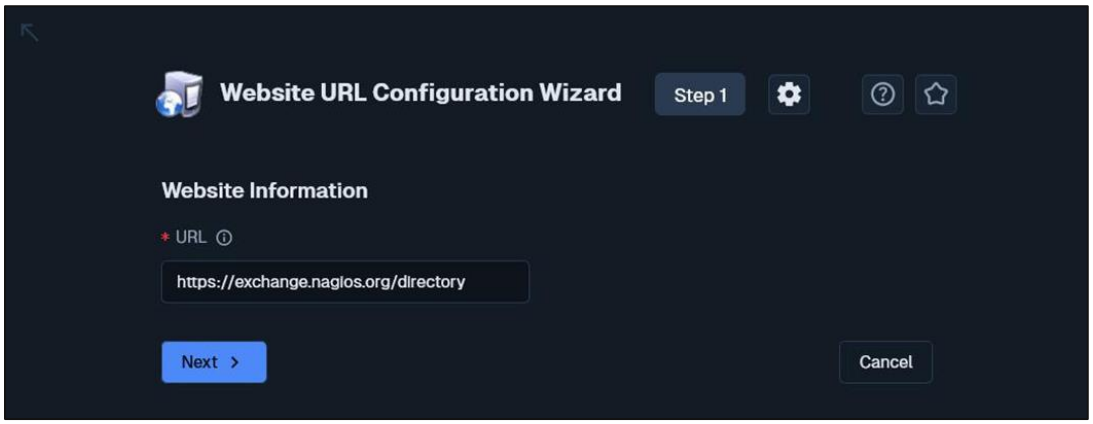
3. To finish, click on **Finish** in the last step of the wizard. This will create new hosts and services and begin monitoring. Once the wizard applies the configuration, click the **View status** details for your website link to see the new host and services that were created.



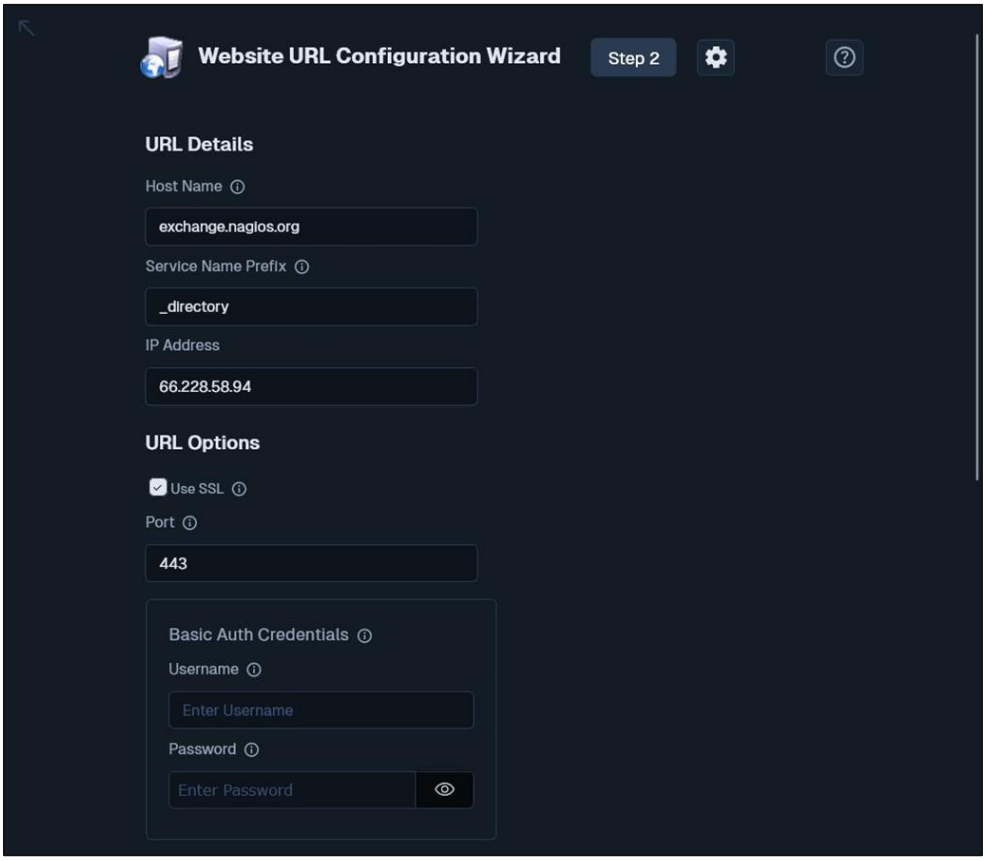
The Website URL Wizard

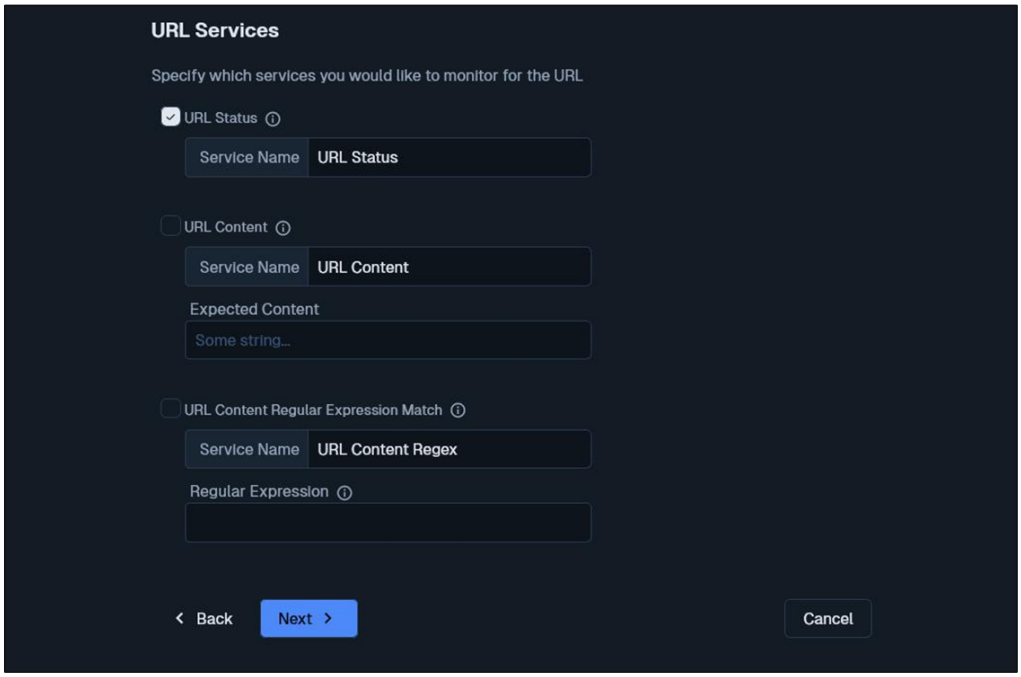
The Website URL wizard is like the Website wizard. Select the Website URL wizard from the list of wizards.

1. On **Step 1** enter the URL to your website.
    a. Click **Next** once you've entered the URL.



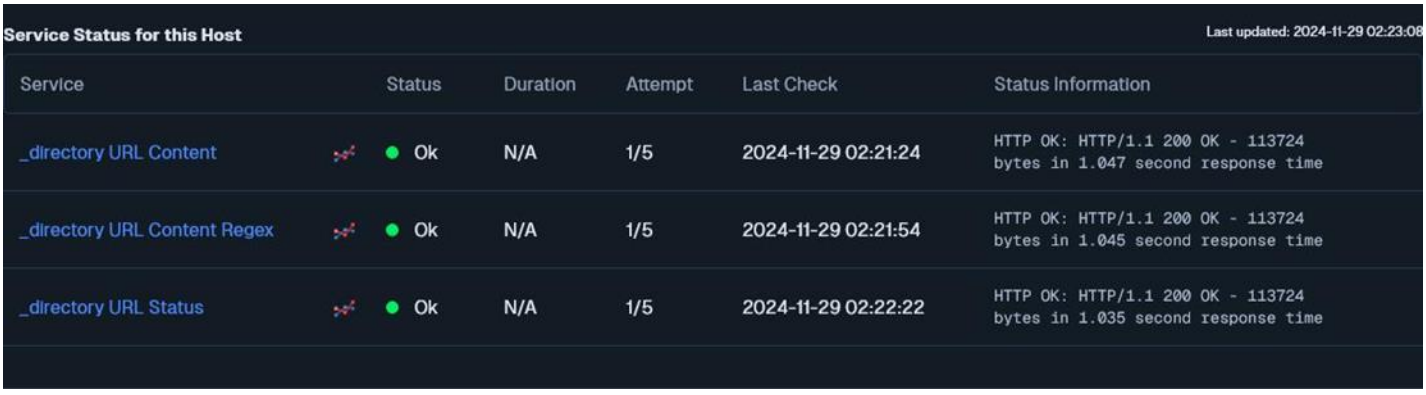
2. **Step 2** is where you define the monitoring options. Here you can define which services you want to add for this site:
   - Service Name Prefix
   - To use SSL (HTTPS) and what port to use
   - Credentials if required
   - A URL Status service
   - A particular string is found in the content of the page
   - A particular string is found in the content of the page using a regular expression





a. Once you've finished making your selections click Next and then complete the wizard by choosing the required options in **Step 3 – Step 5**.

3. To finish, click on **Finish** in the last step of the wizard. This will create new hosts and services and begin monitoring. Once the wizard applies the configuration, click the View status details for your website link to see the new host and services that were created.



The Web Transaction Wizard

A more complex use case of website monitoring would be if you expect the content to be dynamic with user input and actions and want to test that those actions complete as expected. For instance, you might test that a search box works (and what results are returned), whether the purchase and checkout process of your web store is behaving properly, or that a user can log in successfully.

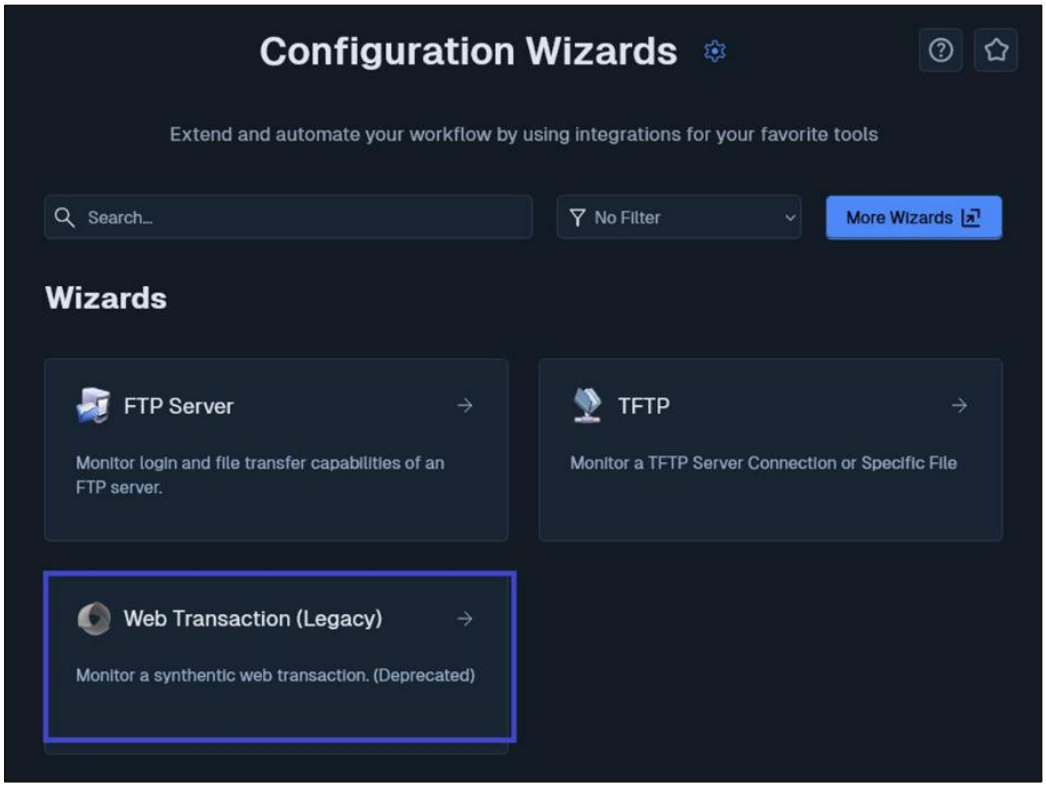
The **Web Transaction wizard** can be used for these types of checks. Additionally, it allows for checking all three of those in succession, and other multi-step procedures where each stage may be dependent on the previous one.

This wizard relies on a tool called WebInject, which handles the transition logic between stages of the transaction. Therefore, you will need to understand how to write the configuration XML in the WebInject syntax to configure these kinds of checks. Some examples are given below, see the WebInject manual for more information.

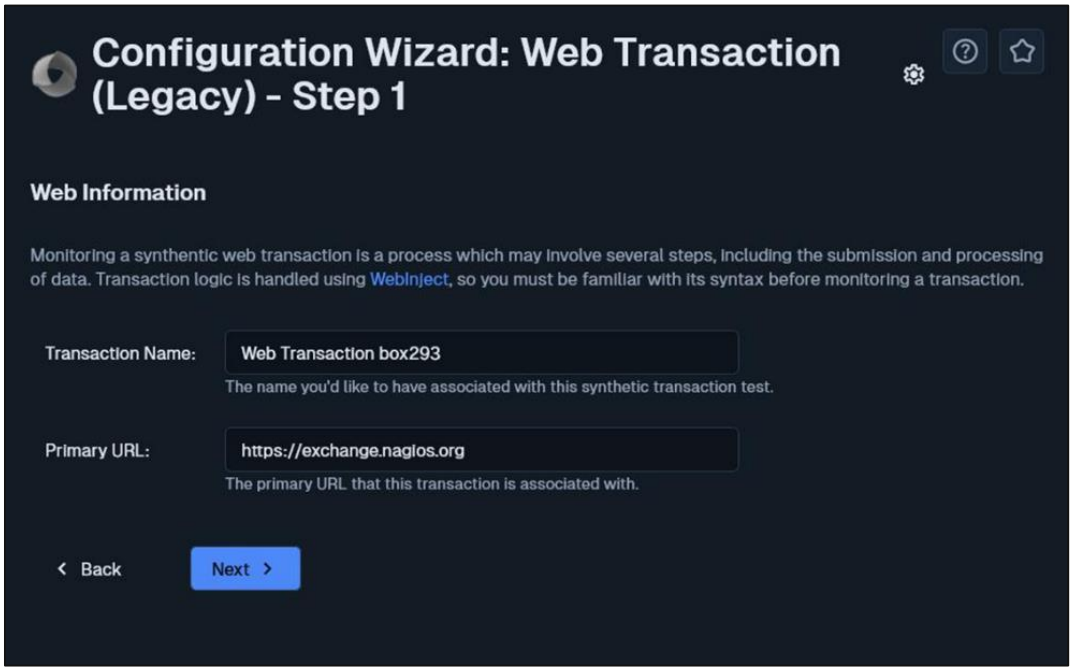
Note that certain special characters need to be escaped. For instance, the `<` should be replaced with `\x3C` so as not to interfere with the XML. Within POST data URL escapes are used, so for instance `@` becomes `%40`.

Providing an example of such a process can get complicated quickly. Here is a simple example where we will search the Nagios Exchange for "box293". I know that the search results will come back with the word "box293" in the result, so I am going to look for this. If for some reason "box293" was not present on the web page the service would go into a critical state.

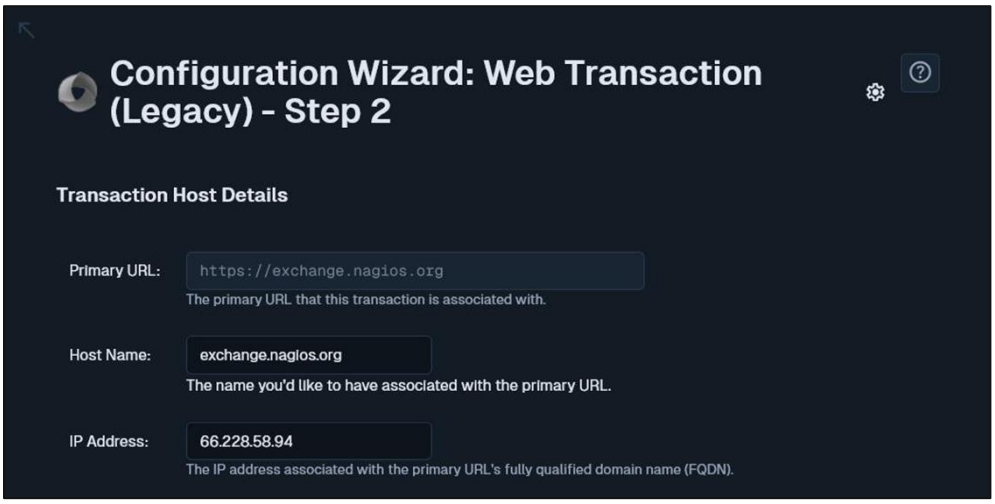
Select the Website Transaction wizard from the list of wizards:



1. On **Step 1** enter the Transaction Name you want to assign to this WebInject test. You will also need to provide the Primary URL for this website. Click **Next**.

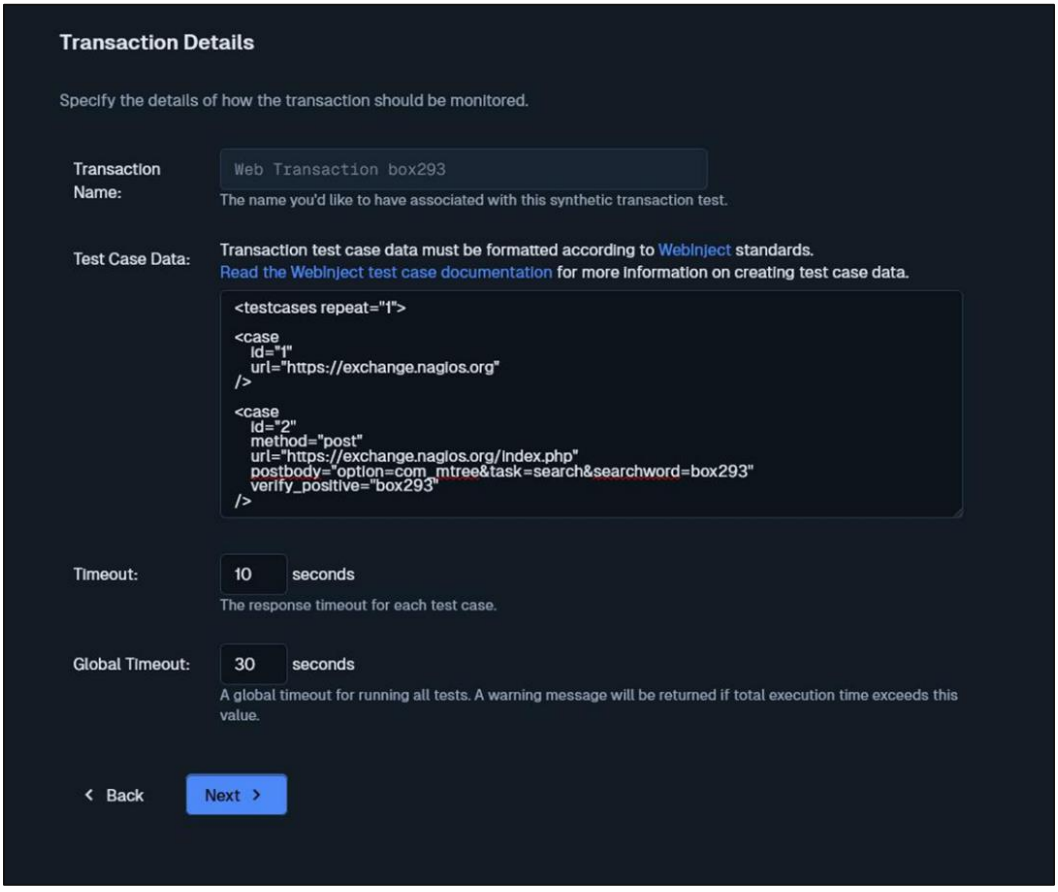


2. On **Step 2** you will need to make sure the **Host Name** and **IP Address** fields are correctly populated.



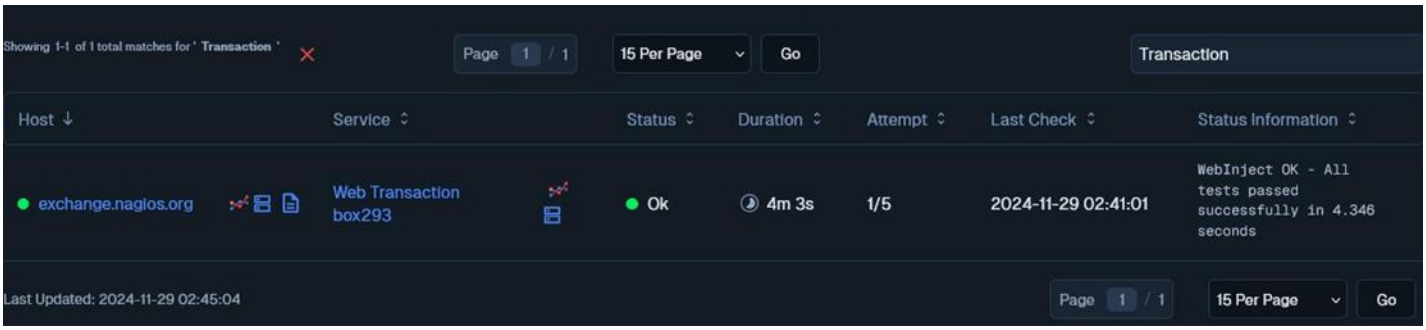
   a. You will then need to provide the **Test Case Data** that will be used for this transaction. The text field will already be populated with the first case of id 1.

b. In the screenshot you can see an additional case of id 2 has been added.
c. The magic being performed here is:
    i. `url =` This is the URL that is being used for this test case.
    ii. `postbody =` These are the fields / commands that are posted to the url. This varies depending on your actual website. In this example it is searching for the word box293.
    iii. `verifypositive =` This is what needs to exist in the data returned, in this example the word box293 must exist for this case to be completed successfully.
d. You can also specify the timeout periods that apply.



3. Once you've finished making your selections click **Next** and then complete the wizard by choosing the required options in **Step 3 – Step 5**. To finish, click on Finish in the last step of the wizard. This will create new hosts and services and begin monitoring.

4. Once the wizard applies the configuration, click the **View status** details for the Nagios Exchange link to see the new host and services that were created.



Here is the test case data shown on the screenshot on step 2 of the wizard.

```
<testcases repeat="1">
<case
id="1"
url="https://exchange.nagios.org/"
/>
<case
id="2"
method="post"
url="https://exchange.nagios.org/index.php"
postbody="option=com_mtree&task=search&searchword=box293"
verifypositive="box293"
/>
</testcases>
```

Logging into Nagios XI Test Case Example

Below is an example of an attempt to log into XI.

```
<testcases repeat="1">
```

```
<case
id="1"
url="http://192.168.55.21/nagiosxi/login.php"
parseresponse='var nsp_str = "|";'
/>
<case
id="2"
method="post"
url=http://192.168.55.21/nagiosxi/login.php
posttype="application/x-www-form-urlencoded"
postbody="nsp={PARSEDRESULT}&page-
=auth&debugg=&pageopt=login&username=nagiosadmin&password=welcome&loginButto
n="
verifynegative="var nsp_str"
logrequest="no"
logresponse="no"
/>
</testcases>
```

Troubleshooting WebInject Transactions

If for some reason the transaction did not work correctly, you will need to go and adjust the test case data in the configuration file that was created by the wizard. You may also want to run the transaction at the command line to see exactly what is happening. All transaction configuration files are in the following directory on the Nagios XI server:

```
/usr/local/nagiosxi/etc/components/webinject/
```

In the wizard that was just run, two files were created:

```
exchange_nagios_org__Nagios_Exchange_Search___box293_config.xml
exchange_nagios_org__Nagios_Exchange_Search___box293_testdata.xml
```

The file that ends in _testdata.xml is what contains the test case data, it's exactly as you pasted it into the configuration wizard. The file that ends in _config.xml is what tells WebInject how to run the test data.

To demonstrate that the test case data is working, you will change the postbody = line to search for the word "box294", which should return a critical result as there is nothing on the Nagios Exchange contaminating the word "box294".

Once you have made the change, run the test from the command line using the following commands:

```
cd /usr/local/nagiosxi/etc/components/webinject/
./webinject.pl -c exchange_nagios_org__Nagios_Exchange_Search___
box293_config.xml
```

You should receive the following output:

```
WebInject CRITICAL - Test case number 2 failed |time=10.126;30;;0
```

If you wanted more verbose output, you need to edit the _config.xml file and change the reporttype to standard:

```
<reporttype>standard</reporttype>
```

Now when you run the command above the output will be something like:

```
Starting WebInject Engine...
-------------------------------------------------------
Test: exchange_nagios_org__Nagios_Exchange_Search___box293_ testdata.xml - 1
Passed HTTP Response Code Verification (not in error range)
TEST CASE PASSED
Response Time = 3.482 sec
-------------------------------------------------------
Test: exchange_nagios_org__Nagios_Exchange_Search___box293_
testdata.xml - 2
Verify : "box293"
Failed Positive Verification
Passed HTTP Response Code Verification (not in error range)
TEST CASE FAILED
Response Time = 3.337 sec
-------------------------------------------------------
Start Time: Thu Dec 8 17:09:14 2016
Total Run Time: 6.862 seconds
Test Cases Run: 2
```

```
Test Cases Passed: 1
Test Cases Failed: 1
Verifications Passed: 2
Verifications Failed: 1
```

When you've finished troubleshooting don't forget to change the reporttype back to nagios:

```
<reporttype>nagios</reporttype>
```

Example: Using An Online Store

This example provides a more comprehensive demonstration of how WebInject could be used online store and put an item in a shopping cart via the O'Reilly Media's web site. It's worth noting that this example was created some time in the past and may not actually work now because the website may have been updated by now, however the concept is still the same and worth retaining in the documentation.

- Step 1 confirms that the login page loads
- Step 2 provides your authentication credentials and then checks that they were accepted, and follows the redirect to the members page in step 3
- Step 4 checks the price on “Nagios, Second Edition (by Wolfgang Barth)”
- Step 5 adds it to your shopping cart
- Step 6 confirming it remains in your cart properly after that and appears to be in stock
- Finally, the last two steps log you out and check that the home page loads

By carefully crafting the different steps and plenty of sufficiently specific verifypositive and verifynegative parameters, a great deal of information can be confirmed through this single Nagios service.

```
<testcases repeat="1">
<testvar varname="USER">1rc94d+86yw3m9jrqj18@sharklasers.com</testvar>
<testvar varname="PASS">holden123</testvar> <case
id="1"
description1="Login page"
url="https://members.oreilly.com/account/login"
parseresponse='_authentication_token" type="hidden" value="|"|escape'
<testcases repeat="1">
<testvar varname="USER">1rc94d+86yw3m9jrqj18@sharklasers.com</testvar>
<testvar varname="PASS">holden123</testvar> <case
id="1"
```

```
description1="Login page"
url="https://members.oreilly.com/account/login"
parseresponse='_authentication_token" type="hidden" value="|"|escape'
verifypositive="Sign in"
/>
<case
id="2"
description1="Sign in"
url="https://members.oreilly.com/account/login"
method="post"
postbody="email=${USER}&password=${PASS}&_authentication_token=
{PARSEDRESULT}"
verifypositive="https://members.oreilly.com/account/benefits"
parseresponse="found at |;"
/>
<case
id="3"
description1="Members page"
url="{PARSEDRESULT}"
verifypositive="view or edit your account information"
/>
<case
id="4"
description1="Book price"
url="http://oreilly.com/catalog/9781593271794/"
verifypositive="59.95"
/>
<case
id="5"
description1="Book added to cart"
url="https://epoch.oreilly.com/shop/cart.orm?prod=9781593271794.BOOK"
verifypositive="Nagios, 2Ed"
/>
<case
id="6"
description1="Book still in cart"
url="https://epoch.oreilly.com/shop/cart.orm"
verifypositive="Nagios, 2Ed"
```

```
verifynegative="Backorder"
/>
<case
id="7"
description1="Logout"
url=https://members.oreilly.com/account/logout
verifypositive="http://oreilly.com/"
parseresponse="found at |;"
/>
<case
id="8"
description1="Main page"
url="{PARSEDRESULT}"
verifypositive="News &amp; Commentary"
/>
</testcases>
```

For more information, visit the Configuration Wizards documentation.

Finishing Up

This completes the documentation on how to monitor websites effectively with Nagios XI. If you have additional questions or other support-related questions, please visit us at our Nagios Support Forum, Nagios Knowledge Base, or Nagios Library:

Visit Nagios Support Forum Visit Nagios Knowledge Base Visit Nagios Library