ZABBIX 7.0

Browser Monitoring

Sergejs Olonkins
Quality assurance engineer

Synthetic web monitoring

Synthetic Monitoring involves simulating user interactions with a website using automated scripts to test functionality and monitor performance.

Key features:

- Proactive Testing: Performed regularly, even if there are no real users on the site.
- Scenario-Based Testing: Can test specific scenarios, such as login processes, form submissions, or navigation flows.
- Baselines and Benchmarks: Helps in establishing performance baselines and benchmarks for comparison over time.

Advanced scenarios

Monitoring scenarios are created in JavaScript (Duktape engine)

Because browser item emulates a real browser, it is possible to:

- Log on and log out from the website
- Fill and submit different forms
- Navigate through multiple pages
- Collect values, attributes and properties of web page elements
- Create complex if - else scenarios

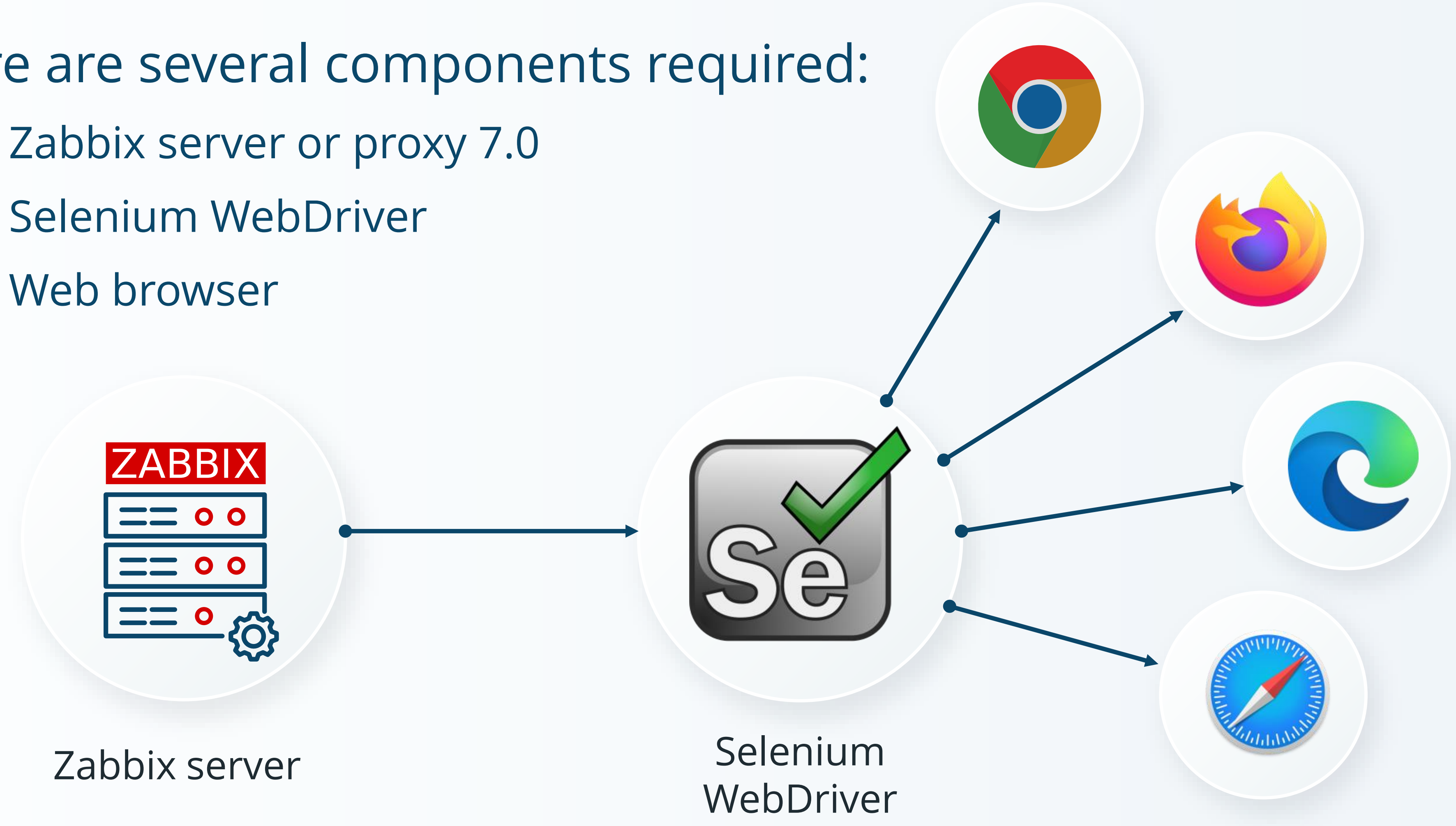
ZABBIX 7.0

Monitoring Environment Setup

Monitoring requirements

There are several components required:

- Zabbix server or proxy 7.0
- Selenium WebDriver
- Web browser



Zabbix server configuration

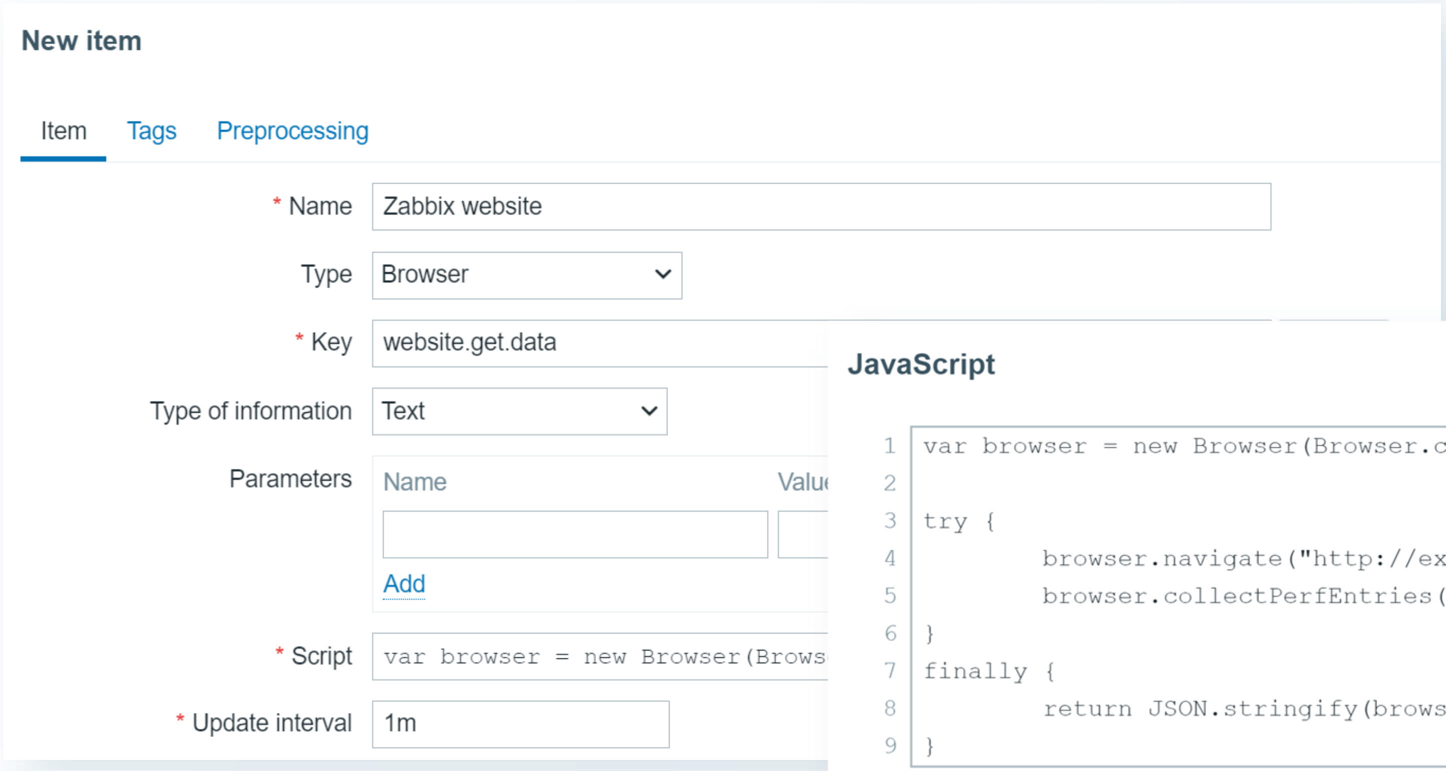
```
####### Browser monitoring #######

### Option: WebDriverURL
#       WebDriver interface HTTP[S] URL. For example http://localhost:4444 used with
#       Selenium WebDriver standalone server.
#
# Mandatory: no
# Default:
# WebDriverURL=
WebDriverURL=http://192.168.0.1:4444

### Option: StartBrowserPollers
#       Number of pre-forked instances of browser item pollers.
#
# Mandatory: no
# Range: 0-1000
# Default:
# StartBrowserPollers=1
StartBrowserPollers=3
```

ZABBIX 7.0

The Browser item

Browser item

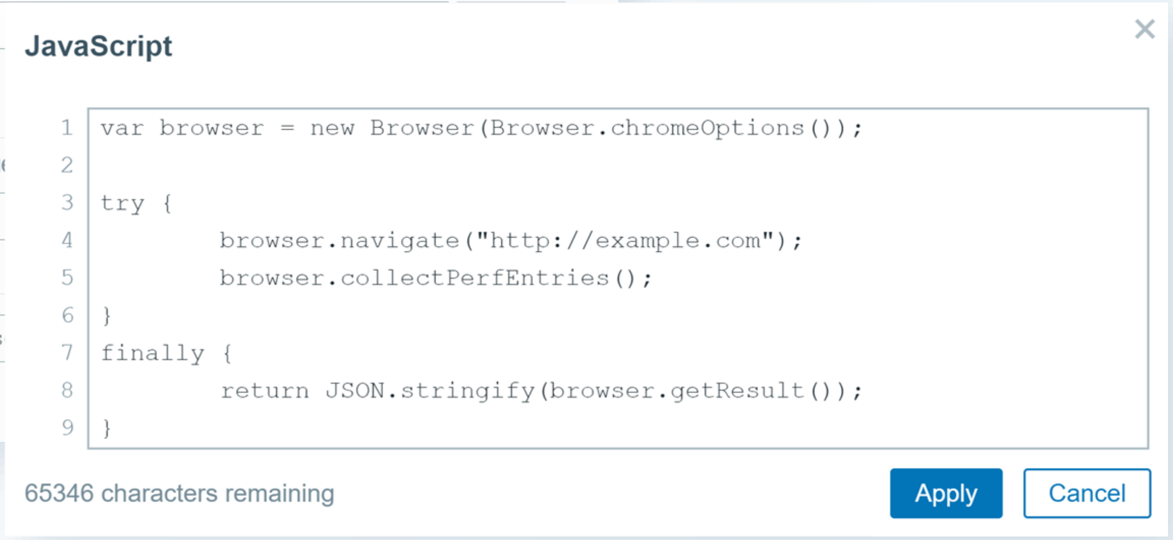
Zabbix 7.0 introduces new item type: Browser



New item

Item | Tags | Preprocessing

* Name	Zabbix website
Type	Browser
* Key	website.get.data
Type of information	Text
Parameters	Name / Value — Add
* Script	var browser = new Browser(Brows
* Update interval	1m



JavaScript

```
var browser = new Browser(Browser.chromeOptions());

try {
        browser.navigate("http://example.com");
        browser.collectPerfEntries();
}
finally {
        return JSON.stringify(browser.getResult());
}
```

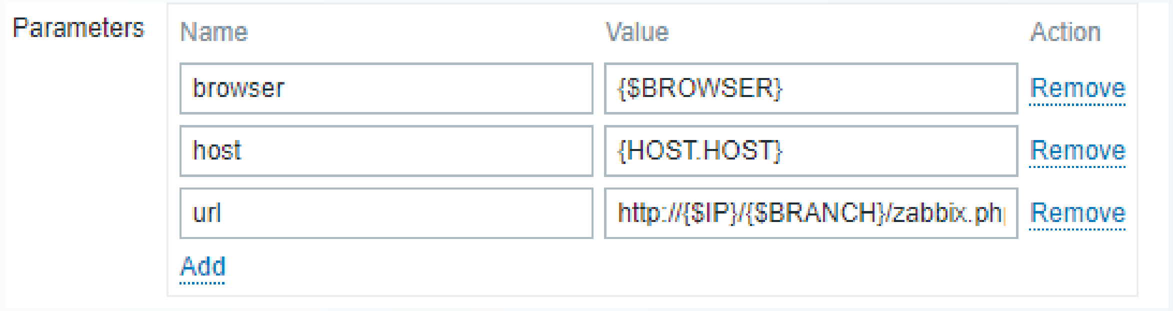
65346 characters remaining

Apply | Cancel

Browser item parameters

It is possible to send custom parameters to the JavaScript:

- Write name and value pairs in the Parameters
- Macros can be used as the browser item parameters

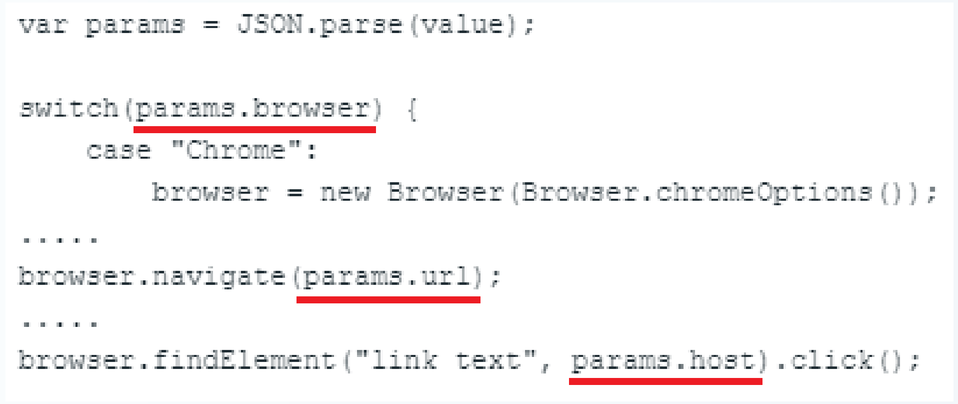


Parameters

Name	Value	Action
browser	{$BROWSER}	Remove
host	{HOST.HOST}	Remove
url	http://{$IP}/{$BRANCH}/zabbix.ph	Remove

Add

Usage in script:

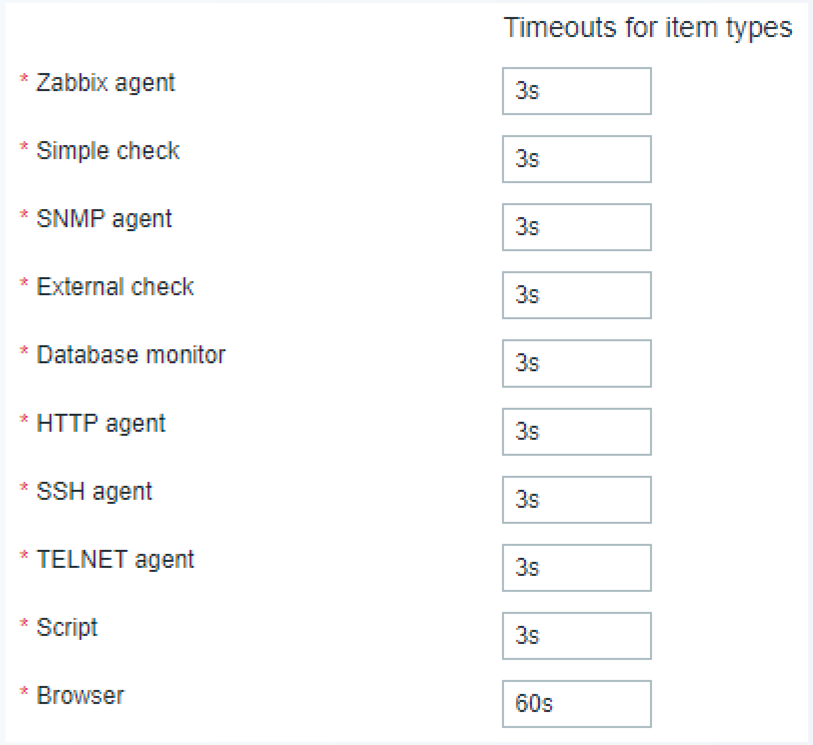


```
var params = JSON.parse(value);

switch(params.browser) {
    case "Chrome":
        browser = new Browser(Browser.chromeOptions());
.....
browser.navigate(params.url);
.....
browser.findElement("link text", params.host).click();
```

Browser item timeout

Data collection timeout can be specified for the browser item:

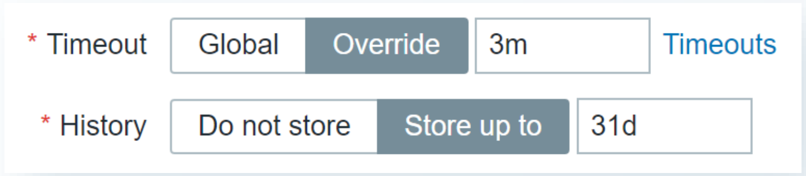
- On the Zabbix server level
- On the Zabbix proxy level
- On the individual item level



Timeouts for item types

* Zabbix agent	3s
* Simple check	3s
* SNMP agent	3s
* External check	3s
* Database monitor	3s
* HTTP agent	3s
* SSH agent	3s
* TELNET agent	3s
* Script	3s
* Browser	60s

Item override



* Timeout	Global [✓ Override]	3m Timeouts
* History	Do not store [✓ Store up to]	31d

Browser item output

The browser item collects all performance metrics in the JSON format.

Line ‘*browser.collectPerfEntries("open page");*’ will result in:

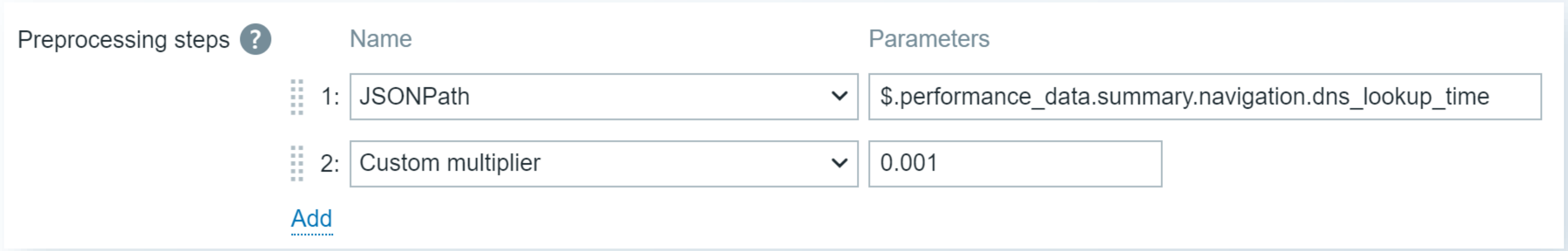
```
{
 "duration": 2.1230485439300537,
 "performance_data": {
  "details": [
   {
    "mark": "open page",
    "navigation": {
     "entry_type": "navigation",
     "dom_content_loaded_event_start": 0.09439999999910593,
     "domain_lookup_start": 0.0012999999970197678,
     "tls_negotiation_time": 0.0020999999940395355,
     "request_start": 0.0020999999940395355,
     "redirect_start": 0,
     "load_event_start": 0.096,
     "name": "http://192.168.6.197/zabbix-7.0/zabbix.php?action=host.list",
     "connect_end": 0.0012999999970197678,
     .........
```

Individual metrics

Possible to return a single value or to collect multiple metrics.

Data are extracted from the browser item using dependent items:

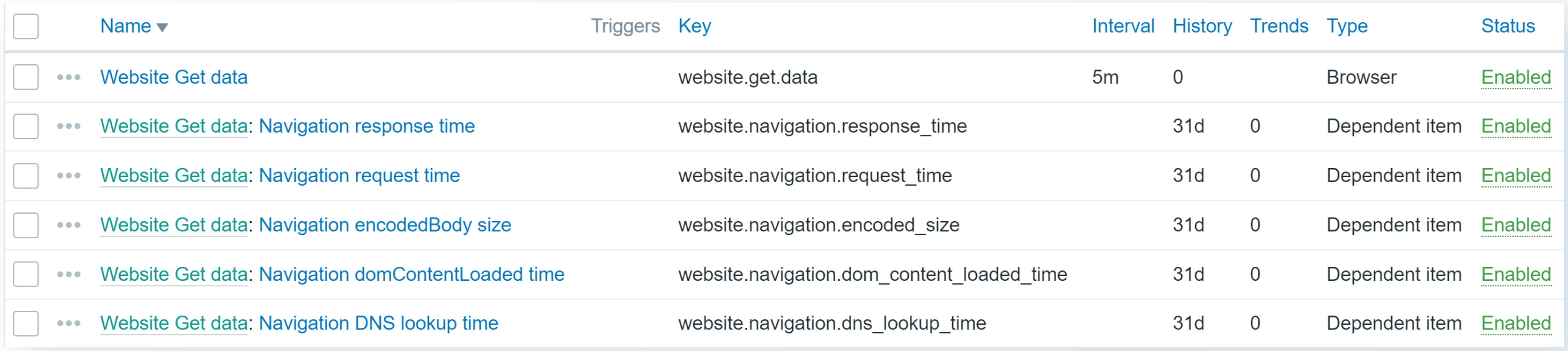
- Browser item collects data in JSON format
- Dependent items use the JSONPath preprocessing step to extract data



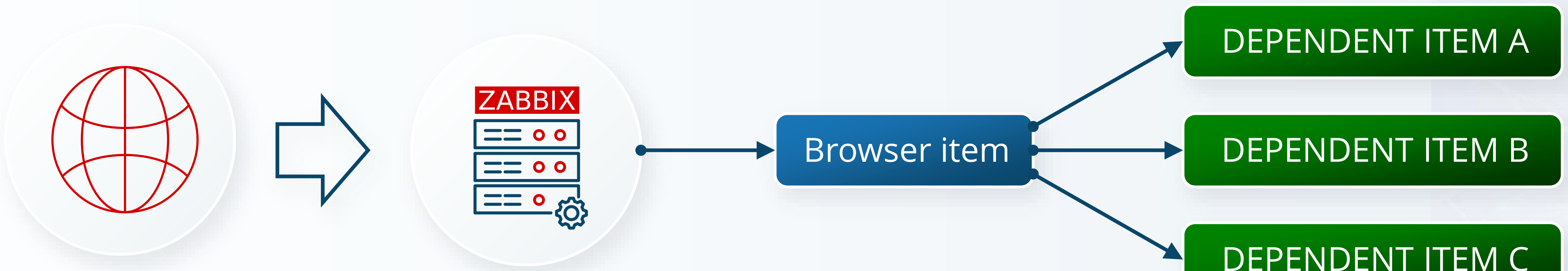
Preprocessing steps

	Name	Parameters
1:	JSONPath	$.performance_data.summary.navigation.dns_lookup_time
2:	Custom multiplier	0.001

Browser item and dependent items



Name ▾	Triggers	Key	Interval	History	Trends	Type	Status
Website Get data		website.get.data	5m	0		Browser	Enabled
Website Get data: Navigation response time		website.navigation.response_time		31d	0	Dependent item	Enabled
Website Get data: Navigation request time		website.navigation.request_time		31d	0	Dependent item	Enabled
Website Get data: Navigation encodedBody size		website.navigation.encoded_size		31d	0	Dependent item	Enabled
Website Get data: Navigation domContentLoaded time		website.navigation.dom_content_loaded_time		31d	0	Dependent item	Enabled
Website Get data: Navigation DNS lookup time		website.navigation.dns_lookup_time		31d	0	Dependent item	Enabled



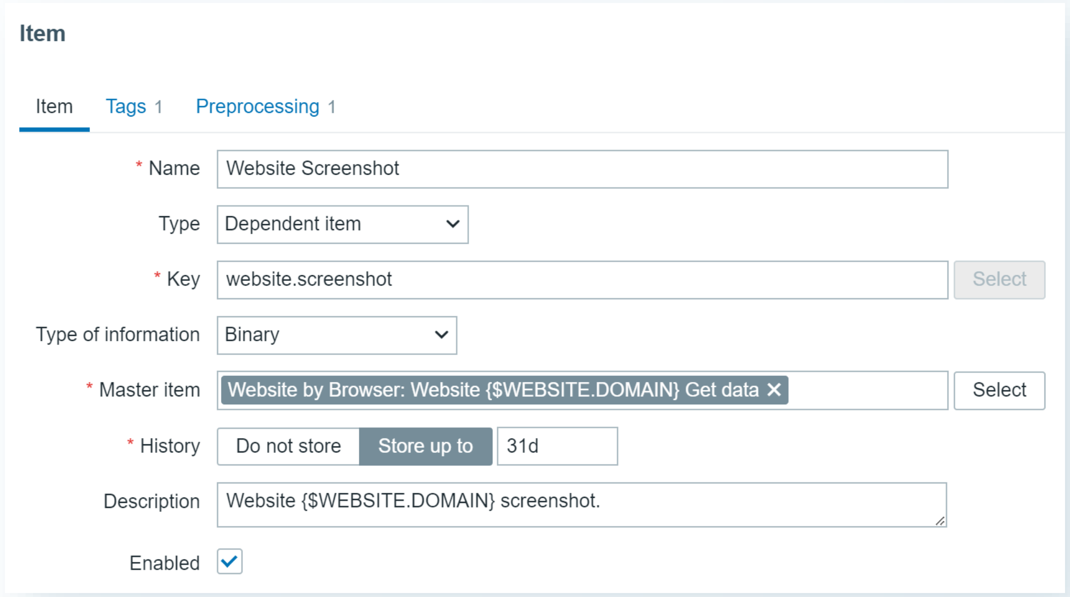
ZABBIX 7.0

Screenshots

Taking screenshots

The browser item can take screenshot from the monitored pages:

- To take a screenshot of the page – use the *getScreenshot()* method
- Screenshot is included in the JSON object in base64 format
- It is extracted into a binary dependent item



Item

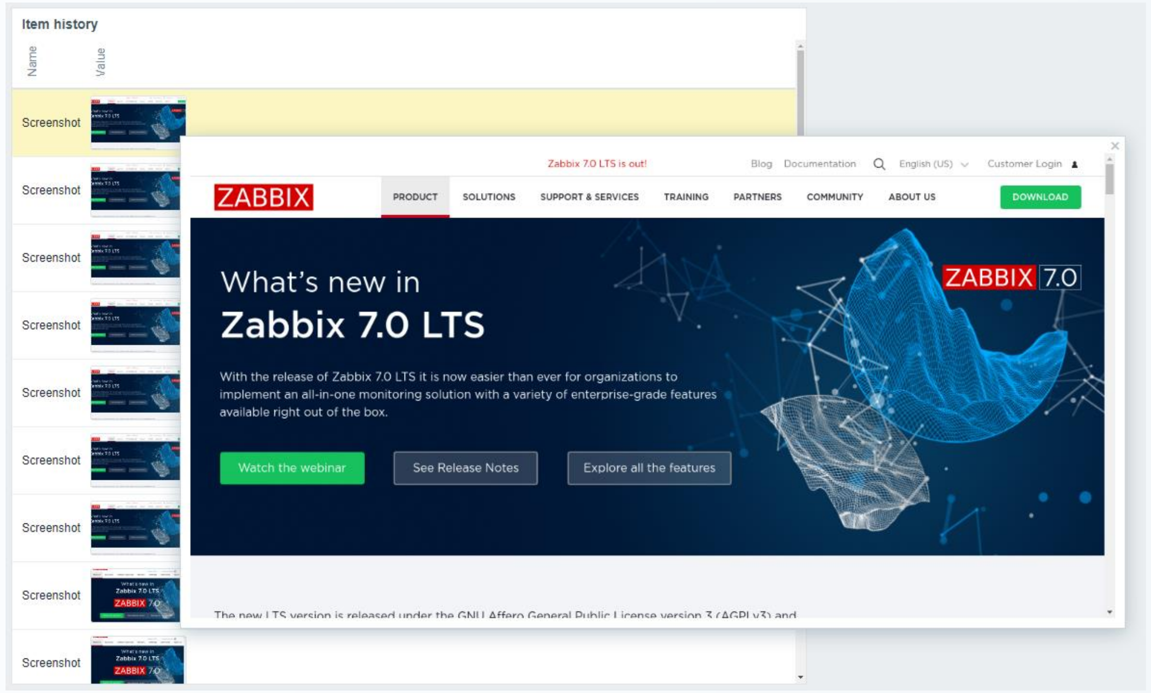
* Name	Website Screenshot
Type	Dependent item
* Key	website.screenshot
Type of information	Binary
* Master item	Website by Browser: Website {$WEBSITE.DOMAIN} Get data
* History	Do not store [✓ Store up to] 31d
Description	Website {$WEBSITE.DOMAIN} screenshot.
Enabled	✓

Important:

Only dependent items have type of information Binary.

Item history widget example

- Default screenshot size is 1920 x 1080 px.
- Supported screenshot size: up to 8000 x 8000 px.
- Screenshot can be displayed using the "Item history" widget.
- Screenshot size is specified by the *setScreenSize(x,y)* method.


Item history

Name	Value
Screenshot	
Screenshot	
Screenshot	
Screenshot	
Screenshot	
Screenshot	
Screenshot	
Screenshot	
Screenshot	

What's new in
Zabbix 7.0 LTS

With the release of Zabbix 7.0 LTS it is now easier than ever for organizations to implement an all-in-one monitoring solution with a variety of enterprise-grade features available right out of the box.

The new LTS version is released under the GNU Affero General Public License version 3 (AGPLv3) and

ZABBIX 7.0

Objects and methods

Objects in browser item script

1. Browser object – manages the session throughout the whole item execution (initialize session => execute all actions in script => close session).
2. Element - an element that is found on the opened web page.
3. Cookie - a cookie returned by the *getCookies()* method.
4. Alert – a web page alert returned by the *getAlert()* method.
5. Result – represents the gathered statistics data gathered by *getResult()* method. Contains the following:
   1. Session statistics (for example total session duration)
   2. Error information (if such occurred)
   3. Performance data (if such collected)
   4. Any collected data that is written directly to the result object

Browser object methods

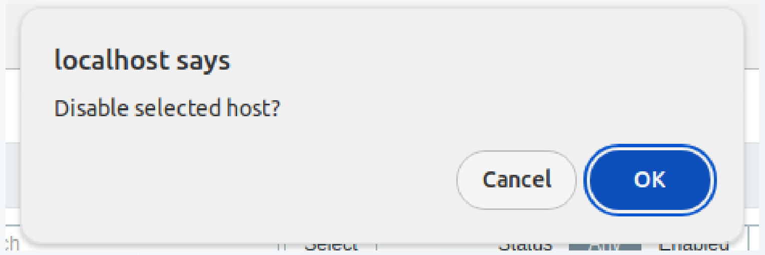
Area	Methods
Retrieving predefined browser options	• chromeOptions() • firefoxOptions() • edgeOptions() • safariOptions()
Timeout management	• setScriptTimeout(timeout) • setSessionTimeout(timeout) • setElementWaitTimeout(timeout)
Collecting data	• collectPerfEntries(mark) • getRawPerfEntries() • getResult() • getScreenshot()
Error related operations	• getError() • setError(message) • discardError()
URL and page related operations	• navigate(url) • getUrl() • getPageSource() • getAlert() • getCookies() • addCookie(cookie)
Locating elements on page	• findElement(strategy, selector) • findElements(strategy, selector)

Element object methods

Method	Description
getAttribute(name)	Return the value of the given attribute of the object, for example: class, id, style, display e.t.c.
getProperty(name)	Return the value of the given property of the object, for example: className, baseURI e.t.c.
getText()	Return element text.
click()	Click on element.
sendKeys(keys)	Send keys to element. Method is used for filling data into editable elements.
clear()	Remove the content of editable element

Alert and Cookie objects

- Alert object is returned by the *getAlert()* method.
- Alert object has the following methods:
  - accept()
  - dismiss()



localhost says

Disable selected host?

- Cookie object is returned by the *getCookies()* method.
- Cookie object doesn't have any methods

Script:

```
browser.navigate(params.url);
browser.addCookie(JSON.parse(params.cookie));
result = browser.getCookies();
```

Output:

```
[{"domain":"192.168.6.197","expiry":1753169400,"httpOnly":true,"name":"🔥 Zabbix 7.0","path":"/zbx-24473-
6.5","sameSite":"Lax","secure":false,"value":"Best release ever"},
{"domain":"192.168.6.197","httpOnly":true,"name":"zbx_session","path":"/zbx-24473-
6.5","sameSite":"Lax","secure":false,"value":"eyJzZXNzaW9uaWQiOiI1Y2ZmZTg0ODM1MTRhNTk2Y2I1OWY2M2NiMTk0Nj
c1ZCIsInNpZ24iOiJjM2E0YjgyZTJlZmIzODQwZWZmMzU1OWMyMzM3OGQ0NzJkOWJkMDYwNmFhMzQzYzdlZTZkZWEy
ZDgyNGI1NzIzIn0%3D"}]
```

ZABBIX 7.0

How to …

Start browser in headless or normal mode

It is possible to make selenium to actually launch the browser normally.

Table shows examples of parameters to be overridden to make the item launch the browser in headless mode (default) and in normal mode:

Browser	Headless mode	Normal mode

Chrome	browser = new Browser(Browser.chromeOptions());	var opts = Browser.chromeOptions();
		opts.capabilities.alwaysMatch['goog:chromeOptions'].args = [];
		var browser = new Browser(opts);

Firefox	browser = new Browser(Browser.firefoxOptions());	var opts = Browser.firefoxOptions();
		opts.capabilities.alwaysMatch['moz:firefoxOptions'].binary = '/usr/bin/firefox';
		opts.capabilities.alwaysMatch['moz:firefoxOptions'].args = [];
		var browser = new Browser(opts);

Edge	browser = new Browser(Browser.edgeOptions());	var opts = Browser.edgeOptions();
		opts.capabilities.alwaysMatch['ms:edgeOptions'].binary = '/usr/bin/microsoft-edge';
		opts.capabilities.alwaysMatch['ms:edgeOptions'].args = [];
		var browser = new Browser(opts);

Safari	-----	browser = new Browser(Browser.safariOptions());

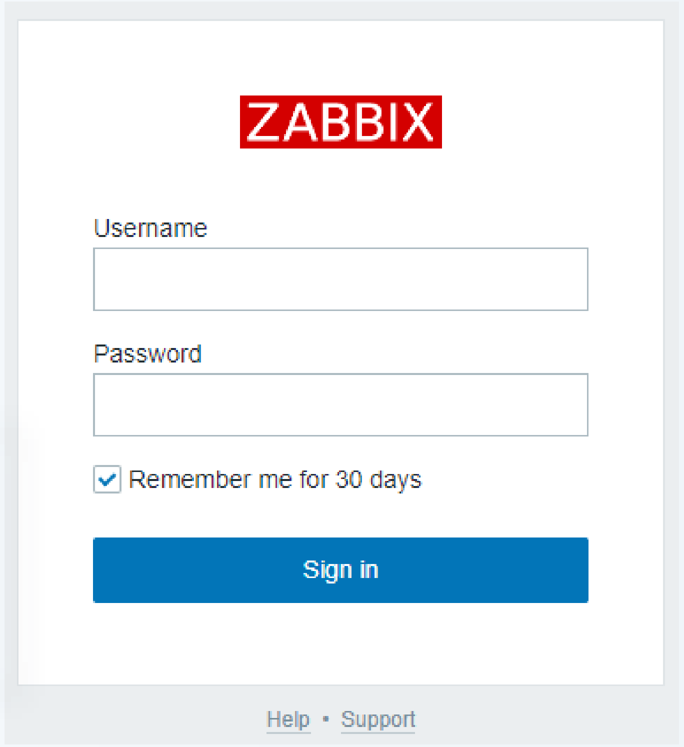
Send values and click on elements (Login)

The below scenario performs the following actions:

- Locate “Username” field and send value “Admin”
- Locate “Password” field and send value “zabbix”
- Locate button “Sign in” and click on it

Script:

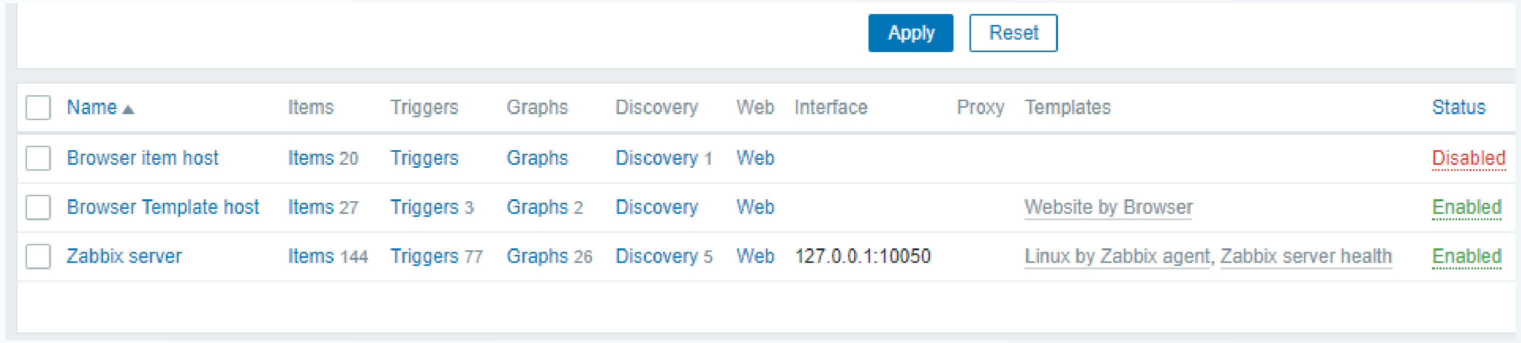
```
var params = JSON.parse(value);
browser = new Browser(Browser.chromeOptions());
browser.navigate(params.url);
browser.findElement("xpath", "//input[@id='name']").sendKeys("Admin");
browser.findElement("xpath", "//input[@id='password']").sendKeys("zabbix");
browser.findElement("xpath", "//button[@id='enter']").click();
```



Locate page elements

The following strategies can be used to locate page elements:

- CSS selector
- Link text
- Partial link text
- Tag name
- Xpath



	Name ▲	Items	Triggers	Graphs	Discovery	Web	Interface	Proxy	Templates	Status
☐	Browser item host	Items 20	Triggers	Graphs	Discovery 1	Web				Disabled
☐	Browser Template host	Items 27	Triggers 3	Graphs 2	Discovery	Web			Website by Browser	Enabled
☐	Zabbix server	Items 144	Triggers 77	Graphs 26	Discovery 5	Web	127.0.0.1:10050		Linux by Zabbix agent, Zabbix server health	Enabled

Script:

```
result = browser.getResult();

result.submit_filter_text = browser.findElement("xpath", "//button[@name='filter_set']").getText();
result.zabbix_server_id = browser.findElement("link text", "Zabbix server").getAttribute("data-hostid");
result.disabled_button_text = browser.findElement("css selector", ".link-action.red").getText();
result.browser_host_count = browser.findElements("partial link text", "Browser ").length;
result.header_id = browser.findElement("tag name", "h1").getAttribute("id");
```

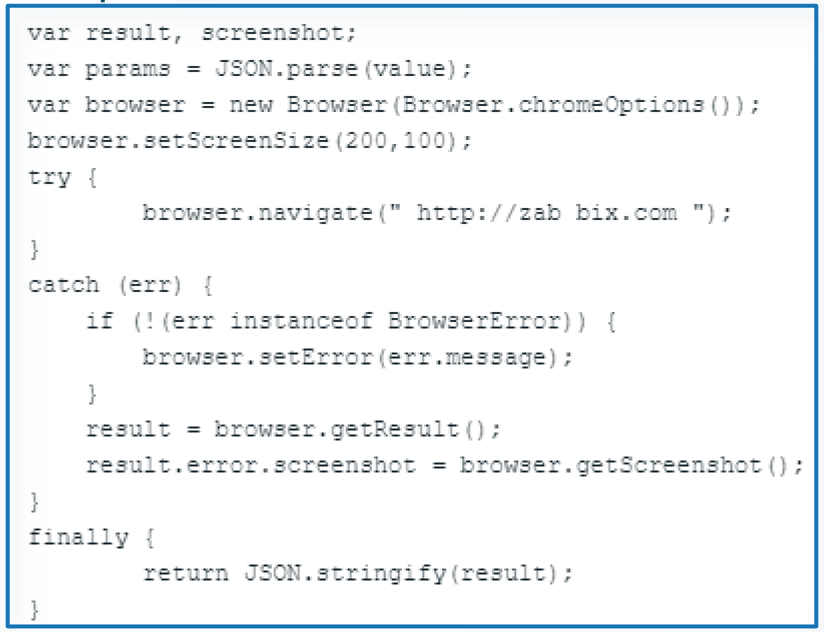
Output:

```
{"duration":1.1547865867614746,"submit_filter_text":"Apply","zabbix_server_id":"10084","disabled_button_text":"Disabled",
"browser_host_count":2,"header_id":"page-title-general"}
```

Processing errors

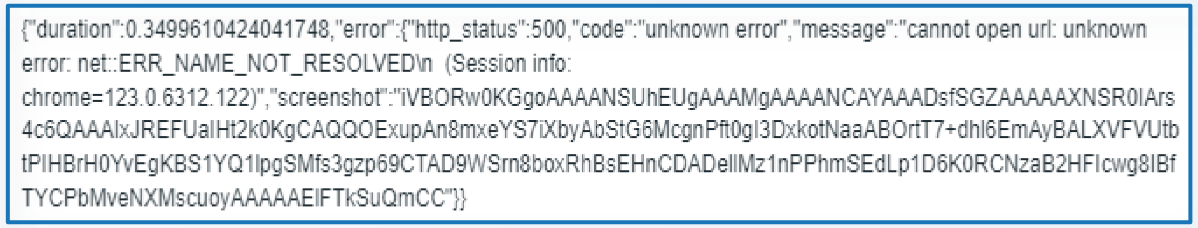
- Both ***browser errors*** and ***web driver errors*** are thrown by Browser item methods.
- ***Web driver errors*** should be specificaly set using the *setError()* method.
- Try => catch => finally statement should be used to retrieve the results.
- Screenshots of the page with error can be added.

Script:



```
var result, screenshot;
var params = JSON.parse(value);
var browser = new Browser(Browser.chromeOptions());
browser.setScreenSize(200,100);
try {
        browser.navigate(" http://zab bix.com ");
}
catch (err) {
    if (!(err instanceof BrowserError)) {
        browser.setError(err.message);
    }
    result = browser.getResult();
    result.error.screenshot = browser.getScreenshot();
}
finally {
        return JSON.stringify(result);
}
```

Output:



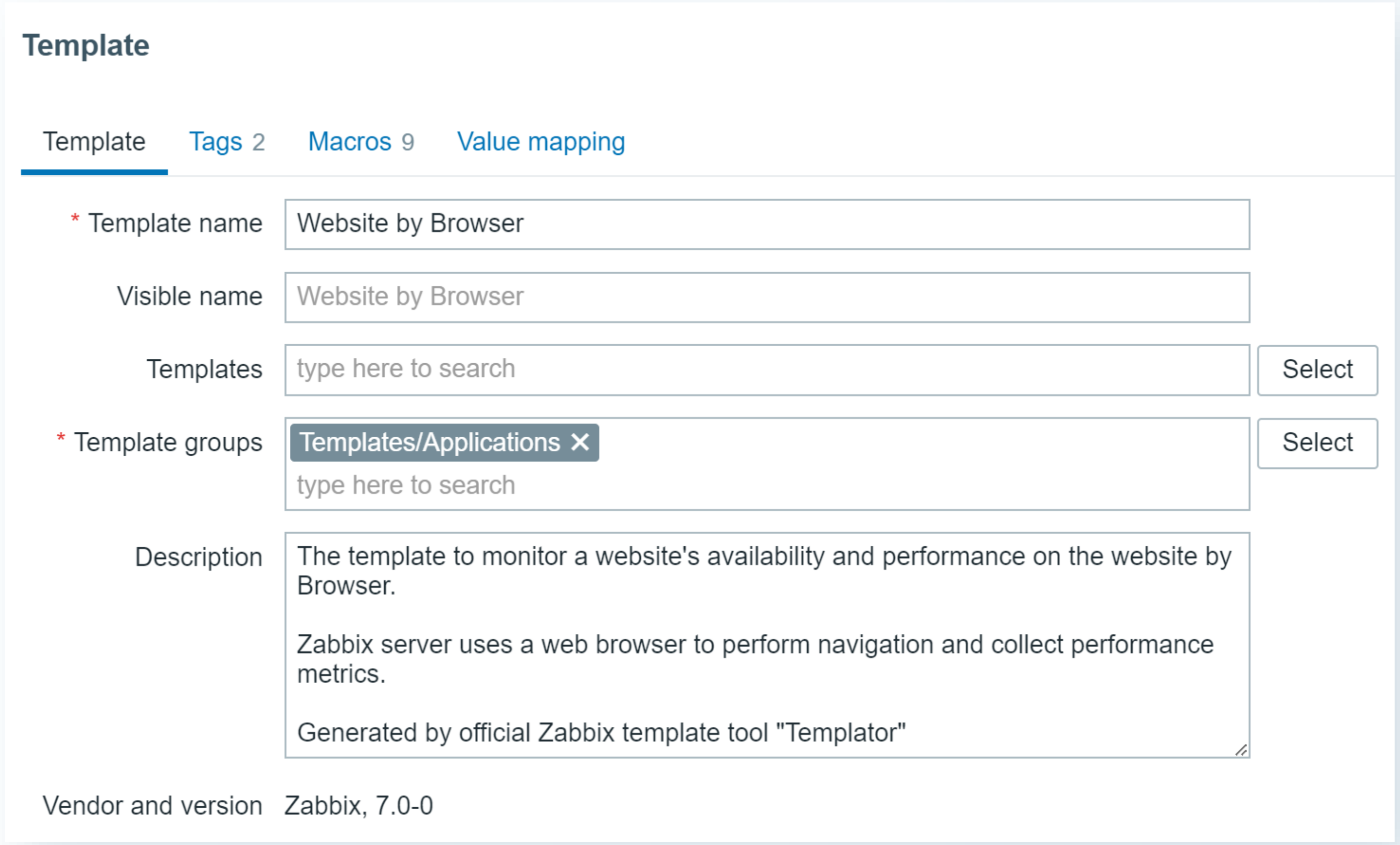
```
{"duration":0.3499610424041748,"error":{"http_status":500,"code":"unknown error","message":"cannot open url: unknown
error: net::ERR_NAME_NOT_RESOLVED\n  (Session info:
chrome=123.0.6312.122)","screenshot":"iVBORw0KGgoAAAANSUhEUgAAAMgAAAANCAYAAADsfSGZAAAAAXNSR0IArs
4c6QAAAlxJREFUalHt2k0KgCAQQOExupAn8mxeYS7iXbyAbStG6McgnPft0gl3DxkotNaaABOrtT7+dhl6EmAyBALXVFVUtb
tPIHBrH0YvEgKBS1YQ1lpgSMfs3gzp69CTAD9WSrn8boxRhBsEHnCDADellMz1nPPhmSEdLp1D6K0RCNzaB2HFIcwg8IBf
TYCPbMveNXMscuoyAAAAAElFTkSuQmCC"}}
```

ZABBIX 7.0

Out-of-box monitoring

Website by Browser template

Zabbix 7.0 comes with the "Website by Browser" template



Template

Template | Tags 2 | Macros 9 | Value mapping

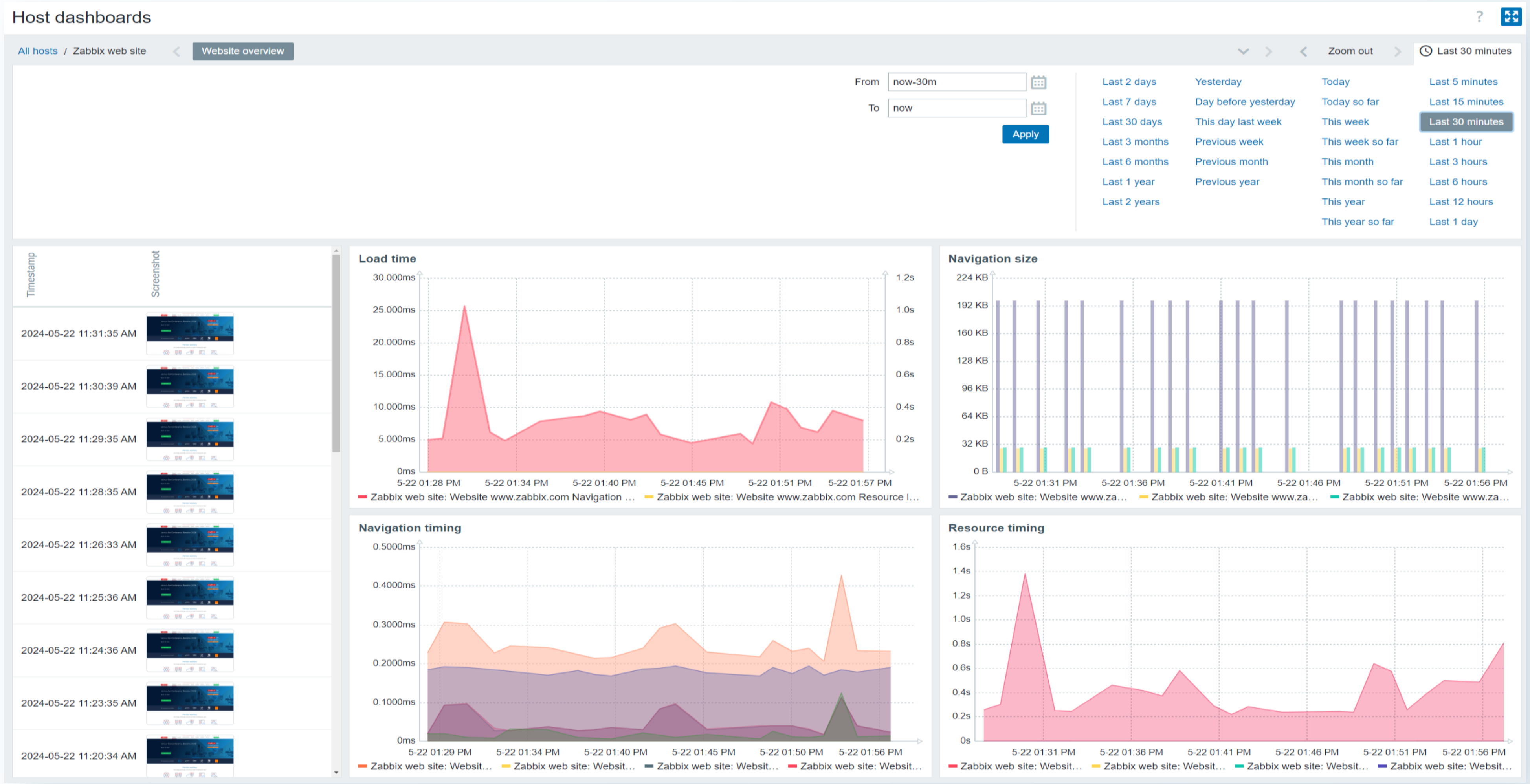
* Template name	Website by Browser
Visible name	Website by Browser
Templates	type here to search
* Template groups	Templates/Applications ✕ type here to search
Description	The template to monitor a website's availability and performance on the website by Browser. Zabbix server uses a web browser to perform navigation and collect performance metrics. Generated by official Zabbix template tool "Templator"
Vendor and version	Zabbix, 7.0-0

Template content

The new template includes:

- A "Browser" item with a data collection script
- 26 dependent items for individual metrics
- 3 predefined triggers
  - Failed to get metrics data
  - Website navigation load time is too long
  - Website resource load time is too long
- 9 User macros (browser type, website domain, screenshot dimensions, etc)
- 2 predefined graphs
- A host dashboard

Result



Host dashboards

All hosts / Zabbix web site — Website overview

From: now-30m

To: now

Last 30 minutes

Timestamp	Screenshot
2024-05-22 11:31:35 AM	
2024-05-22 11:30:39 AM	
2024-05-22 11:29:35 AM	
2024-05-22 11:28:35 AM	
2024-05-22 11:26:33 AM	
2024-05-22 11:25:36 AM	
2024-05-22 11:24:36 AM	
2024-05-22 11:23:35 AM	
2024-05-22 11:20:34 AM	

ZABBIX 7.0

Thank you

Sergejs Olonkins
Quality assurance engineer