



ZABBIX
PREMIUM PARTNER

ZABBIX
CERTIFIED TRAINER

Workshop

Zabbix browser Item Monitoring using Selenium

All our microphones are muted

Ask your questions in Q&A, not in the Chat

Use Chat for discussion, networking or applause

1

Synthetic end-user web monitoring



Synthetic end-user web monitoring

- Synthetic web monitoring
 - Simulate an interaction of a real user with a real browser
 - Performed by writing custom scripts
 - Test various aspects of a website or web application – availability, performance, transaction statuses and more.

Synthetic end-user web monitoring

- Synthetic web monitoring in Zabbix
 - A Selenium Server or a plain WebDriver to perform synthetic web monitoring
 - Selenium Server utilizes a headless browser to perform tests
 - Selenium Server is not provided as a part of Zabbix packages
 - The simplest way to get started – Selenium Docker containers

Synthetic end-user web monitoring

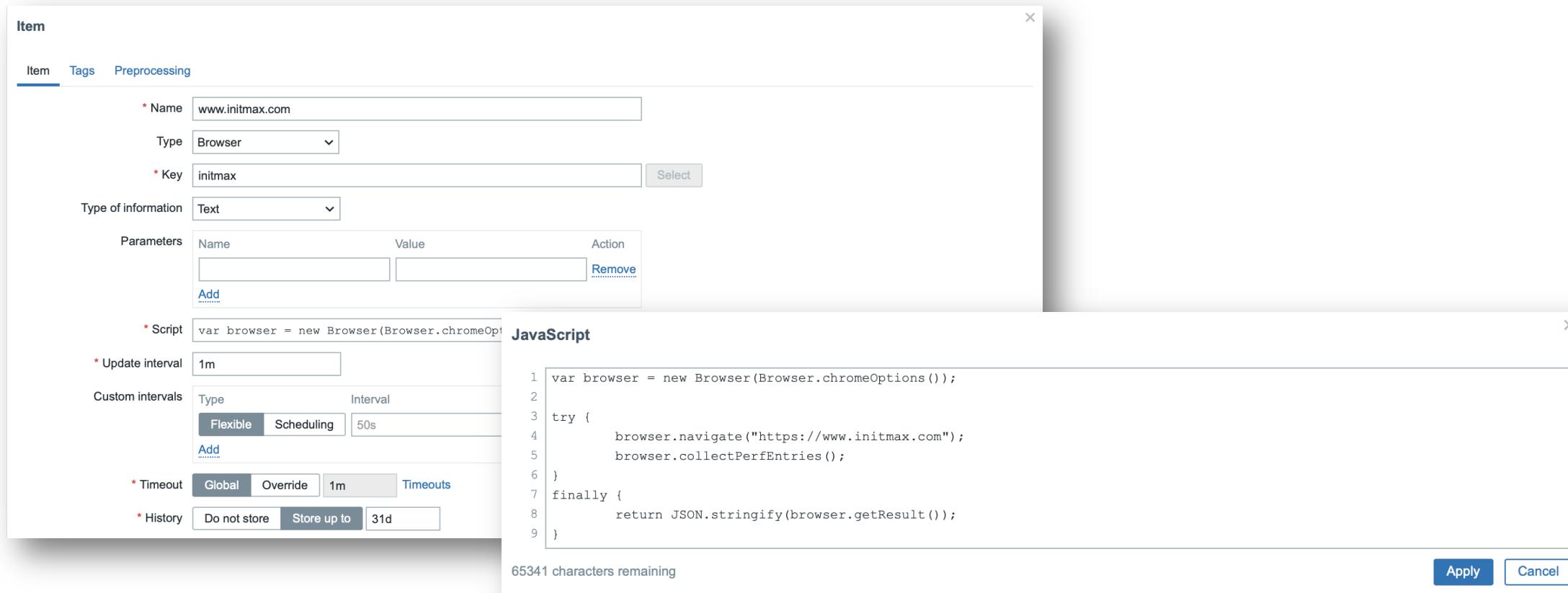
- Synthetic web monitoring in Zabbix
 - Point Zabbix server at the WebDriver URL
 - Browser pollers are responsible for polling Browser items

```
### 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.X.X:4444

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

Synthetic end-user web monitoring

➤ New item type - Browser



The screenshot shows the 'Item' configuration dialog in Zabbix. The 'Tags' tab is selected. The 'Name' field contains 'www.initmax.com'. The 'Type' is set to 'Browser'. The 'Key' is 'initmax'. The 'Type of information' is 'Text'. There are no parameters defined. The 'Script' field contains the following JavaScript code:

```
var browser = new Browser(Browser.chromeOptions());
try {
    browser.navigate("https://www.initmax.com");
    browser.collectPerfEntries();
}
finally {
    return JSON.stringify(browser.getResult());
}
```

The 'Update interval' is set to '1m'. Under 'Custom intervals', 'Flexible' is selected with an interval of '50s'. The 'Timeout' is 'Global' with an override of '1m'. The 'History' settings show 'Do not store' and 'Store up to 31d'. The status bar at the bottom indicates '65341 characters remaining'.

Synthetic end-user web monitoring

- The Browser item uses JavaScript to collect data in JSON
 - The default script collects performance entries and session statistics
 - Website by Browser template is available for more complex scenarios

Website by Browser

[Overview](#)

[Requirements](#)

Zabbix version: 7.0 and higher.

Tested versions

This template has been tested on:

- ChromeDriver 124.0.6367.207, selenium-server-4.0.0-alpha-6

Configuration

Zabbix should be configured according to the instructions in the [Templates out of the box](#) section.

Setup

Install WebDriver. For more information, please refer to the [Selenium WebDriver](#) page. Run selenium-server. Add in configuration file WebDriver interface [HTTP\[S\]](#) URL. For example <http://localhost:4444>

Synthetic end-user web monitoring

- Website by Browser template contains:
 - Items for website navigation and resource statistics
 - Current website screenshot
 - Triggers for slow load times and website availability
 - A dashboard displaying website screenshot and various performance statistics

Synthetic end-user web monitoring

- The new Binary type of information can be used to collect and store images
 - Browser item can collect screenshots of the website in base64 format
 - It can be stored in a binary dependent item

Item

Item Tags 1 Preprocessing 1

* Name: Website Screenshot

Type: Dependent item

* Key: website.screenshot

Select

Type of information: Binary

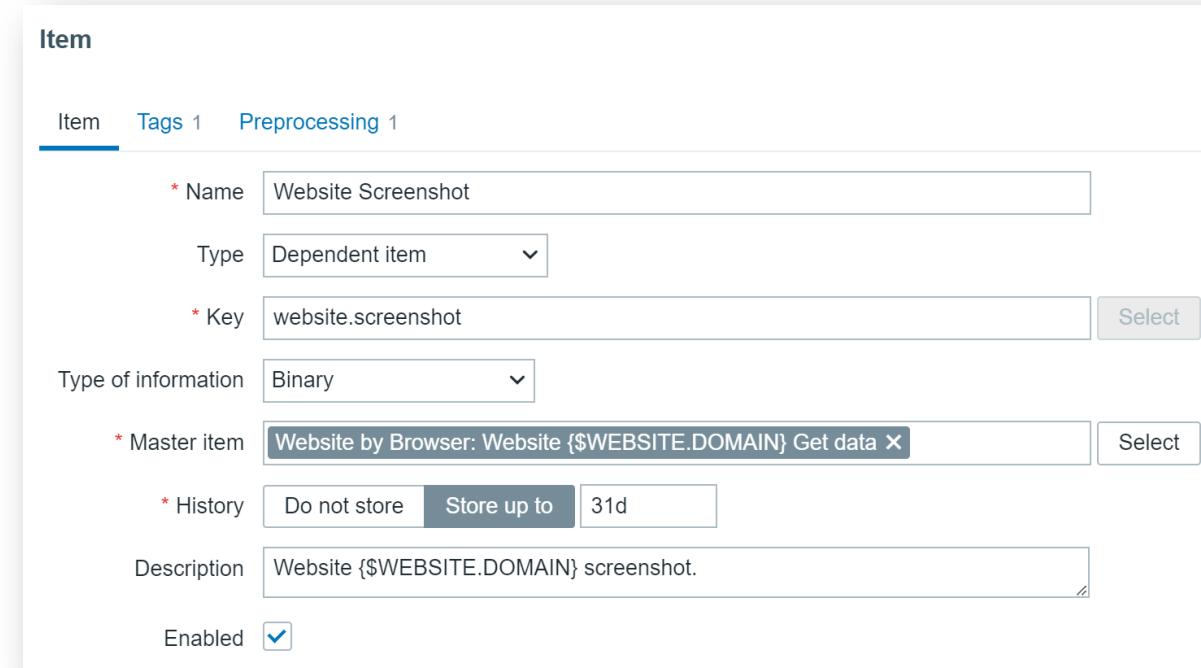
* Master item: Website by Browser: Website {\$WEBSITE.DOMAIN} Get data

Select

* History: Do not store Store up to: 31d

Description: Website {\$WEBSITE.DOMAIN} screenshot.

Enabled:



Synthetic end-user web monitoring

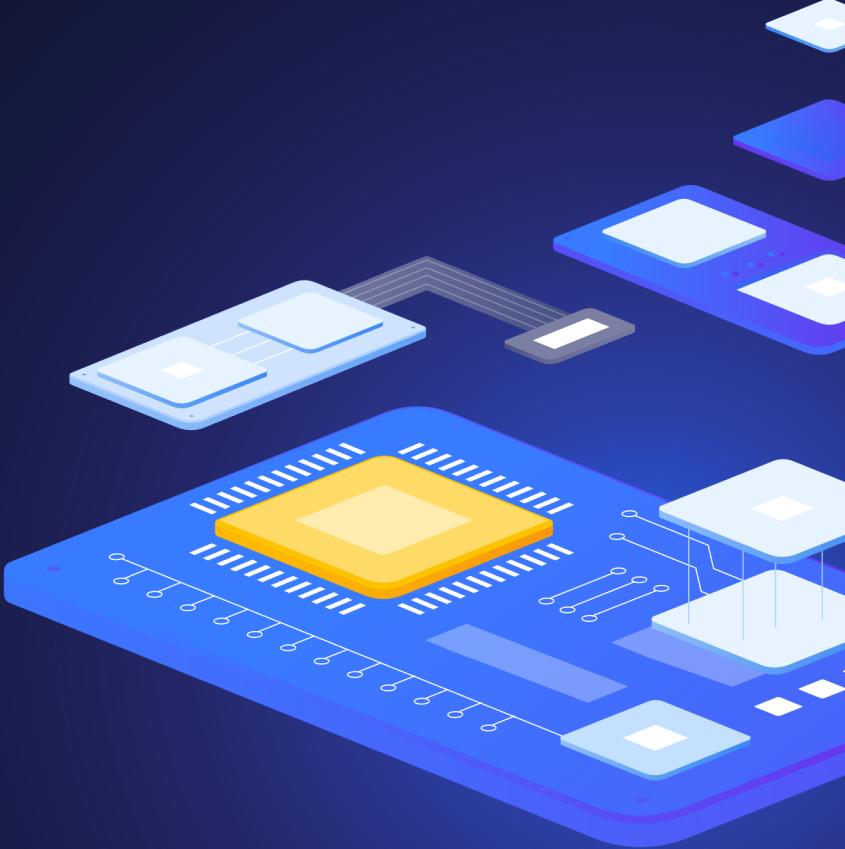
- The item history widget can now display screenshots:



- TIP: <https://www.initmax.com/wiki/installation-and-basic-usage-of-browser-item/> EN
- TIP: <https://www.initmax.cz/wiki/instalace-a-zakladni-pouziti-pro-item-browser/> CZ

2

Used sources



Used sources

➤ Zabbix documentation

<https://www.zabbix.com/documentation/7.0/en/manual/config/items/itemtypes/browser?hl=Selenium>

https://www.zabbix.com/documentation/7.0/en/manual/config/items/preprocessing/javascript/browser_item_javascript_objects

➤ Selenium installation

<https://www.initmax.com/wiki/installation-and-basic-usage-of-browser-item/> EN

<https://www.initmax.cz/wiki/instalace-a-zakladni-pouziti-pro-item-browser/> CZ

➤ Browser plugins

<https://www.selenium.dev/selenium-ide/>

➤ Selenium

<https://github.com/SeleniumHQ/docker-selenium/tree/trunk>

Demo – introduction of lab



Demo Lab

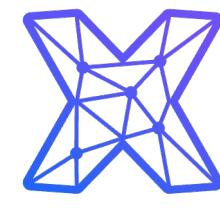
- We are using pre-installed Zabbix server version 7.0 and one additional node for docker
 - student-01.initmax.cz – Zabbix server 7.0 (installed)
 - student-02.initmax.cz – Docker server for Selenium (part of workshop)



Zabbix server
(student-01.initmax.cz)



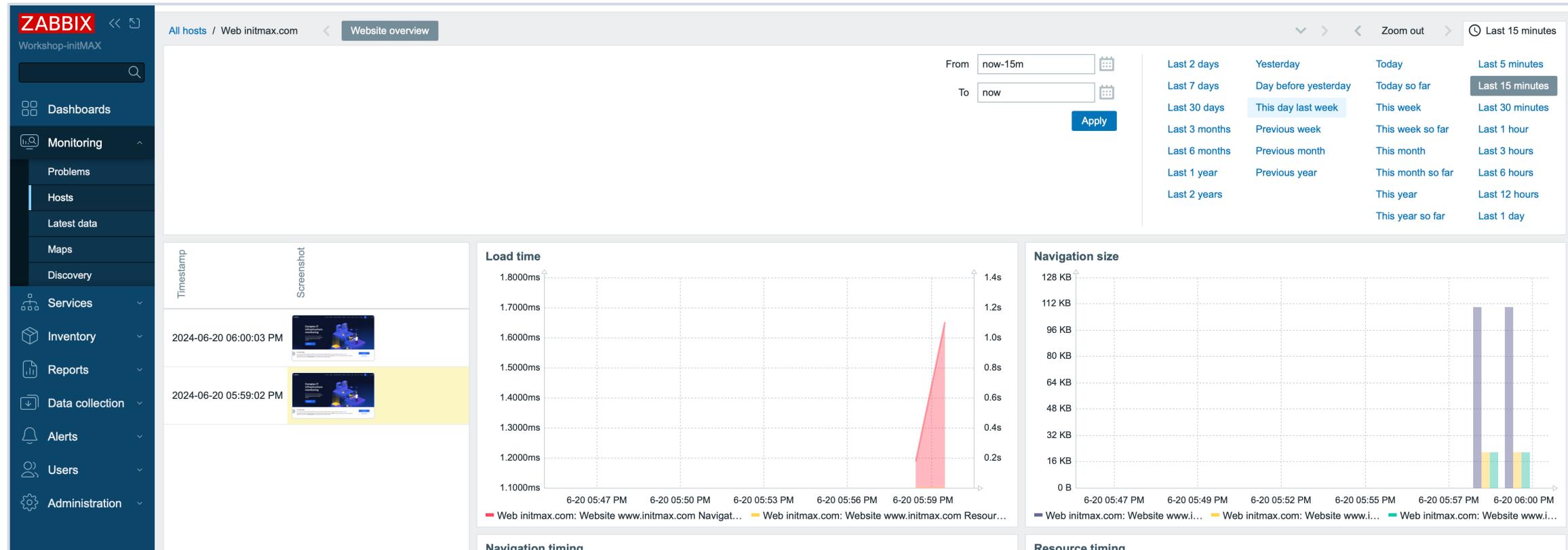
Selenium server
(student-02.initmax.cz)



Monitored webpage
(www.initmax.com)

Browser item script

- ▶ First part load webpage and create screenshot



Browser item script

- Add new macro to item browser and host

{\$WEBSITE.NEWSLETTER.EMAIL} = your email address

```
setParams(params) {
    ['scheme', 'domain','width', 'height', 'email'].forEach(function (field) {
        if (typeof params !== 'object' || !params[field]) {
            throw new Error('Required param is not set: ' + field + '.');
        }
    });
    this.params = params;
},
```

Browser item script

➤ Hide cookies bar

```
// Accept Cookies to hide this element
el = browser.findElement("xpath", "//a[contains(text(),'I accept all')]");
if (el === null) {
    throw Error("cannot find Cookie button");
}
el.click();
```

Browser item script

- Click on Wiki in menu

```
// Click on Wiki
el = browser.findElement("xpath", "//a[contains(text(),'Wiki')]");
if (el === null) {
    throw Error("cannot find Wiki button");
}
el.click();
```

Browser item script

- Browse menu and click on Zabbix

```
// Click on Wiki > Zabbix
el = browser.findElement("xpath", "//li[3]/strong/span");
if (el === null) {
    throw Error("cannot find Zabbix button in menu");
}
el.click();
```

Browser item script

- Browse menu and click on Zabbix > Guides

```
// Click on Wiki > Zabbix > Guides
el = browser.findElement("xpath", "//strong[3]/span");
if (el === null) {
    throw Error("cannot find Guides button in menu");
}
el.click();
```

Browser item script

- Browse menu and click on Zabbix > Guides > Topic

```
// Click on Wiki > Zabbix > Guides > Topic
el = browser.findElement("xpath", "//a[contains(text(),'Installation and basic usage of Item Browser')]");
if (el === null) {
    throw Error("cannot find topic: 'Installation and basic usage of Item Browser' in menu");
}
el.click();
```

Browser item script

- Fill newsletter field with **email** parameter (Our defined macro)

```
// Click and fill newsletter field
el = browser.findElement("xpath", "//input[@id='newsletter-form_contact_email']");
if (el === null) {
    throw Error("Newsletter form not found!");
}
el.click();
el.sendKeys(Website.params.email);
```

Browser item script

➤ Accept GDPR

```
// Click on agree to the processing of personal data
el = browser.findElement("xpath", "//section[@id='newsletter']/div/div/div[2]/div/form/div/div[2]/label/div");
if (el === null) {
    throw Error("Button: 'I agree to the processing of personal data.' has not be found!");
}
el.click();
```

Browser item script

- Submit newsletter form

```
// Send newsletter form
el = browser.findElement("xpath", "//section[@id='newsletter']/div/div/div[2]/div/form/div/a");
if (el === null) {
    throw Error("Newsletter form cannot be sent!");
}
el.click();
```

Browser item script

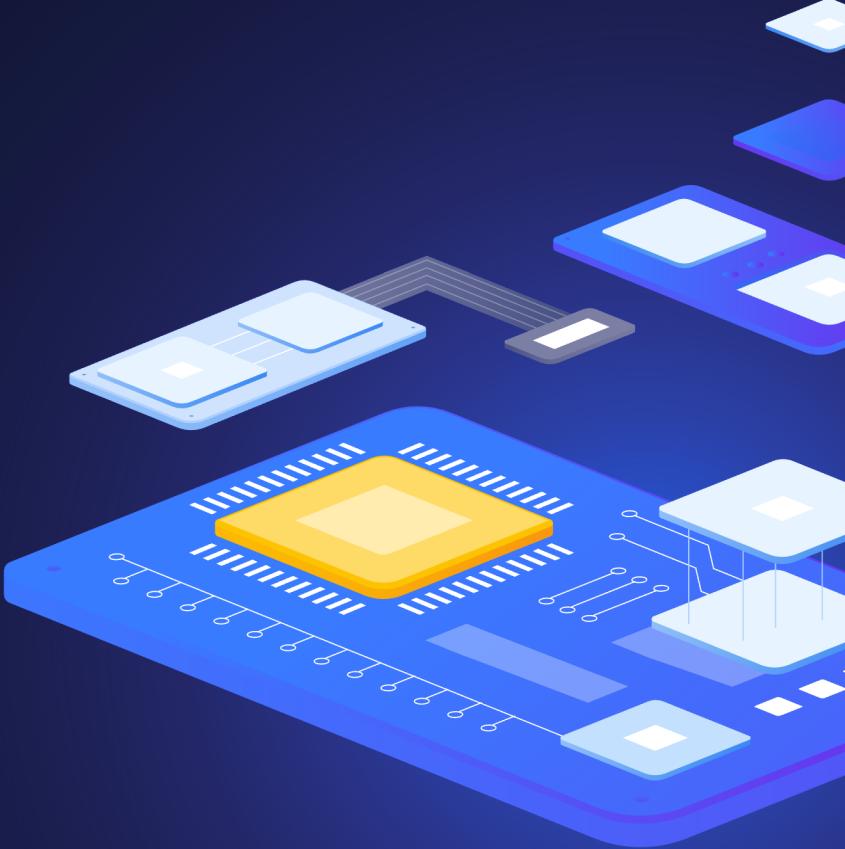
- Use work around for timeout

```
// Wait for a specific element that indicates the form has been successfully sent
browser.setElementWaitTimeout(10000); //10s
var loggedInElement = browser.findElement("xpath", "//h3[contains(.,'SOMETHING IMPOSSIBLE')]");
if (loggedInElement === null) {
}
```

Demo



Questions?



Contact us:

Phone:



+420 800 244 442

Web:



<https://www.initmax.cz>

Email:



tomas.hermanek@initmax.cz

LinkedIn:



<https://www.linkedin.com/company/initmax>

Twitter:



<https://twitter.com/initmax>

Tomáš Heřmánek:



+420 732 447 184