

# Web監視の進化系 「ブラウザアイテム」

ZABBIX '24

CONFERENCE

JAPAN



Zabbix Japan LLC

サポートエンジニア兼Zabbix 認定トレーナー

渡邊 隼人

#ZabConfJp2024

# 自己紹介1

- 渡邊 隼人 (わたなべ はやと)  
✓ サポートチーム チームリーダー
- Conference登壇9年目くらい?  
✓ 1ページで30分話せるくらいに成長
- 懇親会やテクニカル相談会、いつでも気軽に話しかけてください

# Zabbix 7.0 LTS、今年最大の私のトピック

- Interop24のマネジメント & モニタリング部門で最高賞🏆



「Best of ShowAward」 をもらい浮かれるエンジニア 渡邊 隼人メンバー (42)

# 今年のトピック

- Interop24のマネジメント & モニタリング部門で最高賞🏆

絶対に私のプレゼンと7.0 LTSの機能が良かったから

「Best of ShowAward」をもらい浮かれるエンジニア 渡邊 隼人メンバー (42)



# 7.0 LTS のWeb監視

ZABBIX '24

CONFERENCE

JAPAN

# これまでのWeb監視

- ZabbixのWeb監視といえば
- Web監視
  - ✓ 1 ~ n階層のステップ実行
  - ✓ シナリオやステップから情報取得
    - ダウンロード速度
    - 応答時間
    - 応答コード
- HTTPエージェント/スクリプト
  - ✓ HTTP(s)でリクエストでコンテンツを取得
- Zabbixエージェント [web.page.xxxxx]
  - ✓ Zabbixエージェント経由で簡単な情報を取得
- これまでより、複雑なWebサイトやWebアプリケーションの監視が必要  
→ ブラウザアイテムの登場

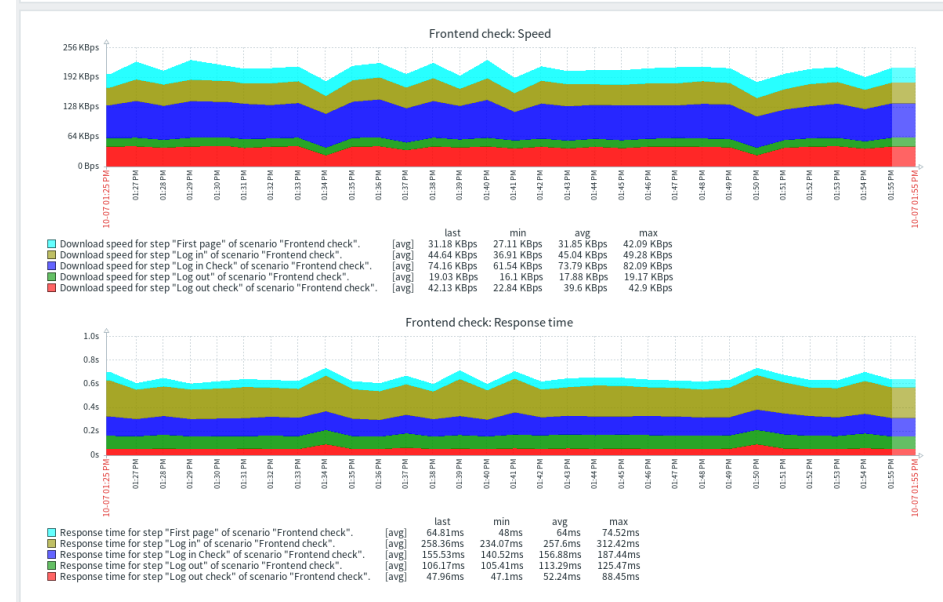
Details of web scenario: Frontend check

Step	Speed	Response time	Response code	Status
First page	31.18 KBps	64.81ms	200	OK
Log in	44.64 KBps	258.36ms	200	OK
Log in Check	74.16 KBps	155.53ms	200	OK
Log out	19.03 KBps	106.17ms	200	OK
Log out check	42.13 KBps	47.96ms	200	OK
<b>TOTAL</b>		<b>632.82ms</b>		<b>OK</b>

From  To

Zoom out Last 30 minutes

Last 2 days	Yesterday	Today	Last 5 minutes
Last 7 days	Day before yesterday	Today so far	Last 15 minutes
Last 30 days	This day last week	This week	Last 30 minutes
Last 3 months	Previous week	This week so far	Last 1 hour
Last 6 months	Previous month	This month	Last 3 hours
Last 1 year	Previous year	This month so far	Last 6 hours
Last 2 years		This year	Last 12 hours
		This year so far	Last 1 day



# 新しいWeb監視

アイテムタイプ「ブラウザ」

#ZabConfJp2024

6

ZABBIX '24

CONFERENCE

JAPAN

# 「ブラウザ」 アイテムとは



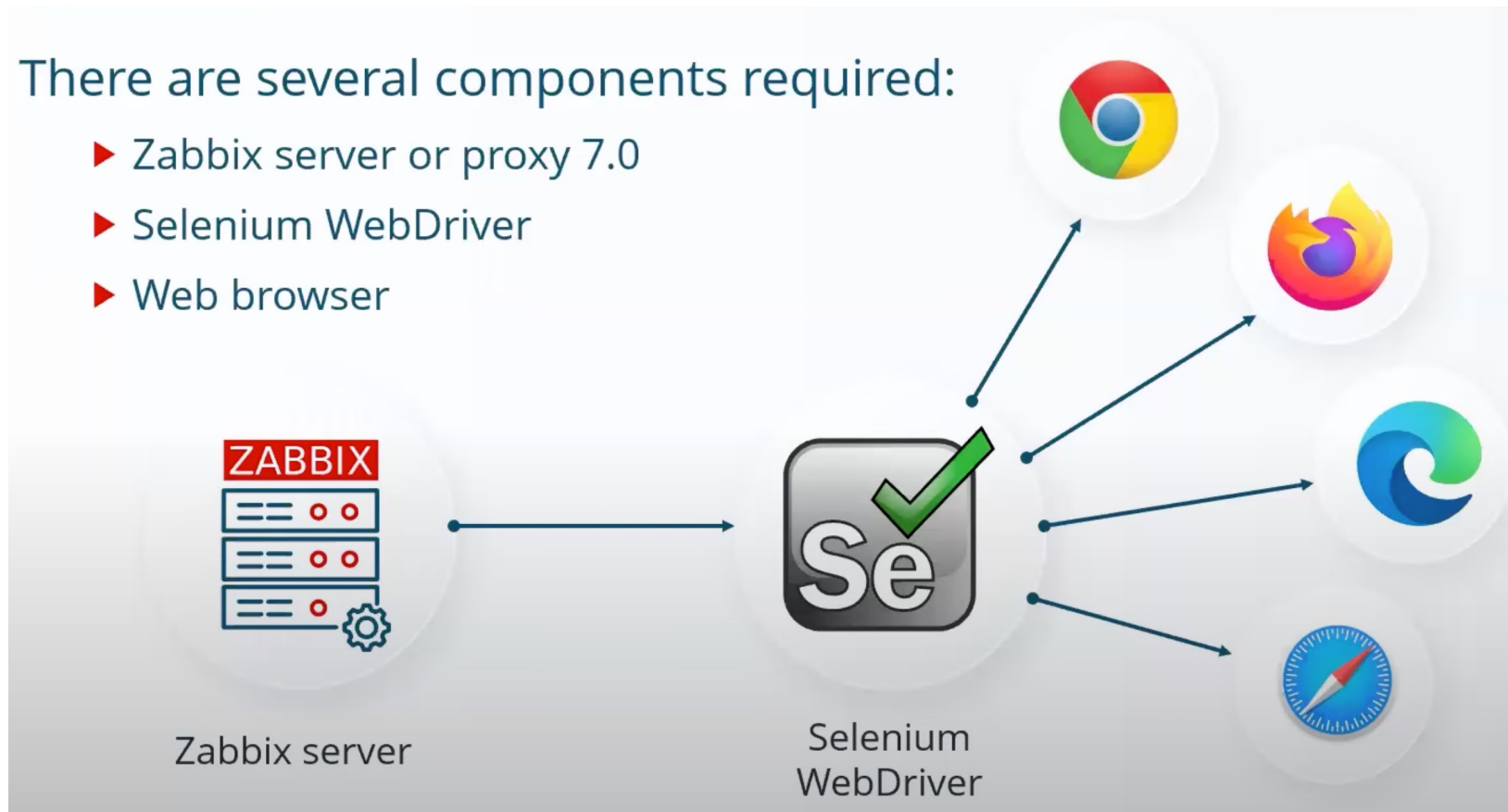
# ブラウザアイテムとは？

- SeleniumとWebドライバを使用し、chrome等のブラウザを利用してWebページにアクセスする監視アイテム
- Seleniumとは...
  - ✓ Seleniumはブラウザ自動化を可能にし、それを支えるツール群とライブラリー群のプロジェクト
  - ✓ Selenium WebDriverを使用し、Zabbixからブラウザをネイティブに操作する
- 監視処理はJavaScriptでコーディング。

# 構成イメージ

There are several components required:

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



## 構成

Zabbix serverかproxy 7.0以上

Selenium WebDriver

Webブラウザ

## サポートブラウザ

Google Chrome

Microsoft Edge

Mozilla Firefox

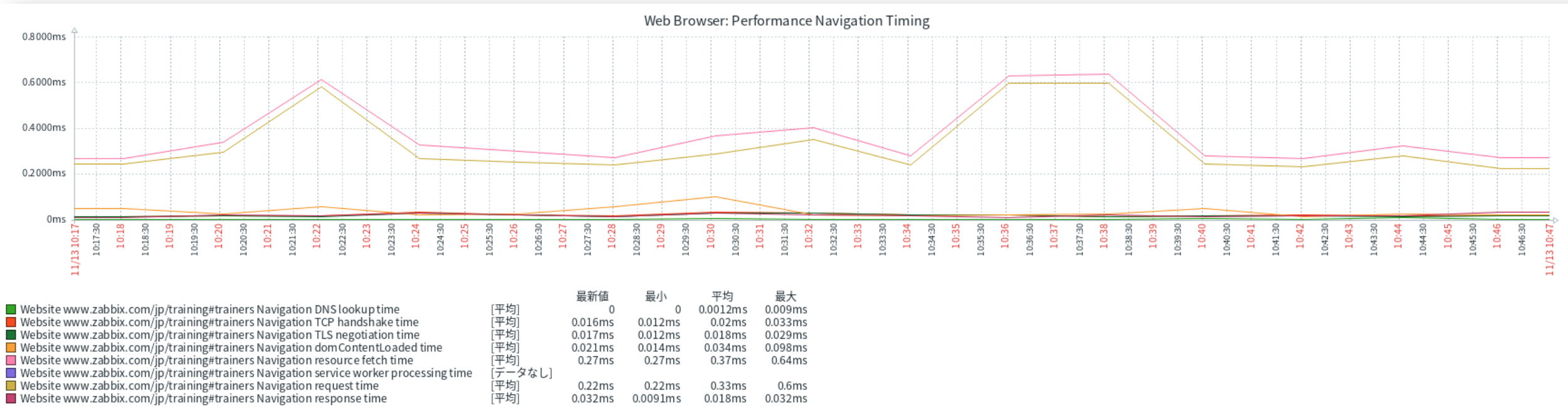
Safari

Zabbix Meetup online, August 2024: Synthetic browser monitoring in Zabbix 7.0

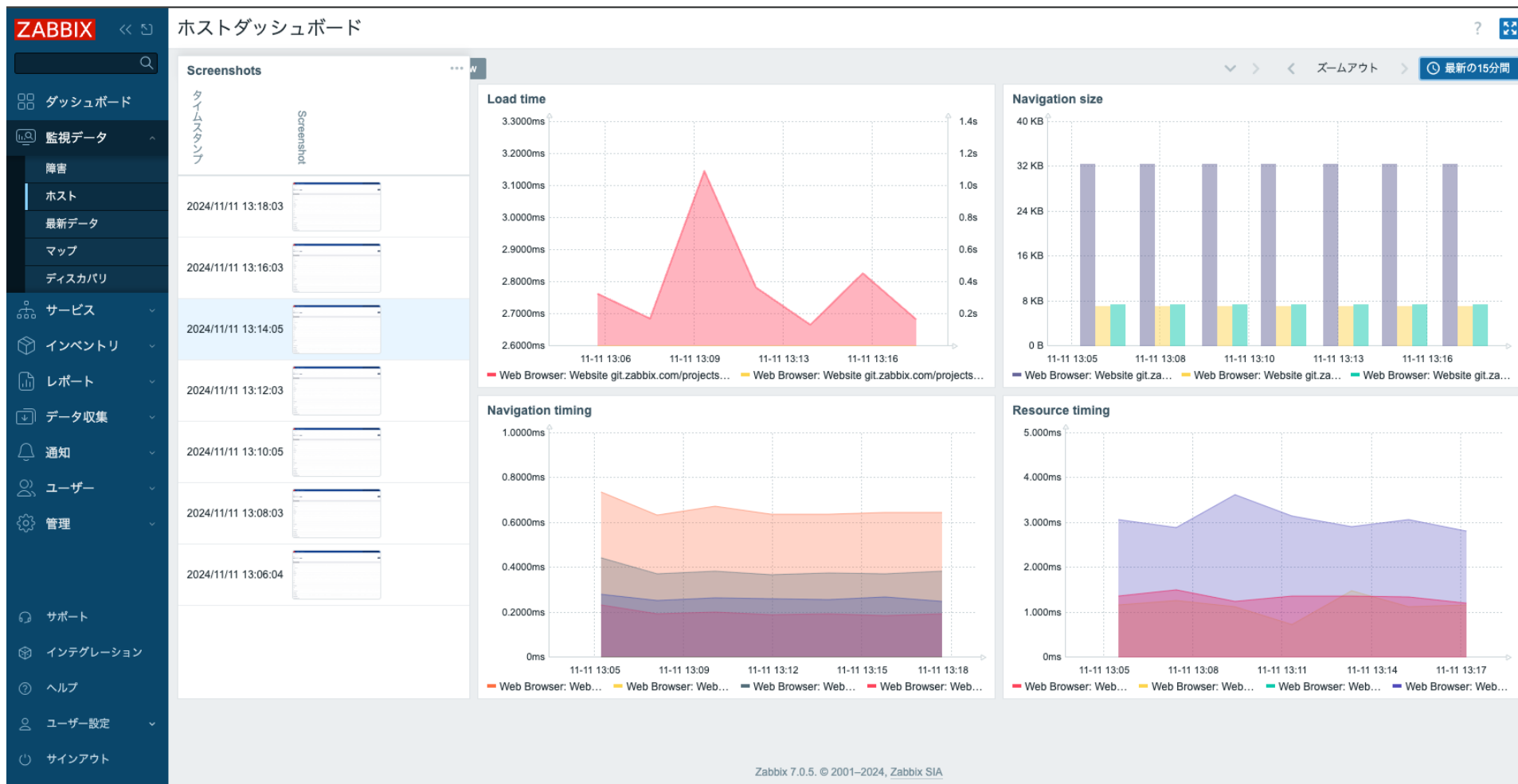
[https://www.youtube.com/watch?v=7SSwdA\\_4kiY](https://www.youtube.com/watch?v=7SSwdA_4kiY)

# ブラウザアイテムで取得できる情報

- 結果取得までの時間
- ステータスコード、エラーメッセージ
- パフォーマンス情報、スクリーンショット



# 取得した結果



# 取得するスクリーンショット情報

- サイズ
  - ✓ デフォルト  
1920 x 1080 px
  - ✓ 最大  
8000 x 8000 px
- 表示方法
  - ✓ ウィジェット  
「アイテムのヒストリ」
- フォント
  - ✓ Selenium環境に依存  
※日本語フォント注意

ZABBIX

製品 ソリューション & 導入事例 サービス & サポート トレーニング パートナー コミュニティ ZABBIX社について

ダウンロード

## Zabbix公式トレーニングプログラム/認定資格

たった数日でZabbixのほぼすべての機能をご理解いただくために、経験豊富な講師が行うZabbix公式トレーニングプログラムを開発しました。

トレーニング受講者に行ったアンケートの集計結果によると、全体の99%の方にご満足いただいたとの高い評価を頂いております。詳細は、受講者の意見に掲載しております。

概要 プログラム スケジュール オプション 講師 受講者の声 F.A.Q.

### トレーニングのメリット

- 公式認定**  
トップトレーナーの指導のもと、Zabbixをマスターすることで、Zabbix公式認定資格を取得することができます。
- ハンズオントレーニング**  
実践的なタスクをこなし、実際のユースケースを解決することで、新しいスキルを身につけることができます。
- 体系化されたナレッジ**  
Zabbix認定トレーニングはZabbixのすべてを習得するための最も時間的、費用対効果の高い方法です。
- 明確なトレーニングの道**  
世界中で受講可能
- 効率性の向上**

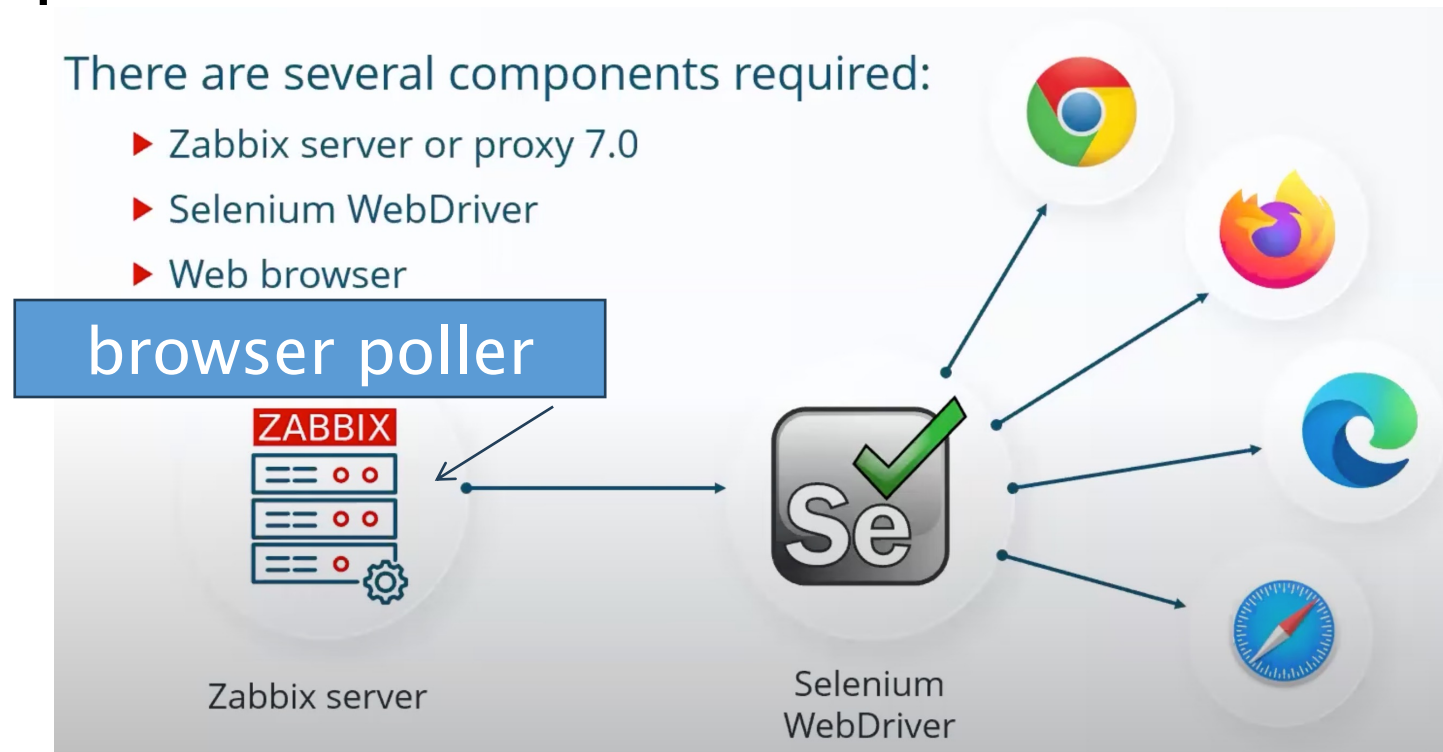
# 認定資格



# ブラウザアイテムの仕組み

# ブラウザアイテム監視を行う為のプロセス

- browser pollerでSeleniumへポーリングリクエスト



Zabbix Meetup online, August 2024: Synthetic browser monitoring in Zabbix 7.0  
[https://www.youtube.com/watch?v=7SSwdA\\_4kiY](https://www.youtube.com/watch?v=7SSwdA_4kiY)

# browser poller

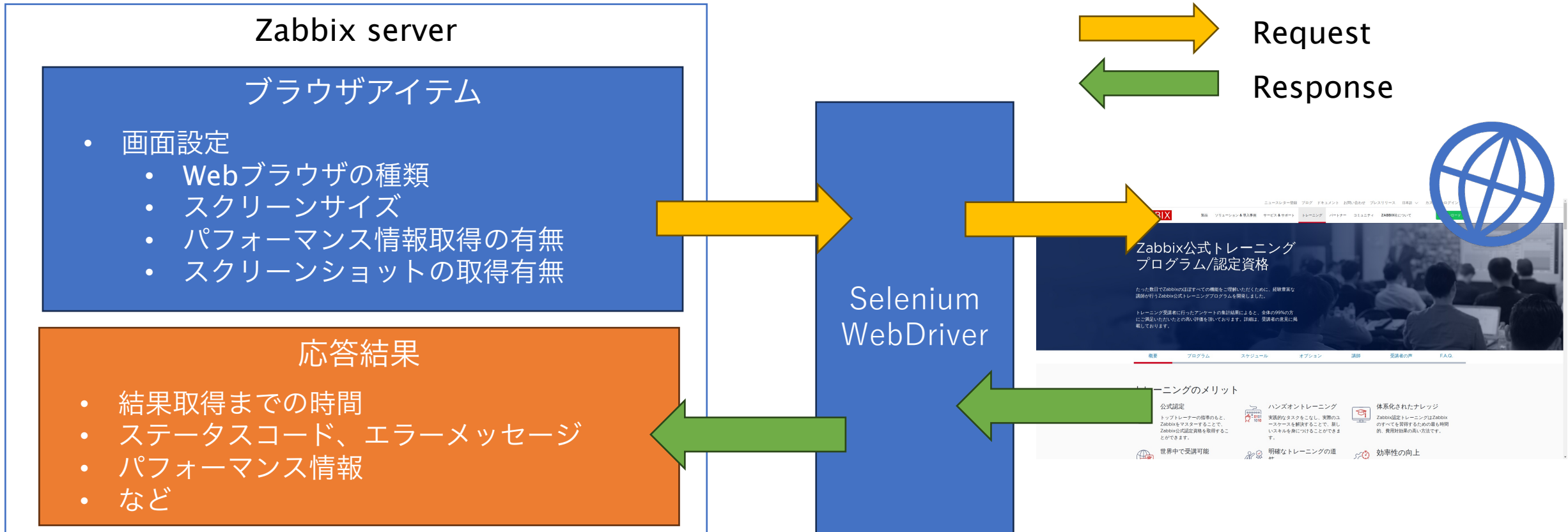
- ブラウザアイテムのJavaScriptコードからWebブラウザを指定しURLに接続
- JavascriptエンジンはDuktapeを使用
  - ✓ 言語で書かれたコンパクトでメモリ使用量の少ないJavaScriptエンジン
  - ✓ Node.js等、サーバーサイドのJavaScriptで使用するrequireは使えない

```
In zbx_es_execute() param:{"browser":"chrome","domain":"git.zabbix.com ...
opened webdriver session with sessionid:9ffe3b56c59f1da992571a7b61eab668
webdriver_release()
webdriver_destroy()
closed webdriver session 9ffe3b56c59f1da992571a7b61eab668
zbx_es_execute() output: '{"duration":3.0035617351531982,"performance_data ...
End of zbx_es_execute():SUCCEED allocated memory: 120477 max allocated or requested memory: 383043 max
allowed memory: 536870912
```

zabbix\_server.logの browser pollerプロセスをデバッグモード出力した情報

# ブラウザアイテムで使用するJavaScriptオブジェクト

- Javascript



# 設定方法

#ZabConfJp2024

17

ZABBIX '24

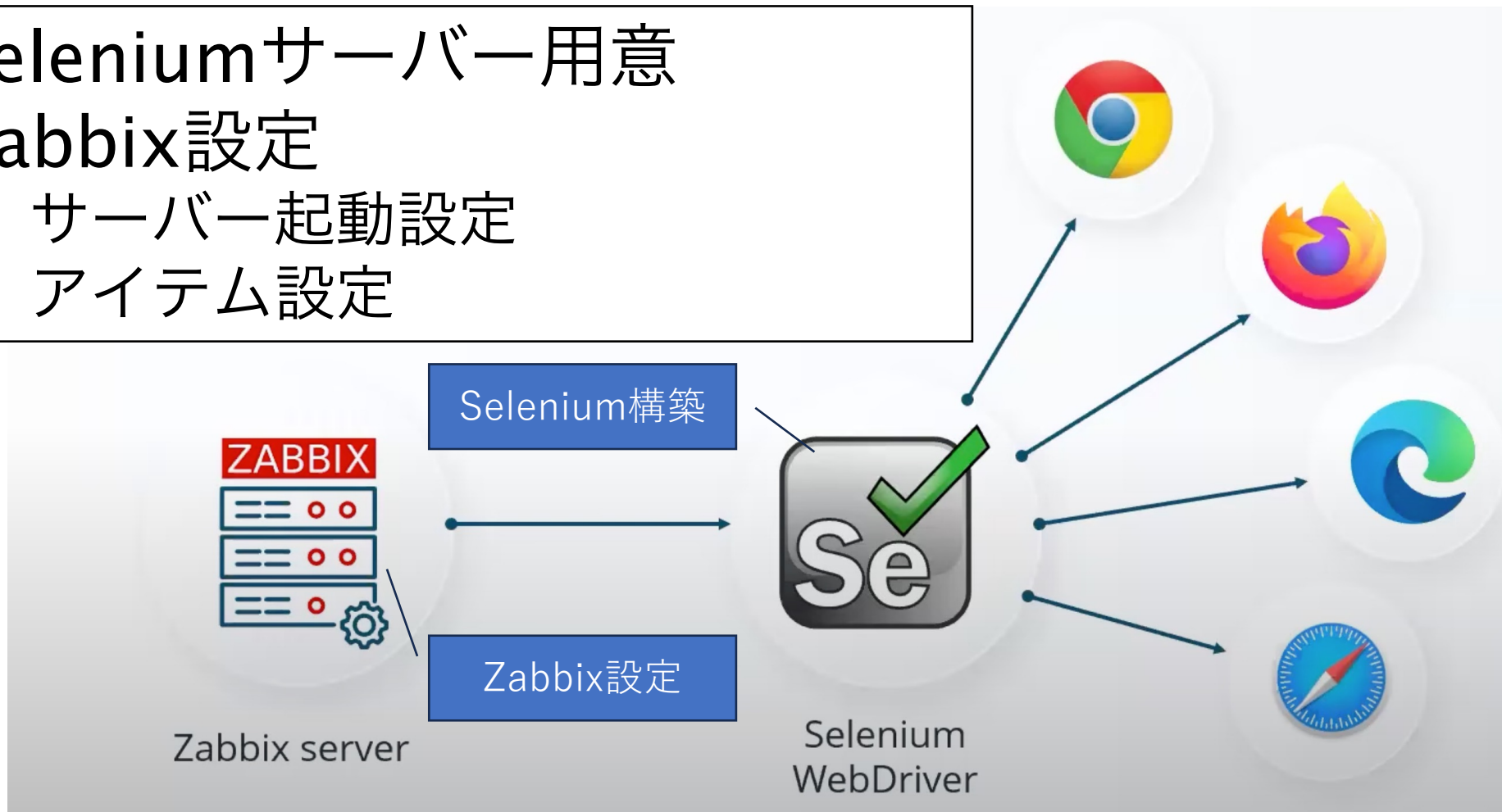
CONFERENCE

JAPAN



# ブラウザ監視までの設定

- Seleniumサーバー用意
- Zabbix設定
  - ✓ サーバー起動設定
  - ✓ アイテム設定



# Seleniumサーバー用意

- Dockerで起動
  - ✓ Selenium公式コンテナを使用
  - ✓ 使用するメモリサイズはクラッシュ回避のため2GBを指定

```
docker run --name browser ¥  
-p 4444:4444 ¥  
-p 7900:7900 ¥  
--shm-size="2g" ¥  
-d selenium/standalone-chrome:latest
```

<https://hub.docker.com/r/selenium/standalone-chrome>

- コンテナを使用せずスタンドアローンで作成

[https://www.selenium.dev/documentation/grid/getting\\_started/#standalone](https://www.selenium.dev/documentation/grid/getting_started/#standalone)

- 今回は簡単に起動できるDockerを使用

# Zabbixサーバー用意

- zabbix\_server.conf

オプション	内容	値
WebDriverURL	Selenium WebDriverサーバ	127.0.0.1:4444
StartBrowserPollers	ブラウザ監視用Pollerプロセス	デフォルト: 1

```
WebDriverURL=127.0.0.1:4444  
StartBrowserPollers=1
```

- Zabbixサーバーのサービスは再起動が必要

```
# systemctl restart zabbix-server
```

# Zabbixサーバー用意

- シンプルなアイテム設定

The screenshot shows the Zabbix 'New Item' configuration interface. The main form is partially visible on the left, showing fields for Name (Zabbix website), Type (Browser), Key (website.get.data), and Data type (Text). A JavaScript code editor window is overlaid on the right, containing the following code:

```
1 var browser = new Browser(Browser.chromeOptions());  
2  
3 try {  
4     browser.navigate("https://www.zabbix.com/jp/");  
5     browser.collectPerfEntries();  
6 }  
7 finally {  
8     return JSON.stringify(browser.getResult());  
9 }
```

The code editor shows a character count of 'あと65338文字' (65338 characters remaining) and buttons for '適用' (Apply) and 'キャンセル' (Cancel). A blue callout box with a white arrow points to the URL in the code, containing the text 'https://example.com を変更' (Change https://example.com).

# Zabbixサーバー用意

- シンプルなアイテム設定

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

シンプルな設定で使ったBrowserオブジェクトのメソッド

メソッド	内容
navigate(url)	URLを設定
collectPerfEntries()	パフォーマンスエントリを収集
getResult()	セッションの取得時間、エラー、パフォーマンスデータ、スナップショット等のオブジェクトを返す



# Zabbixサーバー用意

- アイテム設定

The screenshot shows the Zabbix 'New Item' configuration page. The item name is 'Zabbix website', type is 'Browser', and key is 'website.get.data'. The data type is 'Text'. The script field contains a JavaScript snippet for browser performance monitoring. A modal window titled 'JavaScript' is open, showing the script code. A large white text overlay 'これだけ' (Just this) points to the script. A blue box below the script says 'https://example.com を変更' (Change https://example.com).

```
JavaScript
```

```
1 var browser = new Browser(Browser.chromeOptions());  
2  
3 try {  
4   browser.navigate("https://www.zabbix.com/jp/");  
5   browser.collectPerfEntries();  
6 }  
7 finally {  
8   return JSON.stringify(browser.getResult());  
9 }
```

あと65338  
適用 キャンセル

これだけ

https://example.com を変更

追加 テスト キャンセル

# 取得した情報

```
{"duration":2.0073800086975098,"performance_data":{"details":[{"navigation":{"entry_type":"navigation","dom_content_loaded_event_start":1.0294000000953674,"domain_lookup_start":0.0018000000715255738,"tls_negotiation_time":0.010899999976158143,"request_start":0.014700000047683716,"redirect_start":0,"load_event_start":1.9605,"name":"https://www.zabbix.com/jp/","connect_end":0.014300000071525574,"dom_content_loaded_event_end":1.0697000000476837,"critical_ch_restart":0,"decoded_body_size":231423,"response_end":0.28780000007152556,"secure_connection_start":0.0038000000715255737,"initiator_type":"navigation","encoded_body_size":53956,"render_blocking_status":"non-blocking","not_restored_reasons":null,"worker_start":0,"unload_event_start":0,"response_status":200,"dom_interactive":1.0292000000476837,"start_time":0,"fetch_start":0.001,"activation_start":0,"redirect_end":0,"unload_event_end":0,"first_interim_response_start":0,"next_hop_protocol":"h2","delivery_type":"","load_event_end":1.9724000000953674,"response_start":0.26160000002384187,"domain_lookup_end":0.0018000000715255738,"redirect_count":0,"transfer_size":54256,"dom_complete":1.9604000000953674,"connect_start":0.0018000000715255738,"duration":1.9724000000953674,"type":"navigate"},"resource":{"dom_content_loading_time":0,"load_event_handler_time":0,"unload_event_handler_time":0,"response_time":42.1230000027418,"encoded_size":58,"load_finished":0,"redirect_count":0,"redirect_time":0,"total_size":55,"tcp_handshake_time":0.027300000071525332,"count":62,"transferred_size":658,"dns_lookup_time":0.0028999999761583073,"resource_fetch_time":17.221899999380106,"request_time":0.0584000000953675,"min_protocol":""}]},"summary":{"navigation":{"dom_content_loading_time":0.0402999995231637,"load_event_handler_time":0.011900000095367513,"unload_event_handler_time":0,"response_time":0.02620000004768369,"encoded_size":53956,"load_finished":1.9724000000953674,"redirect_count":0,"redirect_time":0,"total_size":231423,"tcp_handshake_time":0.0125,"tls_negotiation_time":0.010899999976158143,"count":1,"transferred_size":54256,"dns_lookup_time":0,"resource_fetch_time":0.28680000007152556,"request_time":0.24689999997615816,"min_protocol":"h2"},"resource":{"dom_content_loading_time":0,"load_event_handler_time":0,"unload_event_handler_time":0,"response_time":42.1230000027418,"encoded_size":58,"load_finished":0,"redirect_count":0,"redirect_time":0,"total_size":55,"tcp_handshake_time":0.027300000071525332,"count":62,"transferred_size":658,"dns_lookup_time":0.0028999999761583073,"resource_fetch_time":17.221899999380106,"request_time":0.0584000000953675,"min_protocol":""}]},"marks":[]}]}
```

# 取得したデータを細分化

- 保存前処理 + 依存アイテム

The screenshot shows the Zabbix item configuration page for '保存前処理 2'. The main configuration area includes:

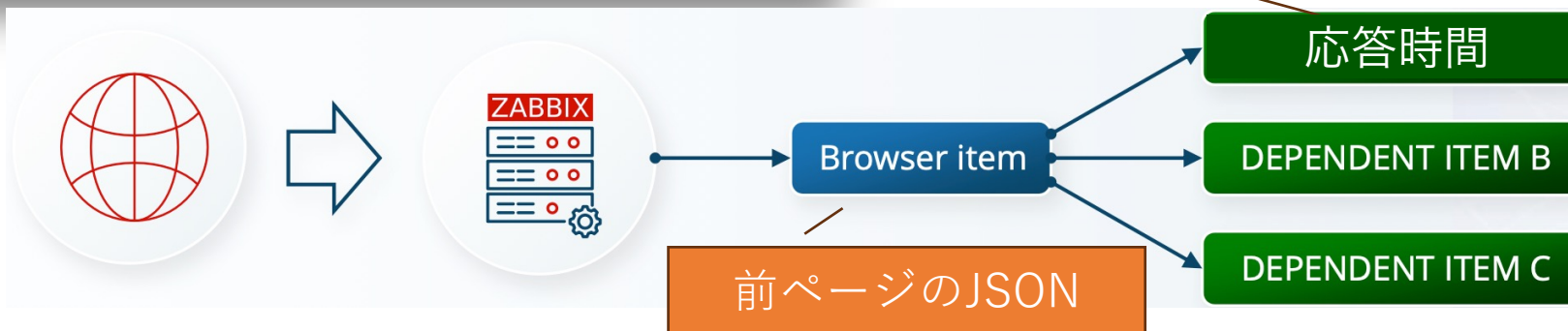
- 名前: Website zabbix.com Navigation response time
- タイプ: 依存アイテム
- キー: website.navigation.response\_time
- データ型: 数値 (浮動小数)
- マスターアイテム: ZabbixConference: Zabbix website
- 単位: s

The '保存前処理の設定' (Pre-save processing settings) section is expanded, showing two rules:

- Rule 1: JSONPath with parameter '\$.performance\_data.summary.navigation.response\_time' and a multiplier of 1.0. It has a checkbox for '失敗時のカスタマイズ' (Customize on failure) and buttons for '値を破棄' (Discard value), '値を設定' (Set value), and 'エラーを設定' (Set error).
- Rule 2: 乗数 (Multiplier) with parameter '0.001'.

The 'データ型' (Data type) is set to '数値 (浮動小数)' (Numeric (Floating point)).

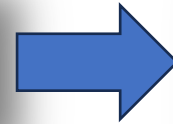
`$.performance_data.summary.navigation.response_time`



# (スクリプトを少しずつ改修)データを加工

- ブラウザアイテム設定のテンプレート化 → 設定の汎用性
  - ✓ パラメータを使用
  - ✓ パラメータの値にユーザーマクロを使用
- ※ Selenium WebDriver側とブラウザは合わせる

パラメータ	名前	値	アクション
	browser	{\$WEBSITE.BROWSER}	削除
	domain	{\$WEBSITE.DOMAIN}	削除



ホスト

ホスト IPMI タグ マクロ 3 インベントリ 暗号化 値のマッピング

ホストマクロ 継承したマクロとホストマクロ

マクロ	値
{\$WEBSITE.BROWSER}	chrome

```
switch (params.browser) {  
  case 'firefox':  
    browser = new Browser.firefoxOptions();  
  case 'safari':  
    browser = new Browser.safariOptions();  
  case 'edge':  
    browser = new Browser.edgeOptions();  
  default:  
    browser = new Browser.chromeOptions();  
}
```

# スクリーンショットを取得したい場合

- スクリプトにgetScreenshotメソッドを追加する

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

try {
  browser.navigate("https://www.zabbix.com/jp/");
  browser.collectPerfEntries();
  const screenshot = '';
  screenshot = browser.getScreenshot();
  const result = browser.getResult();
  result.screenshot = screenshot;
}
finally {
  return JSON.stringify(result);
}
```



The image shows a window titled "JavaScript" with a close button in the top right corner. The window contains the same JavaScript code as the previous block, with line numbers 1 through 12 on the left side. At the bottom of the window, there is a status bar that says "あと65242文字" and two buttons: "適用" (Apply) and "キャンセル" (Cancel).

- 続いて依存アイテム設定



# スクリーンショットを取得したい場合

- 依存アイテム設定
  - ✓データ型を「バイナリ」にする
  - ✓保存前処理に「\$.screenshot」
- 保存されるデータベースのテーブル
  - ✓新しく追加された「history\_bin」

The screenshot shows the Zabbix Item configuration interface. The 'Item' tab is active, and the 'Pre-save processing' section is expanded. The configuration includes:

- Item name: Website zabbix.com Screenshot
- Type: Dependent item
- Key: website.screenshot
- Data type: Binary
- Master item: ZabbixConference: Zabbix website
- History: Do not store (selected), Store up to 31d

Below the configuration, there is a table for pre-save processing settings:

保存前処理の設定	名前	パラメータ	失効
1:	JSONPath	\$.screenshot	<input checked="" type="checkbox"/>

# スクリーンショットのミセカタ

- ダッシュボード「アイテムのヒストリ」
- × 最新データ「ヒストリ」

- アイテムのヒストリで列に「サムネイルを表示」する事で表示

列の変更




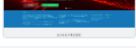





名前:

アイテム:  選択

基本色:

サムネイルを表示:

更新 キャンセル

名前	値
スクリーンショット	
スクリーンショット	
スクリーンショット	
スクリーンショット	
スクリーンショット	
スクリーンショット	
スクリーンショット	
スクリーンショット	
スクリーンショット	

ダッシュボード

タイムスタンプ	値
2024/11/15 10:15:19	binary value
2024/11/15 10:12:18	binary value
2024/11/15 10:09:18	binary value
2024/11/15 10:06:17	binary value
2024/11/15 10:03:18	binary value
2024/11/15 10:00:19	binary value
2024/11/15 09:57:18	binary value
2024/11/15 09:54:18	binary value
2024/11/15 09:51:21	binary value
2024/11/15 09:48:22	binary value

最新データ

# 注意点など

# スクリーンショットのデータ 「すぐ溜まります」

- デフォルトで1920 x 1080のpng画像がバイナリデータとして保存される
  - ✓ 画像データなので、**思っている以上にデータ量が増える**
    - 検証ではスクリーンショット4枚分で30MB
- スクリーンショットの保存方法を検討
  - ✓ ブラウザアイテムの処理で、スクリプト例外処理時のみスクリーンショットを取得
  - ✓ ヒストリの保存期間を短くする
- マスターアイテムはヒストリの保存をしない(Do not store)設定にし、すぐにデータを破棄する

# その他ブラウザアイテムのメソッド

- 要素の検索やCookieを使用したステップ実行をブラウザアイテムのスクリプトで実施

内容	メソッド	
定義済みブラウザオプション	chromeOptions() firefoxOptions()	edgeOptions() safariOptions()
タイムアウトの管理	setScriptTimeout(timeout) setSessionTimeout(timeout)	setElementWaitTimeout(timeout)
情報の取得	collectPerfEntries(mark) getRawPerfEntries()	getResult() getScreenshot()
エラー、例外処理	getError() setError(message)	discardError()
ページ操作	navigate(url) getUrl() getPageSource()	getAlert() getCookies() addCookie(cookie)
要素の検索	findElement(strategy, selector)	findElements(strategy, selector)

# まずは簡単に設定を

- テンプレート「Website by Browser」を活用ください！
  - ✓ {\$WEBSITE.DOMAIN}を変更するだけで簡単監視！
  - ✓ テンプレートを複製しベースのテンプレートとして活用！

ホスト

ホスト IPMI タグ **マクロ 1** インベントリ 暗号化 値のマッピング

• ホスト名

表示名

テンプレート	名前	アクション
	Website by Browser	<a href="#">リンクを削除</a> <a href="#">リンクと保存データを削除</a>

ホスト

ホスト IPMI タグ **マクロ 1** インベントリ 暗号化 値のマッピング

ホストマクロ 継承したマクロとホストマクロ

マクロ	値	説明
{\$WEBSITE.DOMAIN}	<input type="text" value="www.zabbix.com/jp/"/>	<input type="text" value="The domain name."/> <a href="#">削除</a>

[追加](#)



# まとめ

# 「ブラウザ」でもっと大きな監視を狙いませんか

- 最もフィジカルなWeb監視
  - ✓ Selenium WebDriverを使用し、監視対象からパフォーマンスやスクリーンショットを取得
- 最もプリミティブなWeb監視
  - ✓ 使用するのは「JavaScript」。コーディングは必要だが柔軟な対応が出来る監視設定
- 最もエキサイティングなWeb監視
  - ✓ コーディングするんでしょ・・・と初めは考えてしまいましたが、いざ作り出せばキーボードから手が離れない！心躍るブラウザアイテムを是非ご利用ください！

**Thank you**

**ZABBIX '24**

**CONFERENCE**

**JAPAN**

**#ZabConfJp2024**