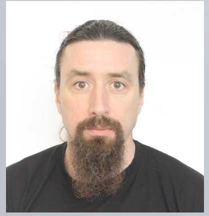
10TH
ZABBIX
SUMMIT
2022

# EXTENDING ZABBIX



**Vladimir Levijev (dimir)**
Software developer

# 01

## WHY EXTENDING ZABBIX?

# WHAT IS EXTENSIBILITY?

- An extensible system does not require changing its **source code** in order to change its behaviour.
- Using extensions is much more **convenient**.
- Allows community to **share** their work with each other.
- The trend of increasing extensibility negatively affects software **security**.

# ZABBIX EXTENSIONS

- Frontend
  - modules
  - rebranding
  - module widget (>6.4)
- Agent
  - agent 2 plugin
  - UserParameter
  - system.run
  - agent C module

- Server
  - External check
  - Zabbix trapper
  - HTTP agent
  - Webhook
  - Alert script
  - Script item
  - Zabbix API
  - server C module

# 02

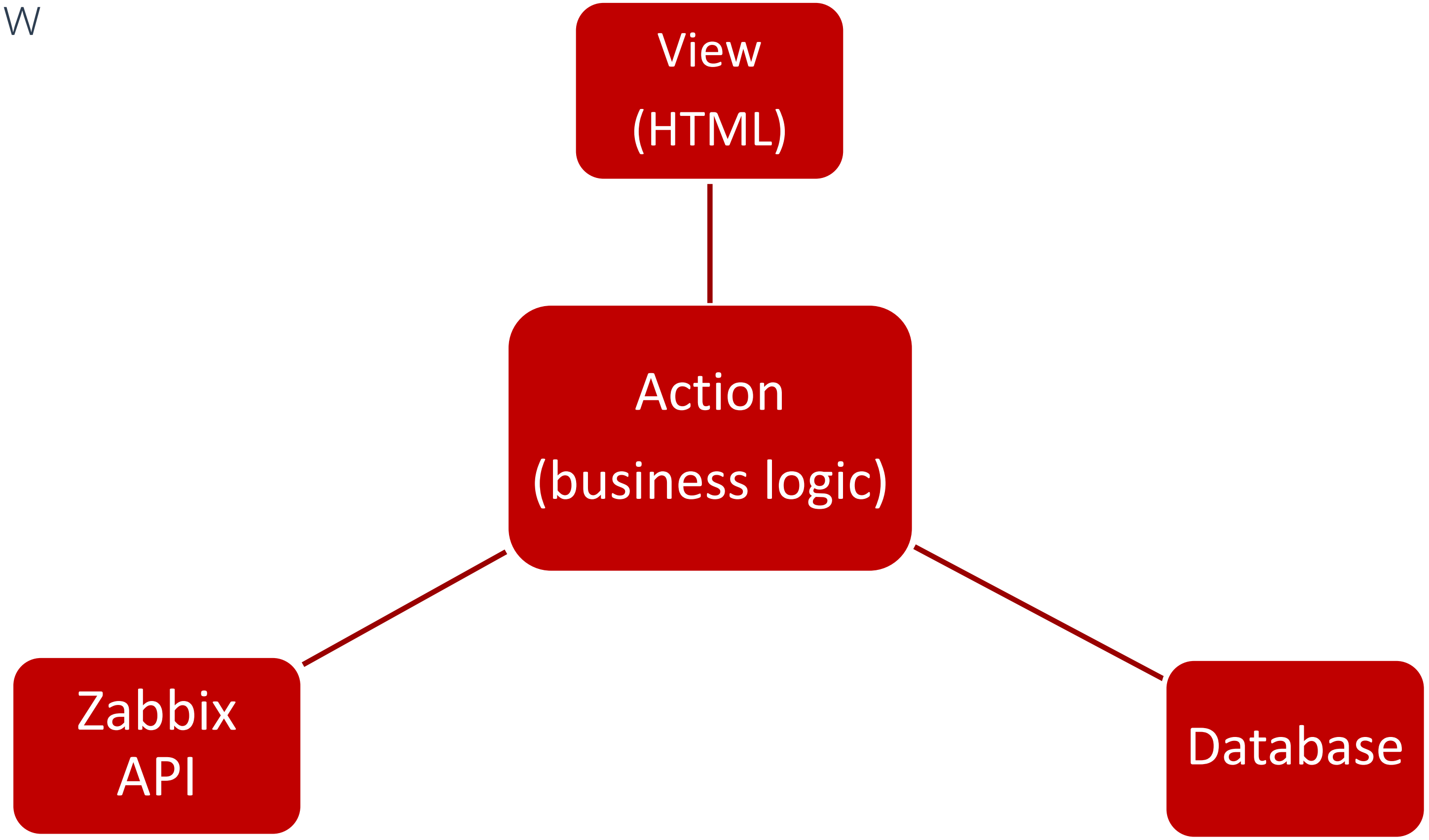
## EXTENDING FRONTEND: MODULES (SINCE 5.0)

# WHAT IS MVC

- Trygve Reenskaug created MVC while working on Smalltalk-79 in the late 1970s.
- MVC was conceived as a general solution to the problem of users controlling a large and complex data set.
- Think of 3 different files handling single web page:
  - Model – talks to a database
  - View – talks to a user
  - Controller – business logic (talks to the Model and the View)

# WHAT IS MVC IN ZABBIX

- Action = Controller + Module
- View



View (HTML)

Action (business logic)

Zabbix API

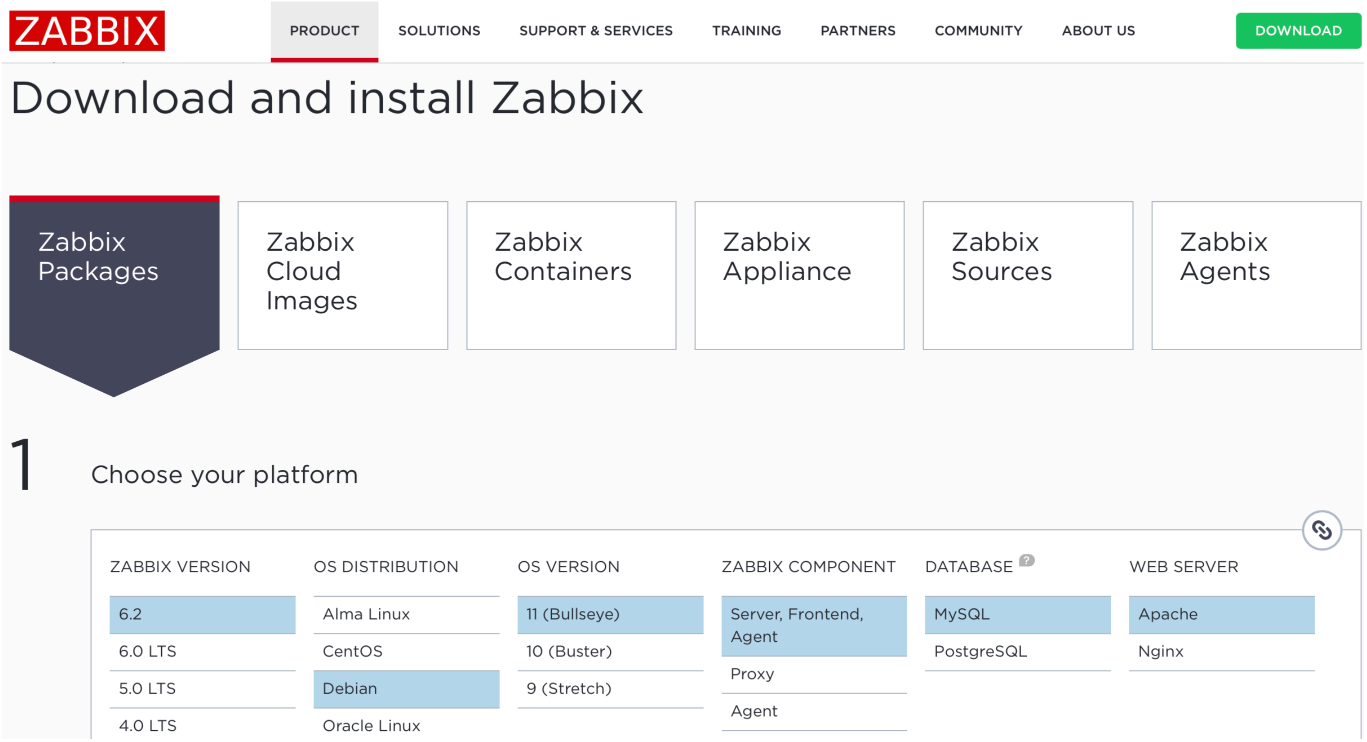
Database

# AN EXAMPLE OF A WORKFLOW IN ZABBIX

- User clicks “Configuration → Hosts”
- The URL contains: /zabbix.php?action=host.list
- This action is handled by file CControllerHostList.php, it gets the list of hosts and and prepares the data the way the view expects it
- That data is passed to the View which is handled by file configuration.host.list.php which outputs HTML

# WHAT YOU NEED TO CREATE ONE

- All you need is Zabbix frontend that is connected to working database.
- Use instructions on *https://www.zabbix.com/download* in order to install it.



## Download and install Zabbix

Zabbix Packages

Zabbix Cloud Images

Zabbix Containers

Zabbix Appliance

Zabbix Sources

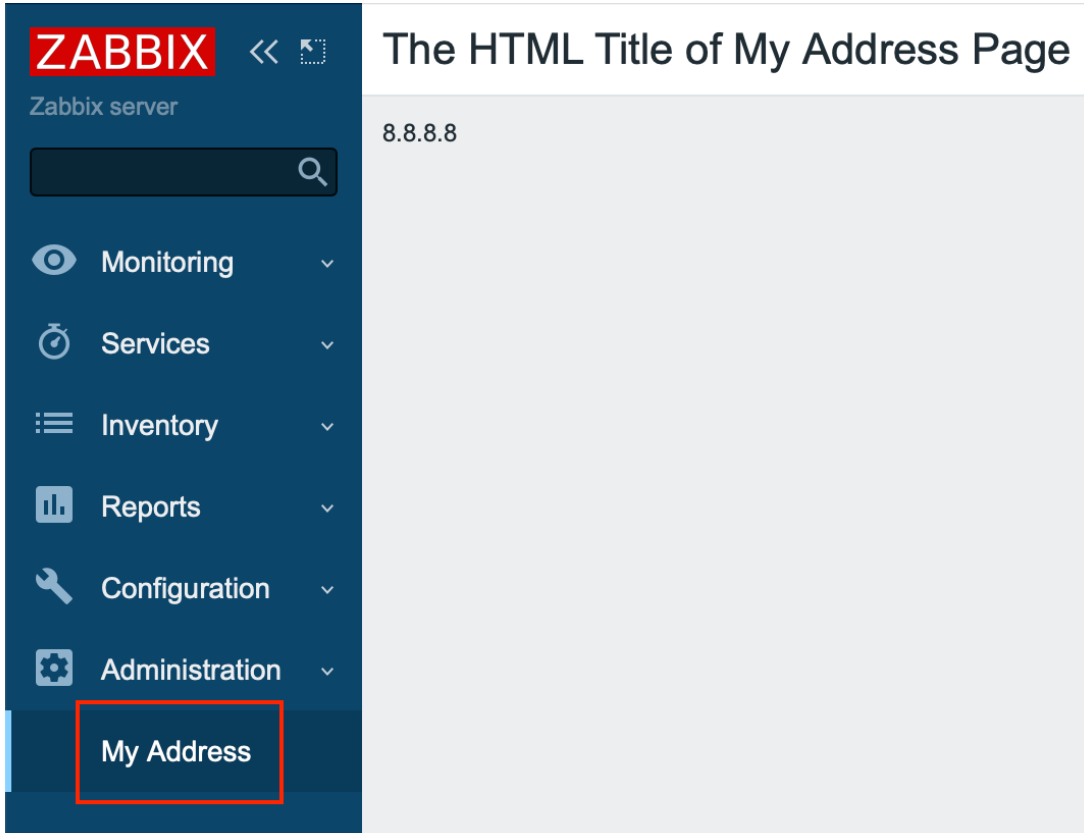
Zabbix Agents

1 Choose your platform

| ZABBIX VERSION | OS DISTRIBUTION | OS VERSION | ZABBIX COMPONENT | DATABASE | WEB SERVER |
|---|---|---|---|---|---|
| 6.2 | Alma Linux | 11 (Bullseye) | Server, Frontend, Agent | MySQL | Apache |
| 6.0 LTS | CentOS | 10 (Buster) | Proxy | PostgreSQL | Nginx |
| 5.0 LTS | Debian | 9 (Stretch) | Agent | | |
| 4.0 LTS | Oracle Linux | | | | |

# MODULE "MY ADDRESS"

- Add a menu entry called "My Address"
- When that is clicked, make a HTTP request to https://api.ipify.org
- Display external IP address



The HTML Title of My Address Page

8.8.8.8

# IT’S EASY!

*http://bash.org/?464385*

<@insomnia> it **only takes three commands** to install Gentoo

<@insomnia> cfdisk /dev/hda && mkfs.xfs /dev/hda1 && mount /dev/hda1 /mnt/gentoo/ && chroot /mnt/gentoo/ && env-update && . /etc/profile && emerge sync && cd /usr/portage && scripts/bootsrap.sh && emerge system && emerge vim && vi /etc/fstab && emerge gentoo-dev-sources && cd /usr/src/linux && make menuconfig && make install modules_install && emerge gnome mozilla-firefox openoffice && emerge grub && cp /boot/grub/grub.conf.sample /boot/grub/grub.conf && vi /boot/grub/grub.conf && grub && init 6

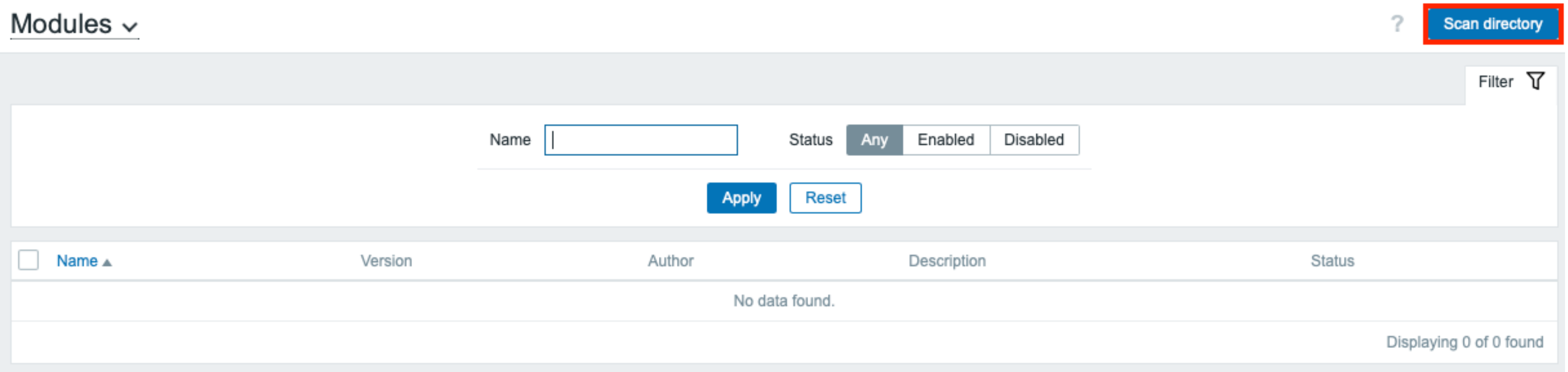
<@insomnia> **that's the first one**

# IT’S ONLY 4 FILES

- Users say: “Creating a Frontend module is too hard!” I will prove they’re wrong!
- It only takes 4 files
- 
  ```
  git clone https://github.com/dimir/zabbix-extensions.git
  ```
- 
  ```
  sudo cp -r zabbix-extensions/frontend/modules/my-address
  /usr/share/zabbix/modules
  ```
- 
  ```
  cd !$/my-address
  ```
- 
  ```
  find -type f
      manifest.json
      Module.php
      actions/MyAddress.php
      views/my.address.php
  ```

# DISCOVER THE MODULE

- Administration → General → Modules → Scan directory



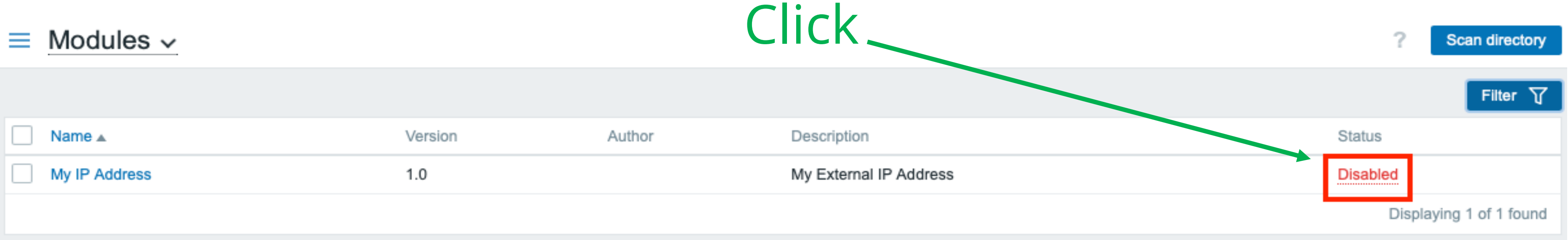
Name: |

Status: [✓ Any] Enabled Disabled

| □ | Name ▲ | Version | Author | Description | Status |
|---|---|---|---|---|---|
| No data found. | | | | | |

Displaying 0 of 0 found

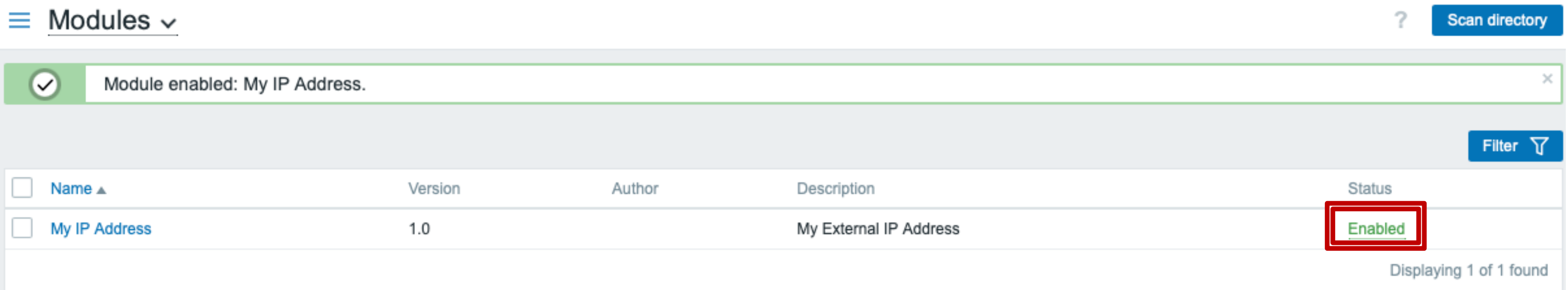
Click



| □ | Name ▲ | Version | Author | Description | Status |
|---|---|---|---|---|---|
| □ | My IP Address | 1.0 | | My External IP Address | Disabled |

Displaying 1 of 1 found

# DISCOVER THE MODULE



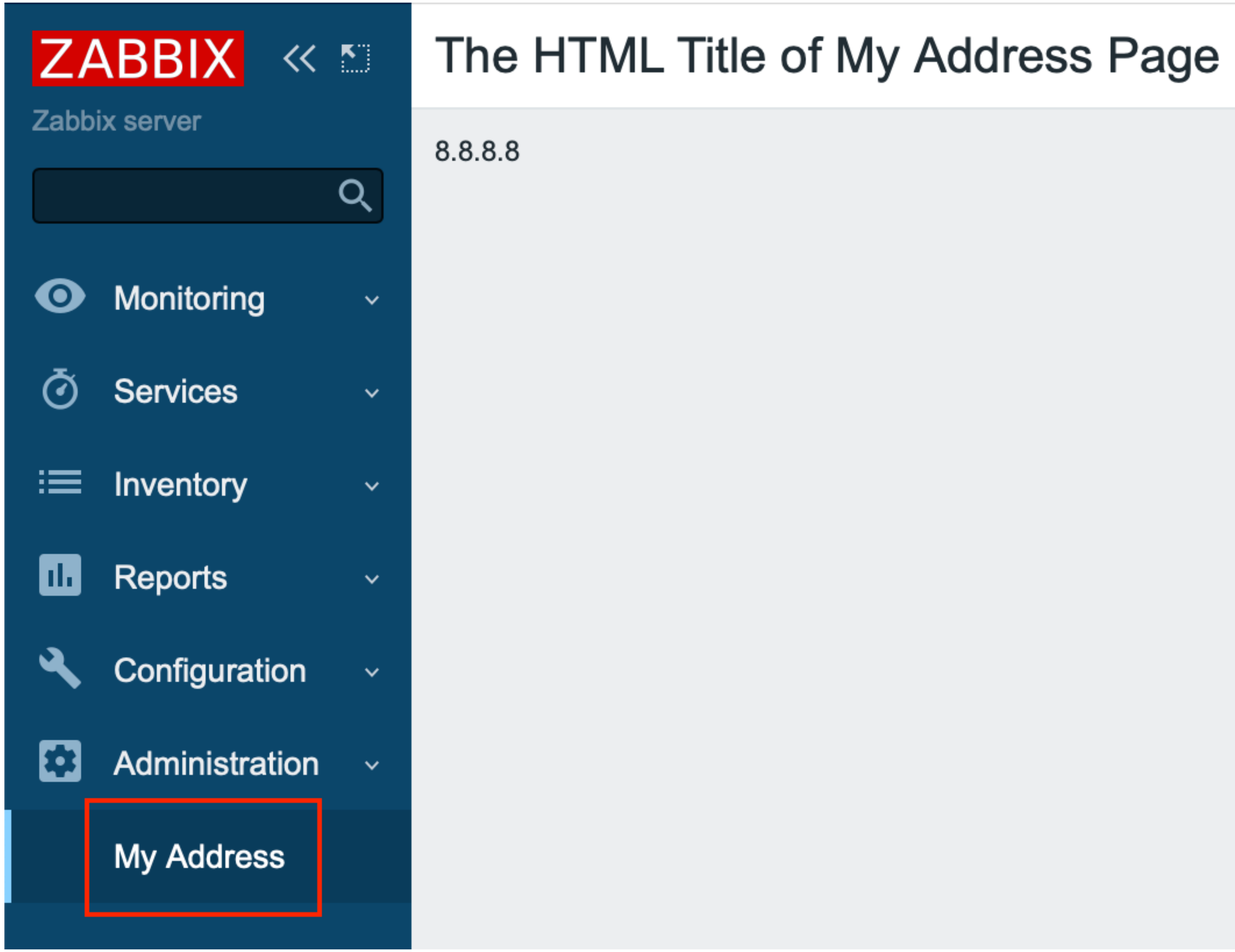
Module enabled: My IP Address.

| Name | Version | Author | Description | Status |
|---|---|---|---|---|
| My IP Address | 1.0 |  | My External IP Address | Enabled |

Displaying 1 of 1 found

```
SQL> select * from module;

+----------+------------+----------------+--------+--------+
| moduleid | id         | relative_path  | status | config |
+----------+----------------+------------+--------+--------+
|        1 | my-address | my-address     |      1 | []     |
+----------+------------+----------------+--------+--------+
```

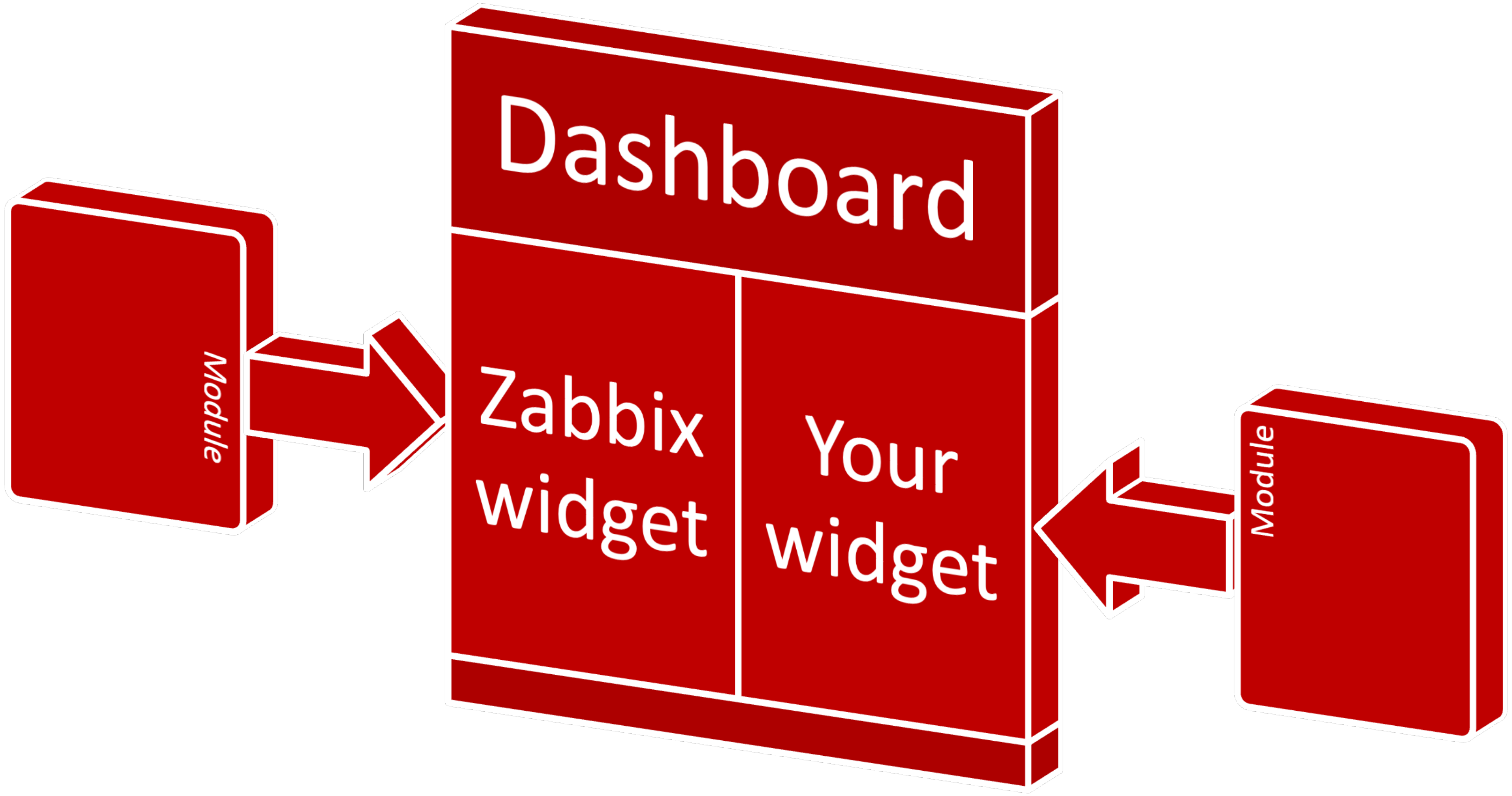
# YOU ARE DONE!



The HTML Title of My Address Page

8.8.8.8

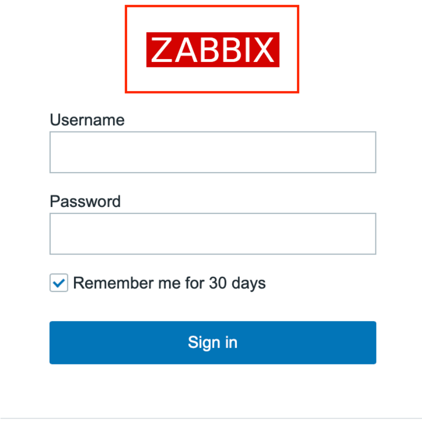
# 6.4: DASHBOARD WIDGET MODULE



# 03

## EXTENDING FRONTEND: REBRANDING (SINCE 5.0)

# WHAT EXACTLY IS REBRANDED



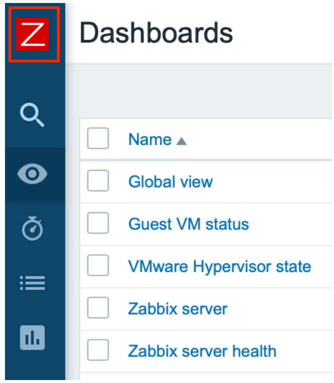
ZABBIX

Username

Password

☑ Remember me for 30 days

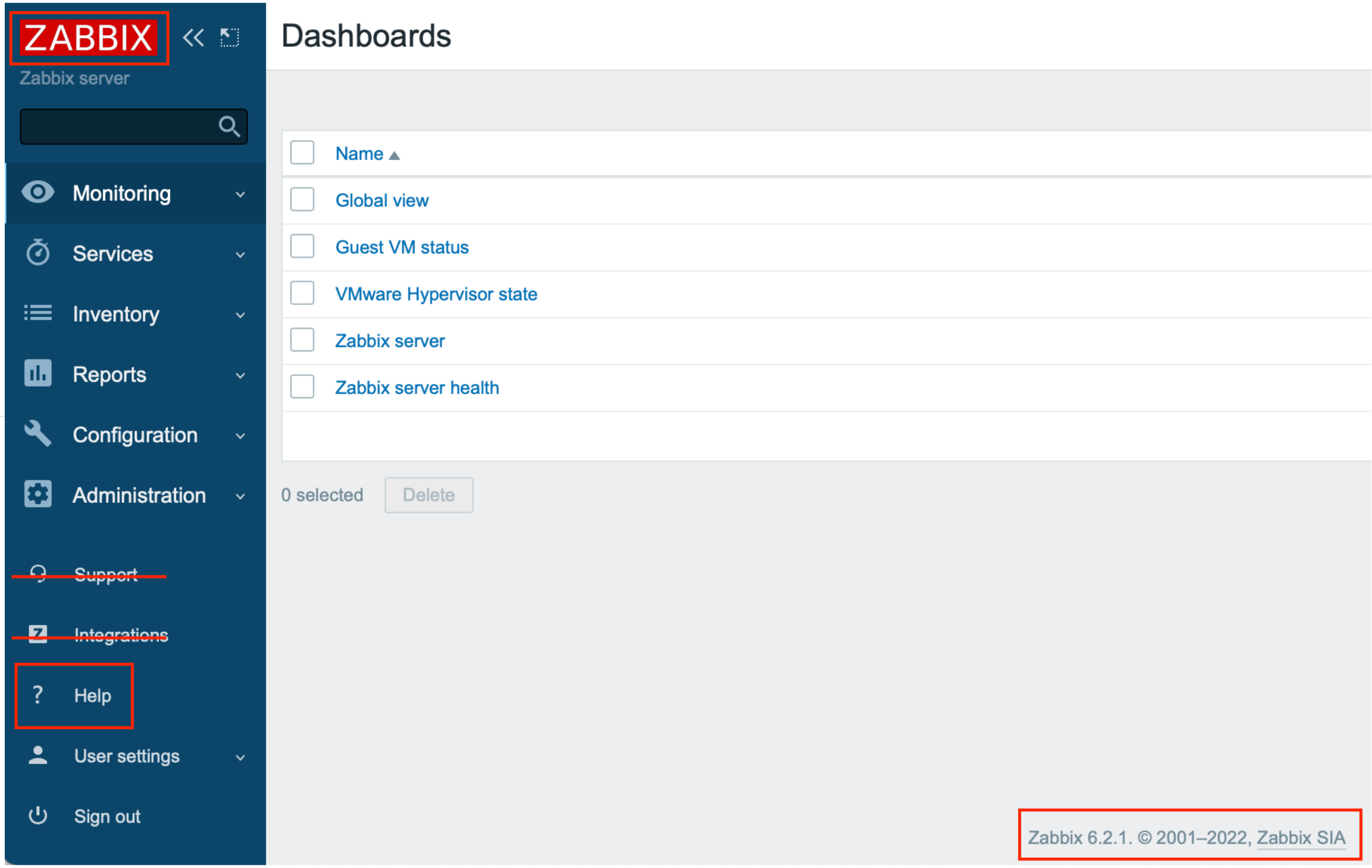
Sign in



Z

Dashboards

| □ | Name ▲ |
|---|---|
| □ | Global view |
| □ | Guest VM status |
| □ | VMware Hypervisor state |
| □ | Zabbix server |
| □ | Zabbix server health |



Dashboards

| □ | Name ▲ |
|---|---|
| □ | Global view |
| □ | Guest VM status |
| □ | VMware Hypervisor state |
| □ | Zabbix server |
| □ | Zabbix server health |

0 selected Delete

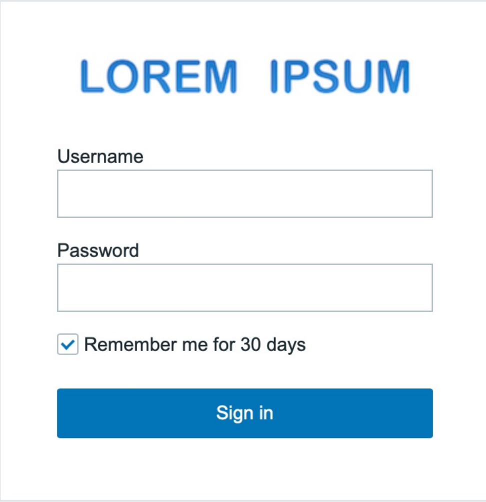
Zabbix 6.2.1. © 2001–2022, Zabbix SIA

# HOW?

```
$ mkdir -p /usr/share/zabbix/local/conf
$ vi brand.conf.php
```

```
<?php
# removes the "support" and "integration" menu elements
# return [];

# in addition rebrands 5 elements
return [
    'BRAND_LOGO'                 => 'https://[…]/lorem-ipsum.png',
    'BRAND_LOGO_SIDEBAR'         => 'https://[…]/lorem.png',
    'BRAND_LOGO_SIDEBAR_COMPACT' => 'https://[…]/l.png',
    'BRAND_FOOTER'               => '(c) Lorem Ipsum',
    'BRAND_HELP_URL'             => 'https://www.example.com/help/'
];
```

# DONE!

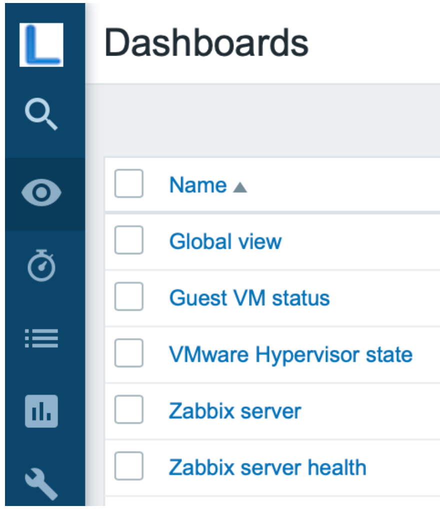


LOREM IPSUM

Username

Password

☑ Remember me for 30 days



## Dashboards

□ Name ▲

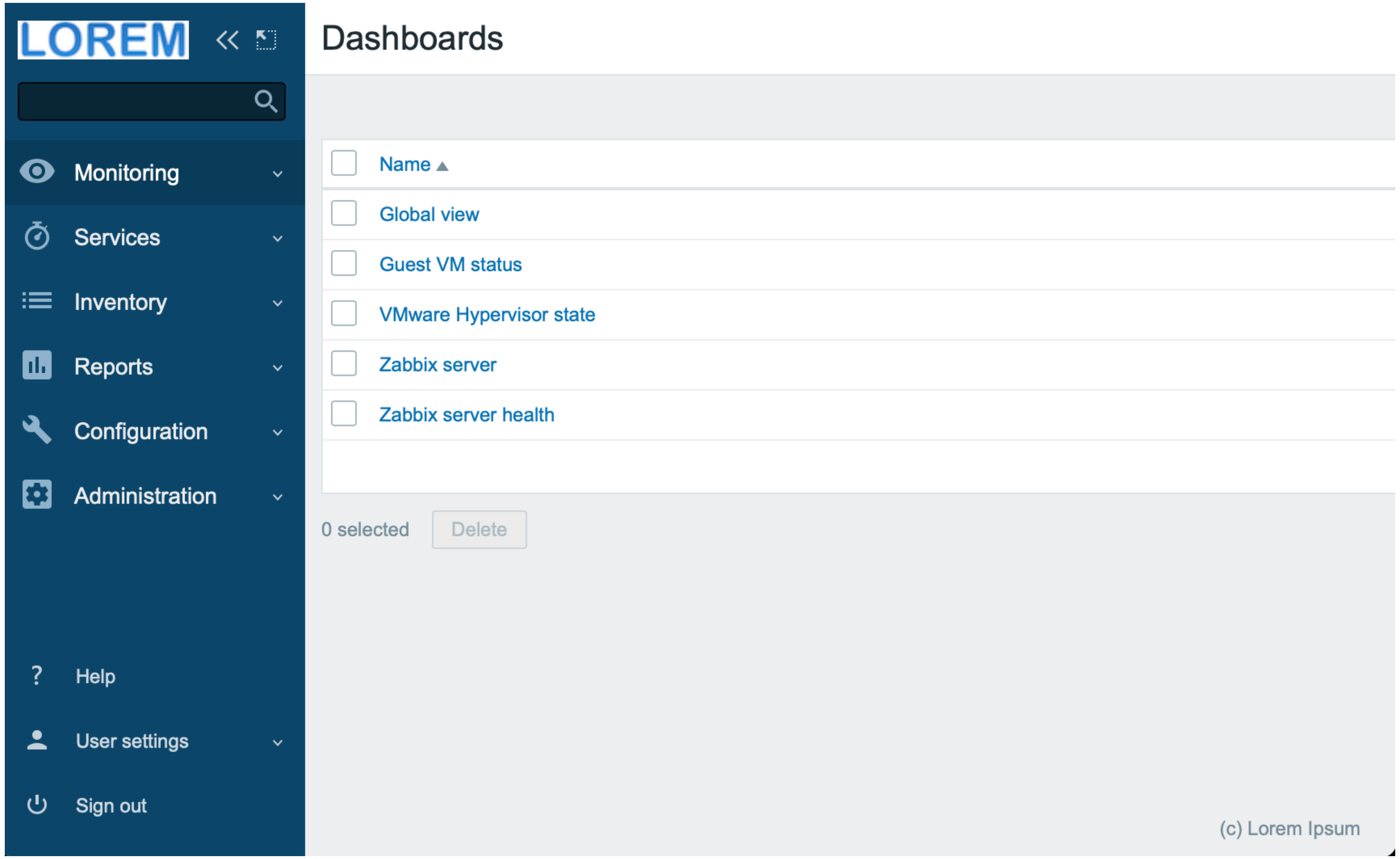
□ Global view

□ Guest VM status

□ VMware Hypervisor state

□ Zabbix server

□ Zabbix server health



## Dashboards

□ Name ▲

□ Global view

□ Guest VM status

□ VMware Hypervisor state

□ Zabbix server

□ Zabbix server health

0 selected

(c) Lorem Ipsum

# 04

## EXTENDING AGENT

# AGENT 2 PLUGINS

- Built-in
  - Since Zabbix 4.4
  - Agent 2 sources are required
  - Agent 2 needs to be compiled together with plugin
  - Agent 2 recompilation is required on upgrade

- Loadable
  - Since Zabbix 6.0
  - Agent 2 sources are not needed
  - Agent 2 compilation is not needed
  - Agent 2 recompilation is not needed on upgrade

# AGENT 2 LOADABLE PLUGIN

```
$ git clone https://github.com/dimir/zabbix-extensions.git
$ cd zabbix-extensions/agent2/plugins/myip
$ less main.go
```

```
[…]
import (
    […]
    "git.zabbix.com/ap/plugin-support/plugin/container"
    "git.zabbix.com/ap/plugin-support/plugin”
)
[…]
func (p *Plugin) Export(…) (…) {
    resp, err := http.Get("https://api.ipify.org")
    defer resp.Body.Close()
    body, err := ioutil.ReadAll(resp.Body)
    return string(body), nil
}
func init() {
    plugin.RegisterMetrics(&impl, "Myip", "myip", "Return the external IP address.")
}
func main() {
    […]
}
```

# INSTALL AGENT 2 PLUGIN

- You need to be connected to the Internet.
- `$ less README`
- `$ sudo make install`

- Tell Zabbix agent 2 where it is.
- `$ echo Plugins.Myip.System.Path=/usr/local/zabbix/go/plugins/myip | sudo tee /etc/zabbix/zabbix_agent2.d/plugins.d/myip.conf`

- Test it

```
root@831ccaa3209c:~/git/zabbix-extensions/agent2/plugins/myip#
```

# UserParameter

- Supported in both agents.

-
  ```
  echo UserParameter=myip,curl -s https://api.ipify.org | sudo tee /etc/zabbix/zabbix_agent2.d/myip.conf
  ```

-
  ```
  zabbix_agent2 -t myip
  myip                                                       [s|8.8.8.8]
  ```

- Or `system.run`

-
  ```
  echo 'AllowKey=system.run[*]' | sudo tee /etc/zabbix/zabbix_agent2.d/system.run.conf
  ```

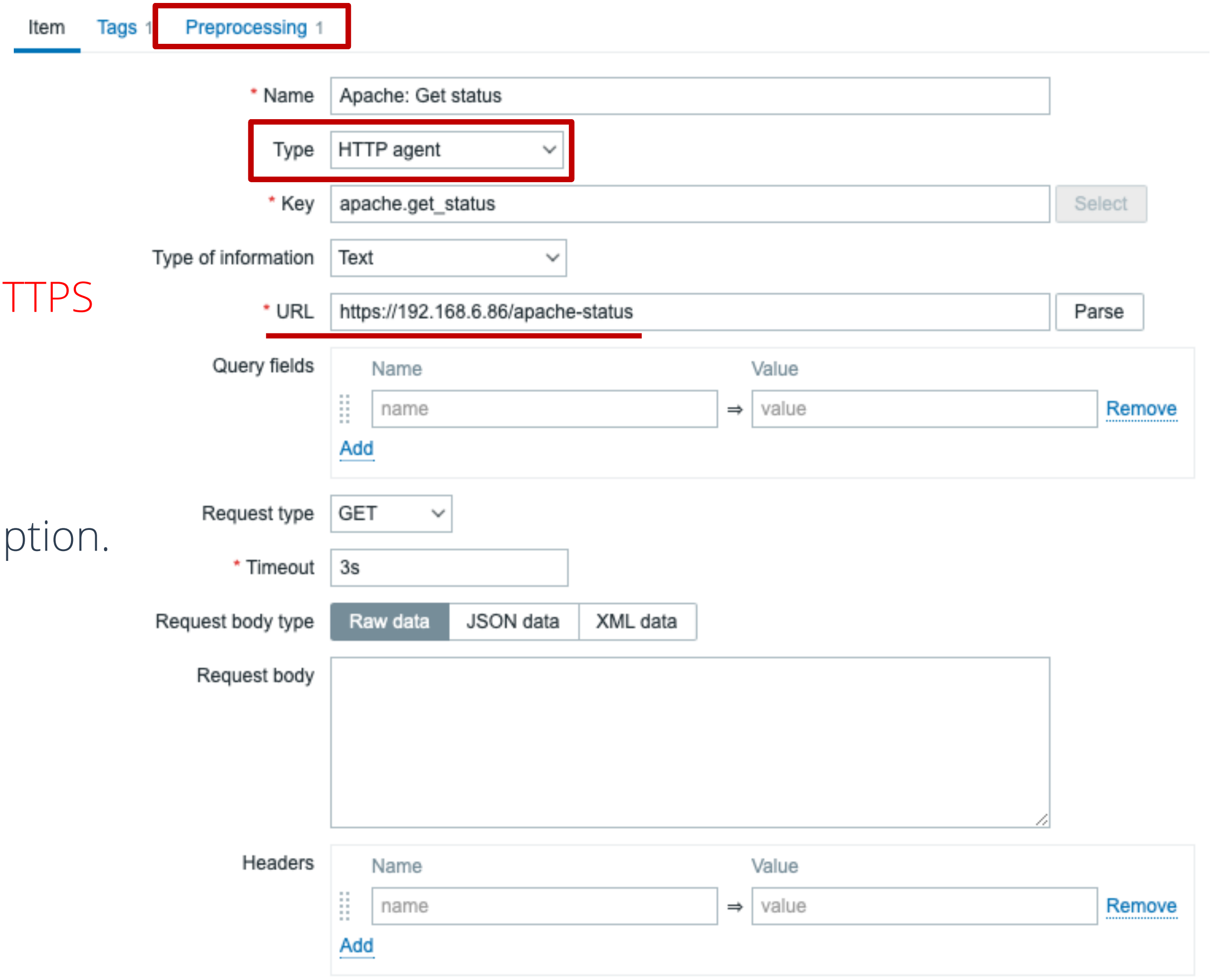
-
  ```
  zabbix_agent2 -t 'system.run[curl -s https://api.ipify.org]'
  system.run[curl -s https://api.ipify.org]                  [s|8.8.8.8]
  ```

# 05

## EXTENDING SERVER

# HTTP AGENT ITEM (SINCE 4.0)

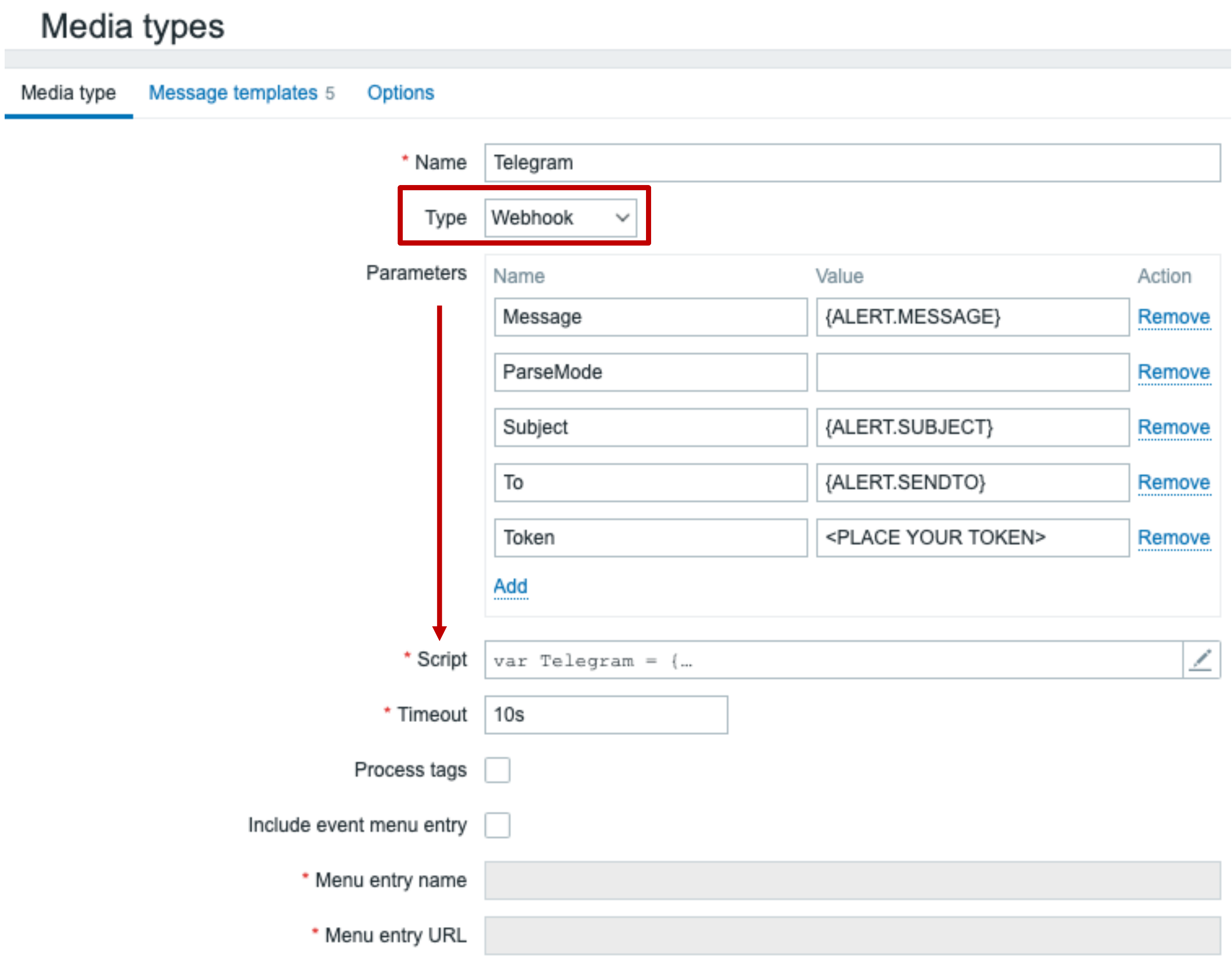
- Collecting data using HTTP/HTTPS protocol.
- Zabbix server/proxy must be compiled with --with-libcurl option.
- Zabbix agent is not needed.



|  |  |
|---|---|
| * Name | Apache: Get status |
| Type | HTTP agent |
| * Key | apache.get_status |
| Type of information | Text |
| * URL | https://192.168.6.86/apache-status |
| Query fields | Name: name ⇒ Value: value |
| Request type | GET |
| * Timeout | 3s |
| Request body type | Raw data / JSON data / XML data |
| Request body |  |
| Headers | Name: name ⇒ Value: value |

# WEBHOOK MEDIA TYPE (SINCE 4.4)

- Media type for making HTTP calls using custom JavaScript code for integration with external software such as helpdesk systems, chats or messengers.



## Media types

|  |  |
|---|---|
| * Name | Telegram |
| Type | Webhook |

Parameters

| Name | Value | Action |
|---|---|---|
| Message | {ALERT.MESSAGE} | Remove |
| ParseMode |  | Remove |
| Subject | {ALERT.SUBJECT} | Remove |
| To | {ALERT.SENDTO} | Remove |
| Token | <PLACE YOUR TOKEN> | Remove |

|  |  |
|---|---|
| * Script | `var Telegram = {…` |
| * Timeout | 10s |
| Process tags | □ |
| Include event menu entry | □ |
| * Menu entry name |  |
| * Menu entry URL |  |

# OFFICIAL WEBHOOKS

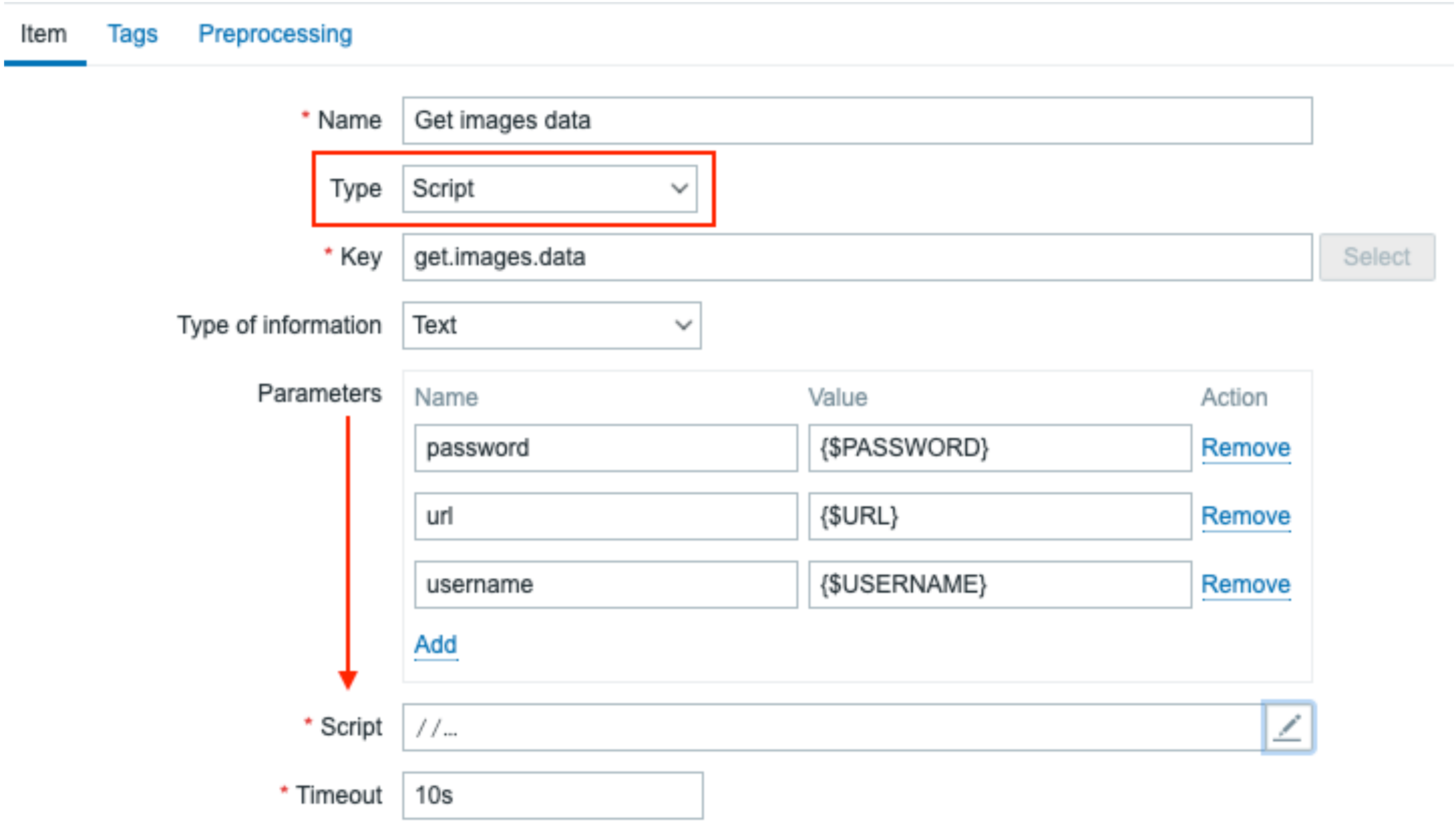
- brevis.one
- Discord
- Express.ms messenger
- Github issues
- GLPi
- iLert
- iTop
- Jira
- Jira Service Desk
- ManageEngine ServiceDesk
- Mattermost
- Microsoft Teams
- Opsgenie
- OTRS
- Pagerduty
- Pushover
- Redmine
- Rocket.Chat
- ServiceNow
- SIGNL4
- Slack
- SolarWinds
- SysAid
- Telegram
- TOPdesk
- VictorOps
- Zammad
- Zendesk

# SCRIPT ITEM (SINCE 5.2)

- Same as HTTP agent item for more complex communications.
- Script item = JavaScript item
- Also uses HttpRequest JavaScript class.
- Similar to Webhook but for data collection.

```
var request = new HttpRequest();

var url = params.url + '/users/login';

request.addHeader('Content-Type: application/json’);

const payload = {"username":params.username,"password":params.password};

response = request.post(url, JSON.stringify(payload));

if (request.getStatus() !== 200) {
        throw 'Request failed with status code ' + request.getStatus() + ': ' + response;
}

if (response !== null) {
    try {
        response = JSON.parse(response);
    }
    catch (error) {
        throw 'Failed to parse response received from Docker Hub API. Check debug log for more
information.';
    }
}

const token = response.token;
```

# SCRIPT ITEM



|  |  |
|---|---|
| * Name | Get images data |
| Type | Script |
| * Key | get.images.data |
| Type of information | Text |

Parameters

| Name | Value | Action |
|---|---|---|
| password | {$PASSWORD} | Remove |
| url | {$URL} | Remove |
| username | {$USERNAME} | Remove |

|  |  |
|---|---|
| * Script | //... |
| * Timeout | 10s |

# EXTERNAL CHECK

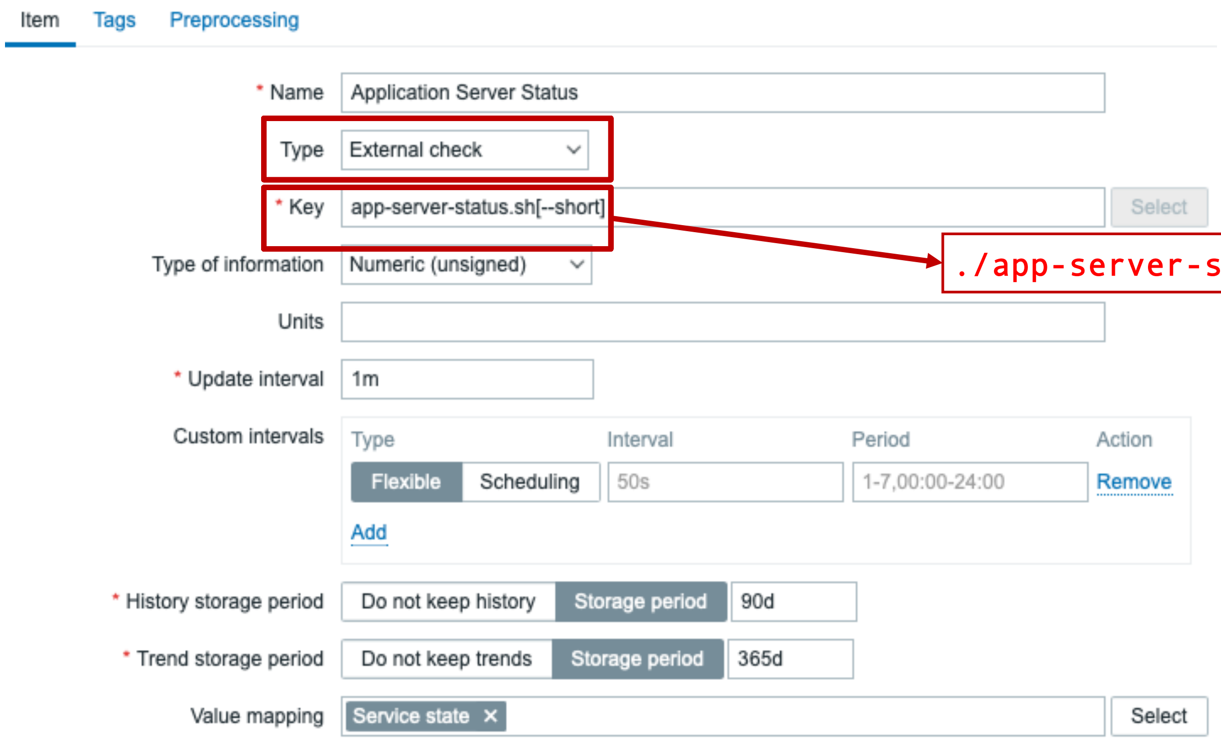
- *External check* is a check executed by Zabbix server or proxy by running an executable, for example a shell script.

- `grep ExternalScripts= /etc/zabbix/zabbix_server.conf`

  # ExternalScripts=/usr/lib/zabbix/externalscripts

- `ls /usr/lib/zabbix/externalscripts`

  app-server-status.sh



| | |
|---|---|
| * Name | Application Server Status |
| Type | External check |
| * Key | app-server-status.sh[--short] |
| Type of information | Numeric (unsigned) |
| Units | |
| * Update interval | 1m |
| Custom intervals | Type: Flexible / Scheduling; Interval: 50s; Period: 1-7,00:00-24:00; Action: Remove |
| * History storage period | Do not keep history / Storage period: 90d |
| * Trend storage period | Do not keep trends / Storage period: 365d |
| Value mapping | Service state ✕ |

```
./app-server-status.sh --short
```

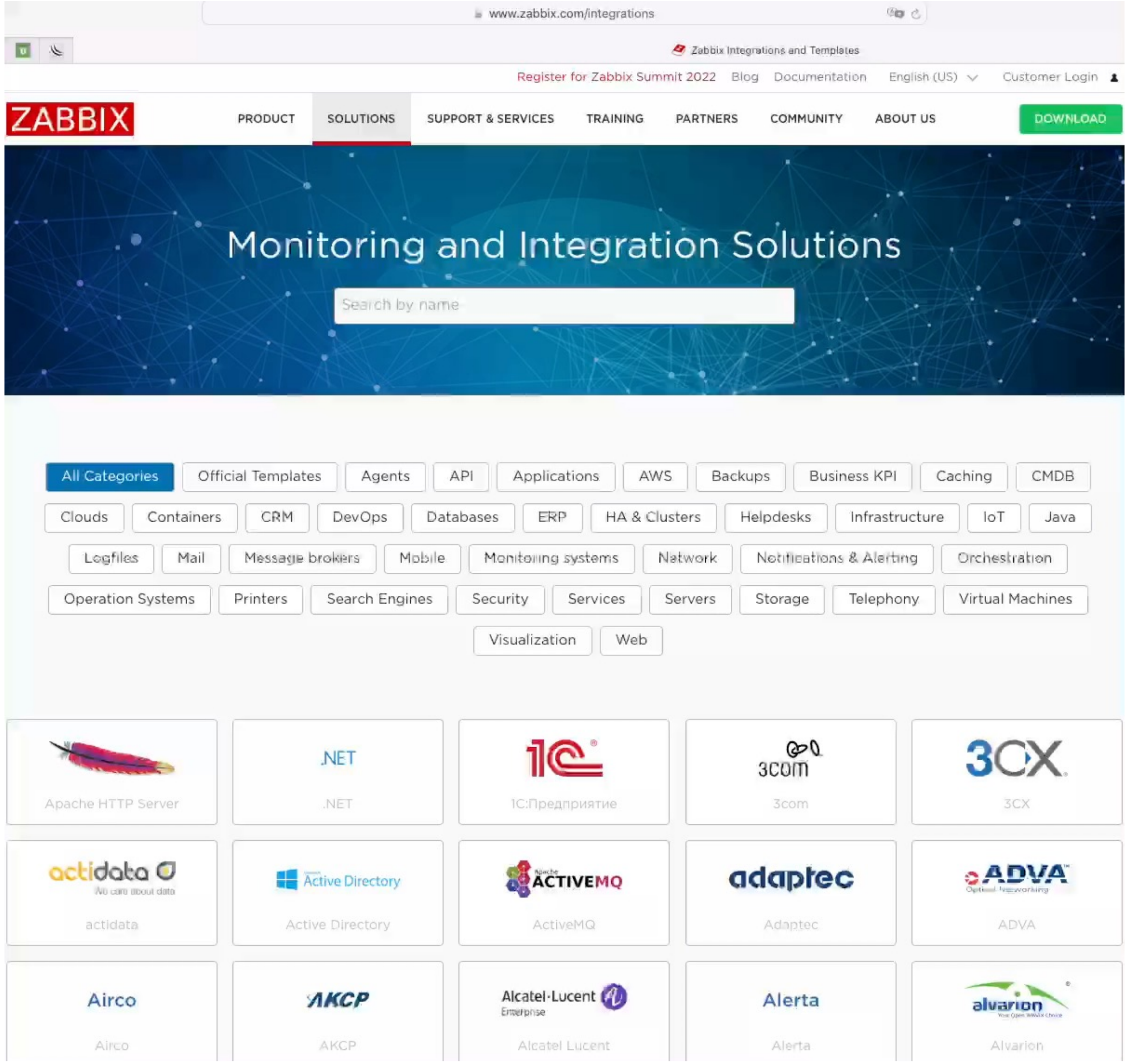
# 06

## ZABBIX API FOR INTEGRATION

# ZABBIX API

- Allows you to
  - **Retrieve** Zabbix configuration (hosts, items, triggers etc.)
  - **Modify** Zabbix configuration
  - **Get** history data
  - Future: push metrics
- Implementation:
  - **JSON-RPC** over HTTP
  - **Not** restful
  - Client must send **POST**, even when retrieving data because the body is required
  - Request and response bodies are **JSON**
  - Is shipped with the Frontend
  - Set of methods e. g. *host.get, host.create, host.update, host.delete*
  - Frontend is **using** it

# OTHER INTEGRATIONS



Monitoring and Integration Solutions

# ALL THE CODE IS ON GIT HUB

- Do not copy the code from this presentation.
- Instead go to https://github.com/dimir/zabbix-extensions

10TH ZABBIX SUMMIT 2022

# THANK YOU! QUESTIONS?