



## 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**.



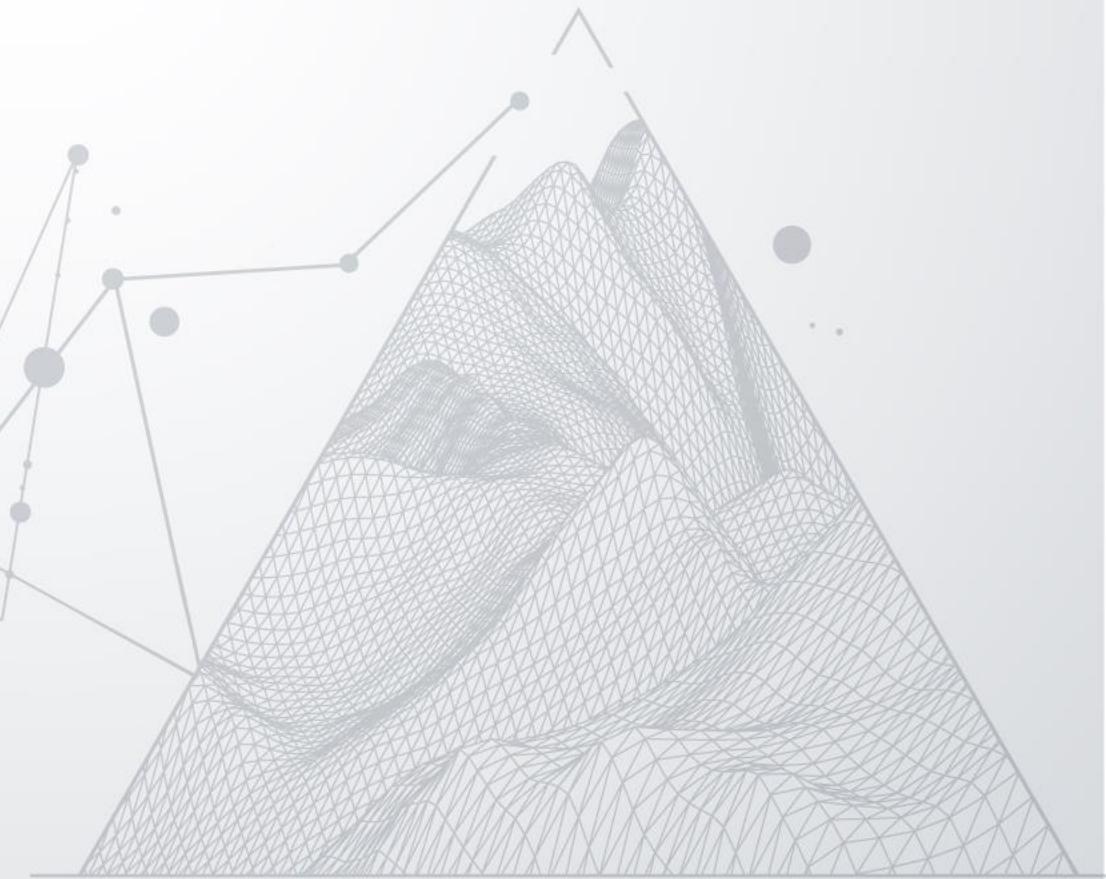
# ZABBIX EXTENSIONS

- ✓ 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)



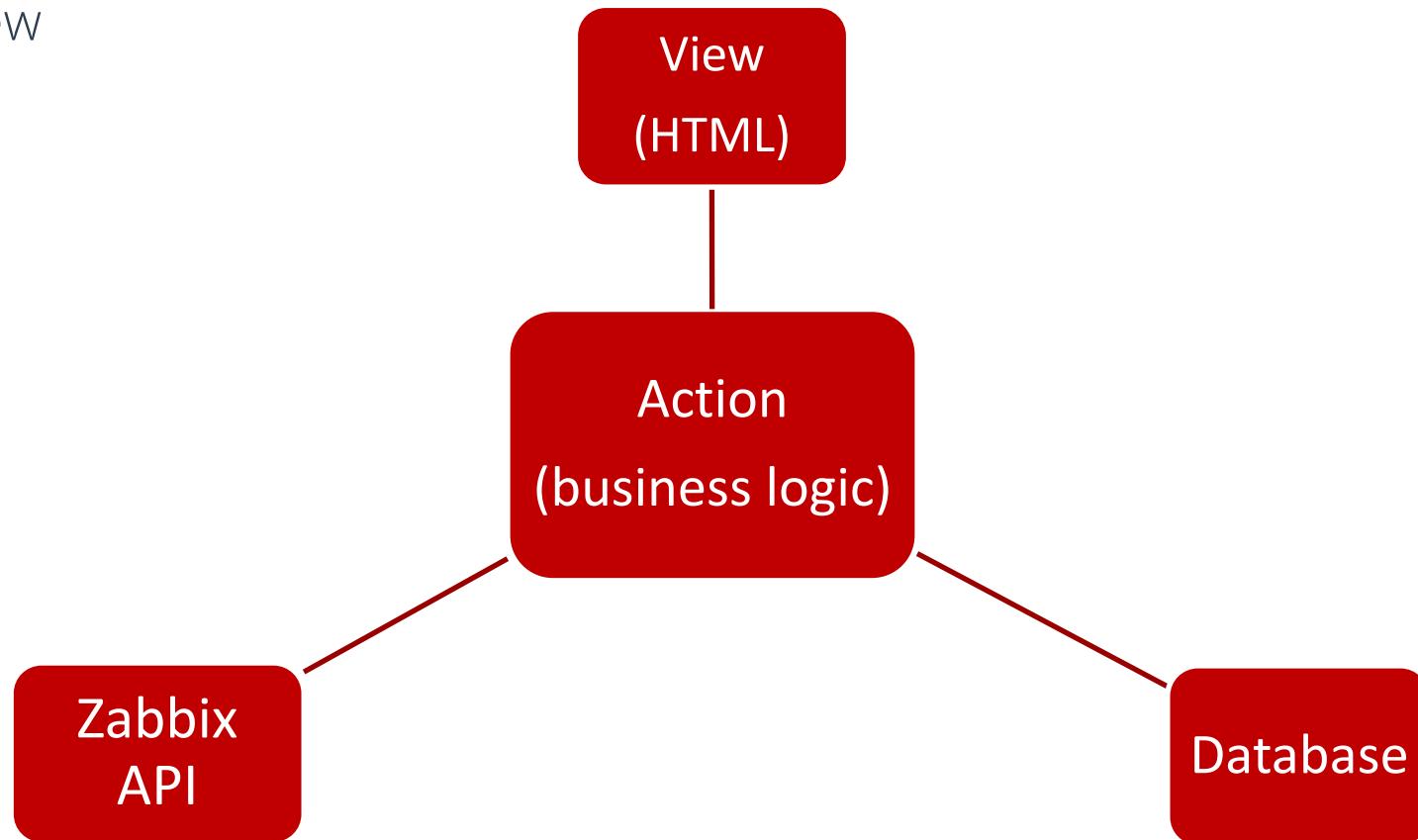
# WHAT IS MVC

- ⓘ 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



# 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 prepares the `data` the way the view expects it



The `$data` variable is automatically available in 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.

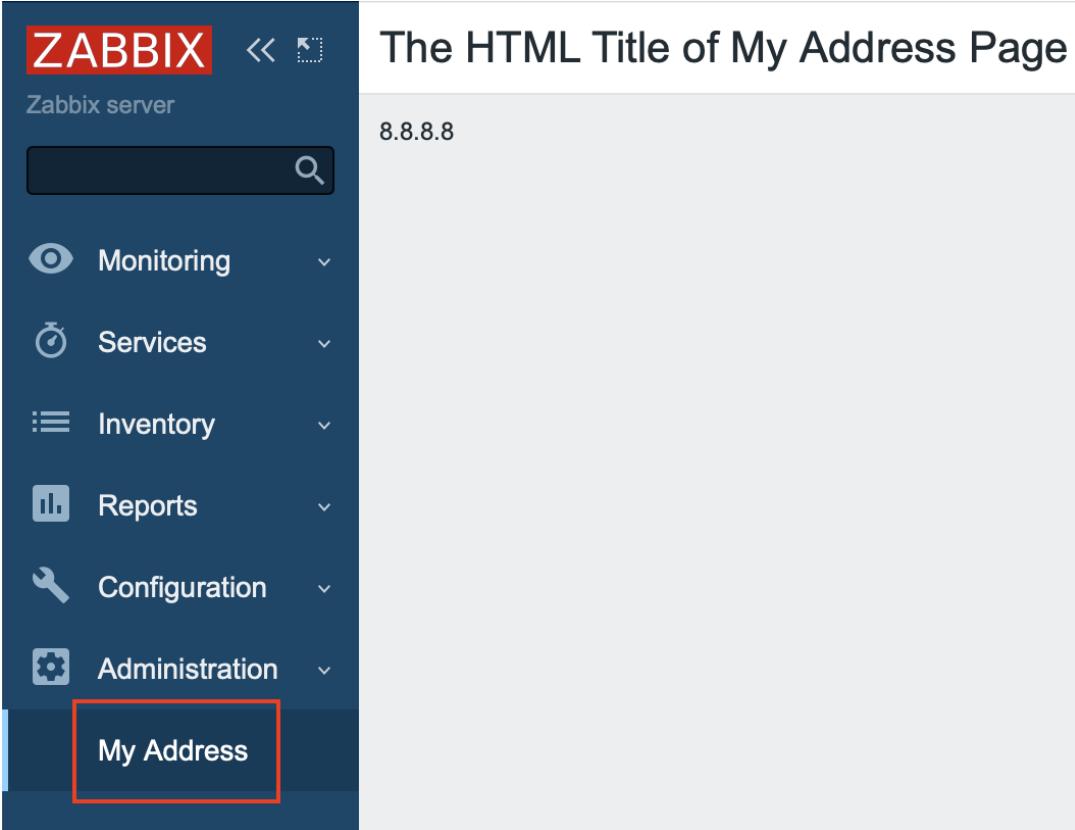
The screenshot shows the Zabbix download page. At the top, there's a navigation bar with links: PRODUCT (highlighted in red), SOLUTIONS, SUPPORT & SERVICES, TRAINING, PARTNERS, COMMUNITY, ABOUT US, and DOWNLOAD. Below the navigation, the title "Download and install Zabbix" is displayed. A large red button labeled "ZABBIX" is on the left. To its right, there are six options: "Zabbix Packages" (selected), "Zabbix Cloud Images", "Zabbix Containers", "Zabbix Appliance", "Zabbix Sources", and "Zabbix Agents".  
  
Below this, a large number "1" is followed by the text "Choose your platform". A table provides specific configurations:

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				



# A SIMPLE EXAMPLE

- ✓ 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 screenshot shows the Zabbix web interface. On the left, there is a dark sidebar with the ZABBIX logo at the top. Below it are several navigation items: Monitoring, Services, Inventory, Reports, Configuration, Administration, and a newly added item, "My Address", which is highlighted with a red rectangular border. To the right of the sidebar, the main content area has a title "The HTML Title of My Address Page" and the IP address "8.8.8.8" displayed.



# IT'S EASY!

```
<@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/bootstrap.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
```

<http://bash.org/?464385>



# IT'S ONLY 4 FILES

- ① git clone https://github.com/dimir/zabbix-extensions.git
- ② sudo cp -r zabbix-extensions/frontend/modules/my-address /usr/share/zabbix/modules
- ③ cd !\$/my-address
- ④ find -type f
  - manifest.json ← declares new action “my.address”
  - Module.php ← adds new menu entry
  - actions/MyAddress.php ← gets the IP address
  - views/my.address.php ← outputs the IP address



# FILE #1: manifest.json

```
$ less manifest.json
```

```
{  
  "manifest_version": 1.0,          <-- version of the manifest, leave it to 1.0  
  "id": "my-address",            <-- unique "identifier" of the module  
  "name": "My IP Address",       <-- visible module name in Frontend  
  "version": "1.0",              <-- not used by framework, useful for upgrades  
  "namespace": "MyAddress",      <-- PHP code "namespace", must match in the code  
  "description": "My External IP Address", <-- short module description visible in Frontend  
  "actions": {                  <-- list of actions that this module implements  
    "my.address": {           <-- the action name (in URL ?action=<action>)  
      "class": "MyAddress",    <-- the action file name actions/MyAddress.php  
      "view": "my.address"    <-- the view file name views/my.address.php  
    }  
  }  
}
```



# FILE #2: ACTION FILE MyAddress.php

- ✓ An action file takes care of the business logic of the action.
- ✓ We must implement 1 PHP class called **MyAddress**.
- ✓ All the implementation is inside the function **doAction()**.

```
$ less actions/MyAddress.php
```

```
<?php
namespace Modules\MyAddress\Actions;

use CControllerResponseData;
use CController;

class MyAddress extends CController {
    public function init(): void {
        $this->disableSIDValidation();
    }
    protected function checkInput(): bool {return true;}
    protected function checkPermissions(): bool {return true;}
    protected function doAction(): void {
        $data = ['my-ip' => file_get_contents("https://api.ipify.org")];

        $response = new CControllerResponseData($data);
        $this->setResponse($response);
    }
}
```



## FILE #3: VIEW FILE my.address.php

- ✓ A View file outputs one element of `$data` array that comes from the Action file.
- ✓ In our case it's a text in an HTML `<div>` element.
- ✓ This view uses Zabbix PHP classes to produce the output but this is not a requirement.
- ✓ No requirements of what is the contents of this file.

```
$ less views/my.address.php
<?php

(new CWidget())
->setTitle(_('The HTML Title of My Address Page'))
->addItem(new CDiv($data['my-ip']))
->show();
```



# FILE #4: MODULE FILE: Module.php

- ✓ Adds a menu entry.

```
$ less Module.php
```

```
<?php
namespace Modules\MyAddress;

use Core\CMSModule;
use APP;
use CMenulItem;

class Module extends CMSModule {
    public function init(): void {
        APP::Component()->get('menu.main')
            ->add((new CMenulItem(_('My Address'))))
                ->setAction('my.address')
            );
    }
}
```



# DISCOVER THE MODULE

- ✓ Administration → General → Modules → Scan directory

Modules ▾

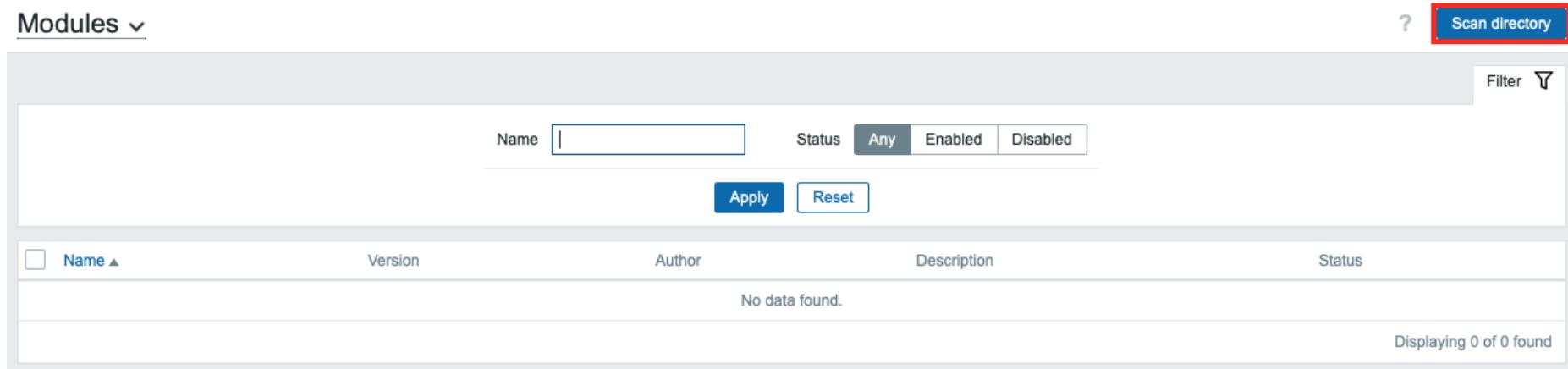
Name | Status Any Enabled Disabled

Apply Reset

<input type="checkbox"/> Name ▲	Version	Author	Description	Status
No data found.				

Displaying 0 of 0 found

Scan directory



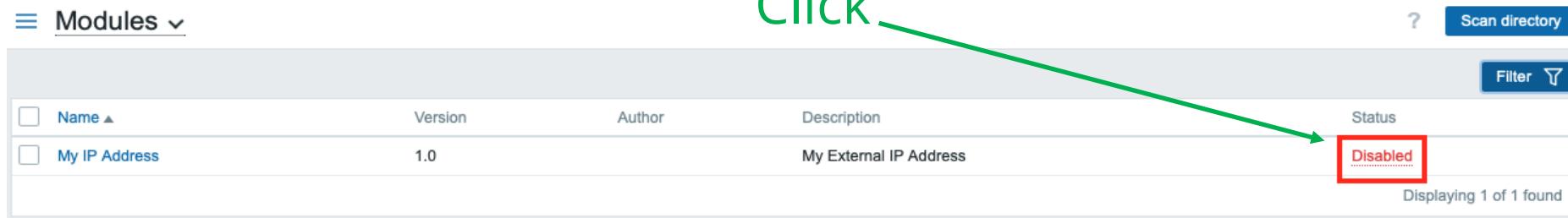
☰ Modules ▾

Click

<input type="checkbox"/> Name ▲	Version	Author	Description	Status
<input type="checkbox"/> My IP Address	1.0		My External IP Address	Disabled

Displaying 1 of 1 found

Scan directory



# DISCOVER THE MODULE

## Modules

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!

ZABBIX < □

Zabbix server

8.8.8.8

The HTML Title of My Address Page

ZABBIX < □

Zabbix server

8.8.8.8

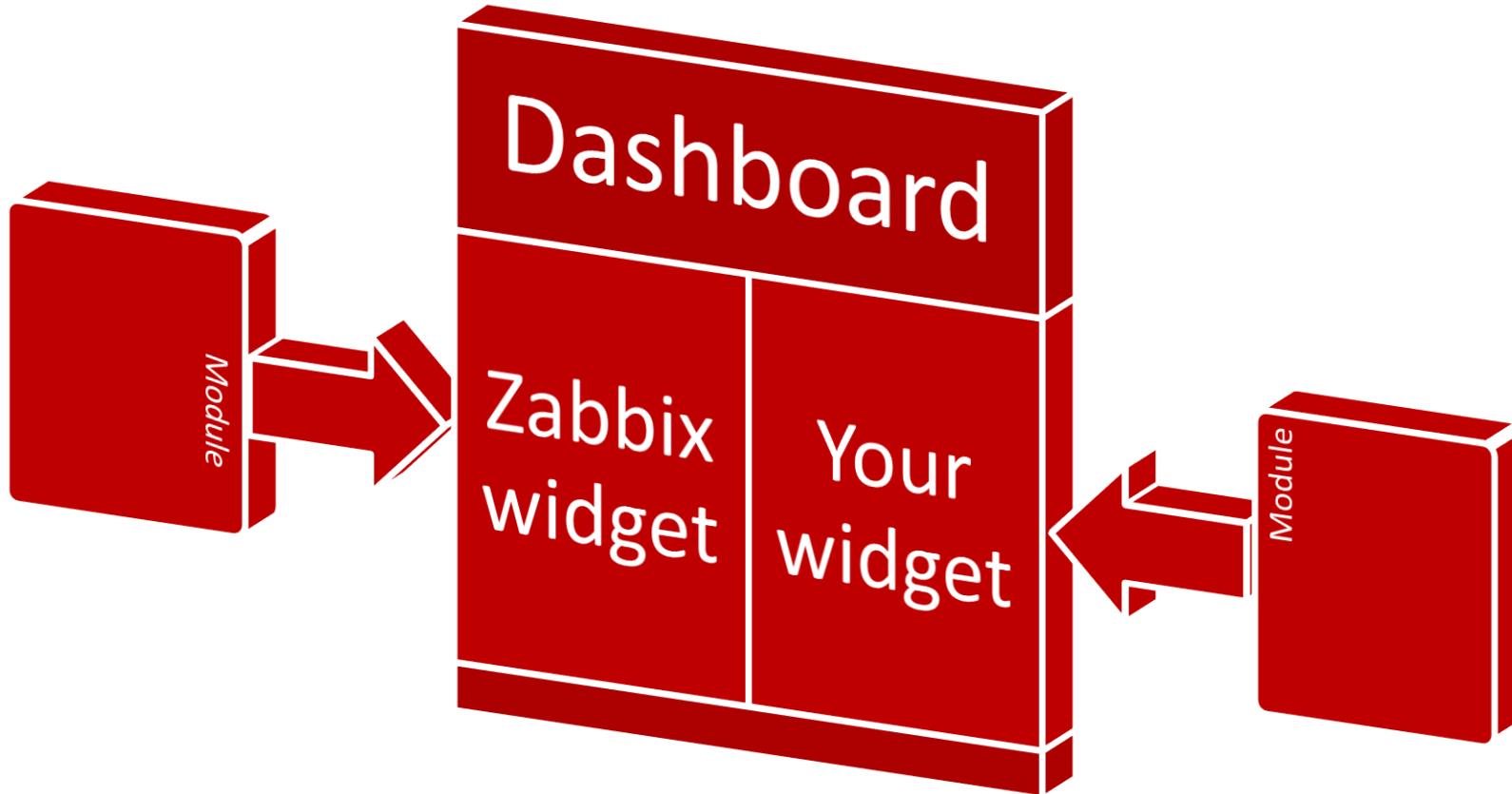
The HTML Title of My Address Page

My Address

- Monitoring
- Services
- Inventory
- Reports
- Configuration
- Administration
- My Address

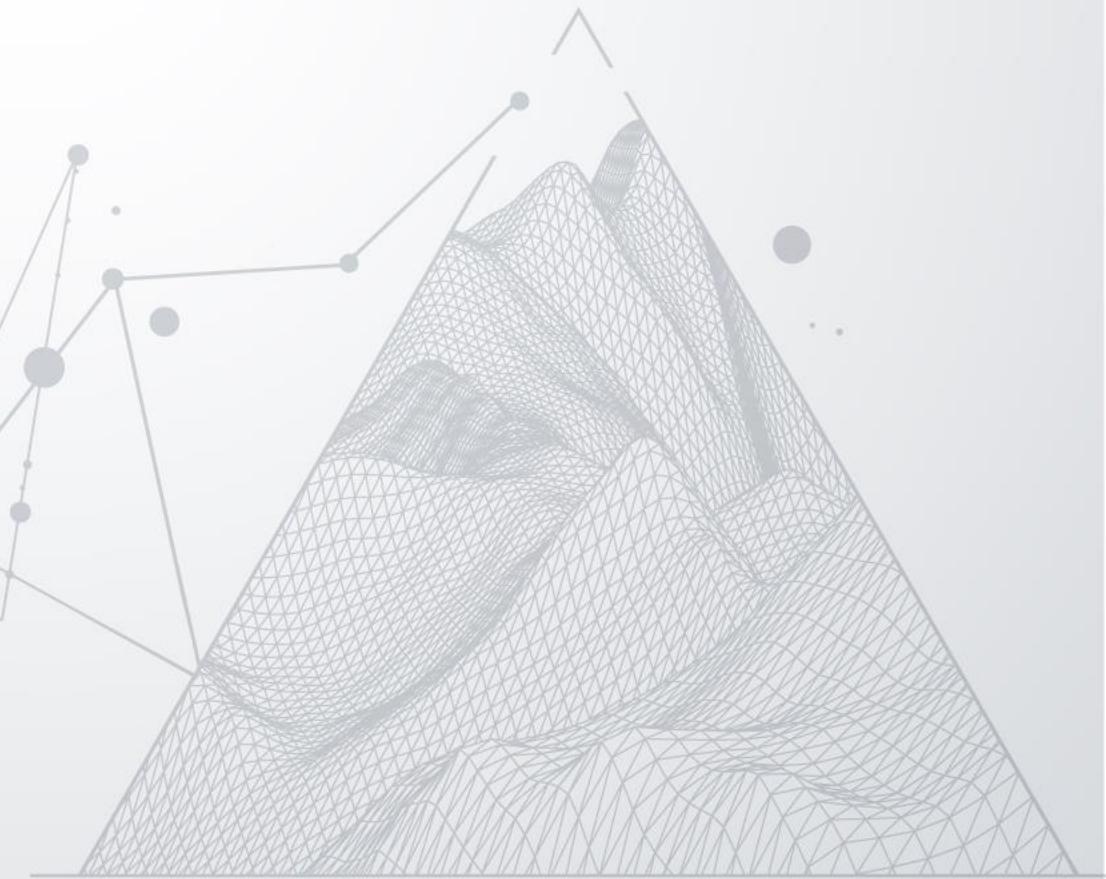


## 6.4: DASHBOARD WIDGET MODULE



# 03

EXTENDING FRONTEND:  
REBRANDING (SINCE 5.0)



# WHAT EXACTLY IS REBRANDED

ZABBIX

Username

Password

Remember me for 30 days

Sign in

ZABBIX

Zabbix server

Monitoring Services Inventory Reports Configuration Administration Support Integrations Help User settings Sign out

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

**Sign in**

## Dashboards

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

## LOREM

Monitoring Services Inventory Reports Configuration Administration

? Help User settings ⚡ Sign out

### Dashboards

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

0 selected **Delete**

(c) Lorem Ipsum



# 04

## EXTENDING AGENT



# AGENT 2 PLUGINS

## ① 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

## ② 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.

Item   Tags 1   Preprocessing 1

\* Name  Type **HTTP agent**

\* Key  Select

Type of information

\* URL  Parse

Query fields

Name	Value
<input type="text" value="name"/>	<input type="text" value="value"/> Remove

Add

Request type

\* Timeout

Request body type

Request body

Headers

Name	Value
<input type="text" value="name"/>	<input type="text" value="value"/> Remove

Add

# 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**

Media type   Message templates 5   Options

\* Name    Type

Parameters

Name	Value	Action
Message	{ALERT.MESSAGE}	<a href="#">Remove</a>
ParseMode		<a href="#">Remove</a>
Subject	{ALERT.SUBJECT}	<a href="#">Remove</a>
To	{ALERT.SENDTO}	<a href="#">Remove</a>
Token	<PLACE YOUR TOKEN>	<a href="#">Remove</a>

[Add](#)

\* Script  [Edit](#)

\* Timeout

Process tags

Include event menu entry

\* Menu entry name

\* Menu entry URL



# OFFICIAL WEBHOOKS

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

Item   Tags   Preprocessing

\* Name

Type  ▼

\* Key  Select

Type of information  ▼

Parameters ↓

Name	Value	Action
password	`\${\$PASSWORD}`	<a href="#">Remove</a>
url	`\${\$URL}`	<a href="#">Remove</a>
username	`\${\$USERNAME}`	<a href="#">Remove</a>

[Add](#)

\* Script  ✎

\* Timeout

# SCRIPT ITEM

```
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;

url = params.url + '/repositories/zabbix/?page_size=1000000';

request.addHeader('Authorization: Bearer ' + token);

response = request.get(url);
```



# 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
```

Item   Tags   Preprocessing

\* 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	Interval	Period	Action
Flexible	Scheduling	50s	1-7,00:00-24:00
Remove			

Add

\* History storage period Do not keep history Storage period 90d

\* Trend storage period Do not keep trends Storage period 365d

Value mapping Service state X

Select



# 06

## ZABBIX API FOR INTEGRATION



# ZABBIX API

- ✓ Retrieve Zabbix configuration (hosts, items, users etc.)
- ✓ Modify Zabbix configuration
- ✓ Get history/trend data
  - Grafana
- ✓ Configuration import/export
- ✓ Migration between production and test environments
- ✓ Automatic configuration management
  - Ansible Zabbix.Community module collection
- ✓ Future: push metrics



# ZABBIX API 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

The screenshot shows the Zabbix website at [www.zabbix.com/integrations](http://www.zabbix.com/integrations). The page has a navigation bar with links for PRODUCT, SOLUTIONS (which is highlighted), SUPPORT & SERVICES, TRAINING, PARTNERS, COMMUNITY, and ABOUT US. Below the navigation is a search bar labeled "Search by name". A large banner in the background features a network graph and the text "Monitoring and Integration Solutions". At the bottom, there is a grid of buttons representing different integration categories: All Categories (selected), Official Templates, Agents, API, Applications, AWS, Backups, Business KPI, Caching, CMDB, Clouds, Containers, CRM, DevOps, Databases, ERP, HA & Clusters, Helpdesks, Infrastructure, IoT, Java (selected), Logfiles, Mail, Message brokers, Mobile, Monitoring systems, Network, Notifications & Alerting, and Orchestration.



# ALL THE CODE IS ON GITHUB

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

Workshop tomorrow at 08:45  
Writing a Zabbix frontend module





THANK YOU!

