

Olá pessoal, hoje vou falar de um micro serviço que criei para simplificar o uso do Zabbix sender, criando um webhook (API), para que você consiga enviar as suas métricas. Aqui vou colocar um “how to” de como seguir mas lembre, quanto mais experiência você tiver com o Zabbix maiores são as possibilidades desse método (Preprocessing!).

Mas o que é o Sender API for Zabbix

É um componente criado para facilitar o uso do recurso “sender” da Ferramenta de monitoramento Zabbix.

“O sender se trata de uma lib. que pode ser integrada diretamente no código para enviar métricas para o Zabbix”

Por muitas vezes é necessário fazer diversas homologações e testes para adicionar novos recursos para a aplicação o que pode inviabilizar uma integração com recursos terceiros ou em alguns casos quando skill técnicos são necessários.

Com o Sender API for Zabbix é necessário apenas apontar para o endpoint (API), disponibilizado passando algumas informações para o host/item

E o valor da métrica será enviado ao Zabbix sem a necessidade de lib. ou pacotes no código da sua aplicação, poderá ser chamado de qualquer lugar com suporte a http.

How to use

É necessário ter um host e item do tipo zabbix trapper para ser passado no body ao realizar o post, a descrição está descrita abaixo:

Criar um host no Zabbix

- Em configurações / Hosts / Create host

Defina um nome e adicione a um grupo (demais opções conforme a sua necessidade).

- No exemplo estou colocando o nome Sender, ao finalizar clique em add.

Hosts

Agora crie um template com o nome desejado.

- Em Templates/Create Template

The screenshot shows the Zabbix interface under the 'Templates' section. A new template is being created with the following details:

- Template name:** sender_api
- Visible name:** (empty)
- Groups:** type here to search
- Description:** (empty)

At the bottom, there are 'Add' and 'Cancel' buttons.

Acesse novamente o template e crie o item conforme imagens a seguir.

The screenshot shows the Zabbix interface under the 'Items' section. A new item is being created with the following details:

- Host groups:** type here to search
- Linked templates:** type here to search
- Name:** sender_api

Below the search fields, there are 'Tags' (And/Or Or), 'Contains tag', 'Equals value', and a 'Remove' button. At the bottom are 'Apply' and 'Reset' buttons.

The main table lists the created item:

	Hosts	Applications	Items	Triggers	Graphs	Screens	Discovery	Web	Linked templates	Linked to templates
<input checked="" type="checkbox"/>	sender_api	Hosts	Applications	Items	Triggers	Graphs	Screens	Discovery	Web	

Create Item:

The screenshot shows the Zabbix interface under the 'Items' section, specifically the search and filter interface. The following filters are applied:

- Host groups:** type here to search
- Hosts:** sender_api
- Application:** (empty)
- Name:** (empty)
- Key:** (empty)
- Type:** all
- Type of information:** all
- State:** all
- Status:** all
- Triggers:** all
- Template:** all
- Discovery:** all
- Update interval:** (empty)
- History:** (empty)
- Trends:** (empty)

At the bottom, there are 'Apply' and 'Reset' buttons. A note says 'Subfilter affects only filtered data'. The main table below shows no data found.

	Name	Triggers	Key	Interval	History	Trends	Type	Applications	Status	Info
No data found.										

Items

The screenshot shows the 'Items' configuration page in Zabbix. The top navigation bar includes links for 'All templates / sender_api', 'Applications', 'Items' (which is selected and highlighted in blue), 'Triggers', 'Graphs', 'Screens', 'Discovery rules', and 'Web scenarios'. Below the navigation, there are two tabs: 'Item' (selected) and 'Preprocessing'. The main form fields are: 'Name' (meu item), 'Type' (Zabbix trapper), 'Key' (sender_metrica), 'Type of information' (Numeric (unsigned)), and 'Units' (empty). A 'Select' button is also visible next to the Key field.

No exemplo acima incluímos o nome para o item, o tipo como Zabbix Trapper (que permite receber dados da API) e a chave (este que é o identificador para a métrica).

Tipo de informação pode ser definido conforme a sua necessidade o Sender API pode enviar Strings ou valores numéricos para o Zabbix, porém o mesmo será interpretado conforme o tipo definido, ressaltando que existem formas para alterar o tipo de dado e ou estrutura usando Preprocessing (não discutidos neste documento).

Agora adicione o template ao host criado no primeiro step.

Hosts

The screenshot shows the 'Hosts' configuration page in Zabbix. The top navigation bar includes links for 'All hosts / Sender', 'Enabled', 'ZBX', 'SNMP', 'JMX', 'IPMI', 'Applications', 'Items' (selected), 'Triggers', 'Graphs', 'Discovery rules', and 'Web scenarios'. Below the navigation, there are several tabs: 'Host' (selected), 'Templates' (selected), 'IPMI', 'Tags', 'Macros', 'Inventory', and 'Encryption'. Under the 'Templates' tab, there is a section for 'Linked templates' with a table header ('Name', 'Action'). Below this is a search input field labeled 'Link new templates' containing 'sender_api' with a delete icon, and a 'Select' button. At the bottom of the screen are five buttons: 'Update', 'Clone', 'Full clone', 'Delete', and 'Cancel'.

Atualize e agora vamos usar o microserviço do Sender API, neste exemplo será configurado no Docker.

Faça o download da imagem no Docker Hub:

- docker push rafasera/sender-api-zbx:latest

 **rafasera / sender-api-zbx**

This repository does not have a description

Inicie a mesma conforme imagem e descrição abaixo

```
rafael@Rafaels-MacBook-Pro ~ % docker run -d -p 80:80 -e API_URL="104.44.133.54" -e API_PORT=10051 rafasera/sender-api-zbx
cedc758baa5b2db5ba4c789308dc3c1b5c8bd12c6529d4f1eddf89429c9bab4f
rafael@Rafaels-MacBook-Pro ~ %
```

Aqui foi exposta a porta 80

Em API_URL informe o endereço do Zabbix.

Em API_PORT informe a porta do seu Zabbix, default é 10051

Após isso aguarde inicializar.

“OBS você pode encontrar o mesmo tutorial acessando o container no caminho default que definiu no meu caso seria a localhost”

Por fim vamos utilizar o Sender-API

“Eu vou usar o Programa Postman para efetuar as requests”

Abra o Postman e acesse a rota:

- <https://filgs.com.br/api/v1/zabbix-sender>

Tipo:

- POST

Body:

```
{  
    "host": "Sender",  
    "item": "sender_metrica",  
    "value": "99"  
}
```

Obs.

- O endereço localhost foi atribuído pelo docker ressaltando que estou usando localmente.
- O body deve conter o host criado no Zabbix para receber a métrica
- A chave do Item (key)
- E o valor da métrica que pode ser enviado por outra ferramenta ou aplicação.

http://localhost/api/v1/zabbix-sender

POST http://localhost/api/v1/zabbix-sender

Params Authorization Headers (8) Body **JSON** Pre-request Script Tests Settings Cookies

Body Beautify

```

1 {
2   "host": "Sender",
3   "item": "sender_metrica",
4   "value": "99"
5 }
```

Body Cookies Headers (7) Test Results Status: 201 Created Time: 44 ms Size: 305 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "title": "Zabbix_Sender_API",
3   "version": "0.0.1",
4   "status": "sucess"
5 }
```

Latest data

Host	Name	Last check	Last value	Change
Sender	API (1 Item)	2021-03-06 15:47:12	99	Graph
	meu item			Displaying 1 of 1 found

Não havendo problemas de estrutura ou comunicação com o Sender-API receberá a mensagem de sucesso.

Em seguida a métrica enviada chegara no seu Zabbix.