

Pessoal tudo bem com todos? Espero que sim.

Peço desculpas por não ter feito mais nenhum post mas final de ano a correria fica imensa como sabem, principalmente para nós que trabalhamos com TI.

Bem, já que tivemos esse tempo parados vamos retomar com um assunto muito legal. O monitoramento de APIs direto no Zabbix sem precisar usar nenhum Script para processar os dados.

Como todos nós sabemos, monitorar o ambiente produtivo é uma questão de sucesso; imaginem o seu usuário acessando um aplicativo qualquer e a API ao qual o mesmo se conecta está fora ou com algum problema de acesso, provavelmente o seu “cliente” não vai ficar feliz e vai te ligar bem chateado.

Então que tal aproveitar que a nova versão do Zabbix 4.0 tem suporte a monitores/sensores HTTP?

Ok, sem mais papo vamos colocar a mão na massa.

visão global

Este tipo de item permite a pesquisa de dados usando o protocolo HTTP / HTTPS. O trapping também é possível usando o remetente Zabbix ou o protocolo do emissor Zabbix.

A verificação do item HTTP é executada pelo servidor Zabbix. No entanto, quando os hosts são monitorados por um proxy Zabbix, as verificações de itens HTTP são executadas pelo proxy.

As verificações de item HTTP não requerem nenhum agente em execução em um host que

está sendo monitorado.

O agente HTTP suporta HTTP e HTTPS. O Zabbix seguirá opcionalmente redirecionamentos (veja a opção Seguir redireciona abaixo). O número máximo de redirecionamentos é codificado para 10 (usando a opção cURL CURLOPT_MAXREDIRS).

Configuração

Para configurar um item HTTP:

- Vá para: Configuração → Hosts
- Clique nos itens na linha do host
- Clique em Criar item
- Insira os parâmetros do item no formulário

Item Preprocessing

* Name

Type

* Key

* URL

Query fields

Name	Value	
scroll	=>	10s <input type="button" value="Remove"/>

Request type

Timeout

Request body type

Request body

```
{
  "query": {
    "bool": {
      "must": [
        "match": {
          "itemid": 28275
        }
      ]
    }
  }
}
```

Headers

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

Required status codes

Follow redirects

Retrieve mode

Convert to JSON

HTTP proxy

HTTP authentication

SSL verify peer

SSL verify host

SSL certificate file

SSL key file

SSL key password

* Host interface

Type of information

Units

* Update interval

Custom intervals

Type	Interval	Period	Action
<input type="text" value="Flexible"/> <input type="text" value="Scheduling"/>	<input type="text" value="50s"/>	<input type="text" value="1-7:00:00-24:00"/>	<input type="button" value="Remove"/>

* History storage period

* Trend storage period

Show value

Enable trapping

Allowed hosts

New application

Todos os campos de entrada obrigatórios estão marcados com um asterisco vermelho.

Exemplos

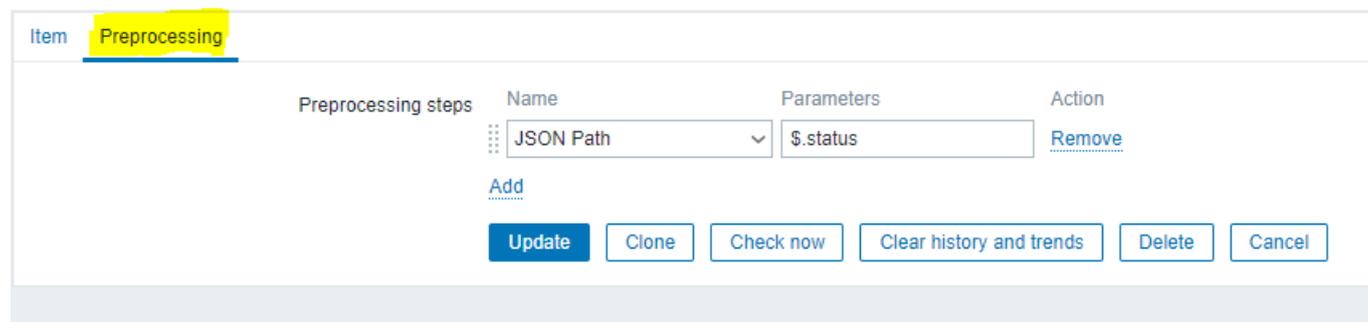
Exemplo 1

Aqui eu tenho o seguinte retorno a partir da API:

```
Pretty Raw Preview JSON
1 {
2   "status": "success",
3   "data": {
4     "resultType": "matrix",
5     "result": []
6   }
7 }
```

Sendo assim vou pegar o nó que identifica o status de sucesso da minha aplicação.

No Zabbix:



Acesse a opção (Guia) Preprocessing.

Preencha os steps

Name -> JSON Path

Parameters -> \$.status

Em parameters é que está a solução desse sensor, para cada nível abaixo devemos colocar da seguinte forma.

Exemplo:

\$.body.exemplo.exemplo1 (aqui faremos a leitura do valor em exemplo1)

Após colocar as informações necessárias clique em ADD.

Alguns minutos depois já pode conferir o resultado em “Latest Data”

Prometheus - teste (1 Item)						
Prometheus	1m	90d	HTTP agent	2018-11-27 11:24:25	success	
Prometheus_teste						

Acima coloquei um exemplo real de uso (meu caso), mas como também quero que todos aprendam aqui vai um exemplo usando a API do <https://openweathermap.org>

Vamos lá.

Primeiro vamos configurar os campos:

* Name

Type

* Key

* URL

Query fields

Name	Value	
<input type="text" value="lat"/>	=> <input type="text" value="35"/>	Remove
<input type="text" value="lon"/>	=> <input type="text" value="139"/>	Remove
<input type="text" value="appid"/>	=> <input type="text" value="b6907d289e10d714a6e88b30761fae2?"/>	Remove

[Add](#)

Request type

Timeout

Devemos colocar o Name, Key, URL e aqui vamos definir o Query fields.

Em URL coloque: <https://samples.openweathermap.org/data/2.5/weather?>

Em Query fields: lat = 35

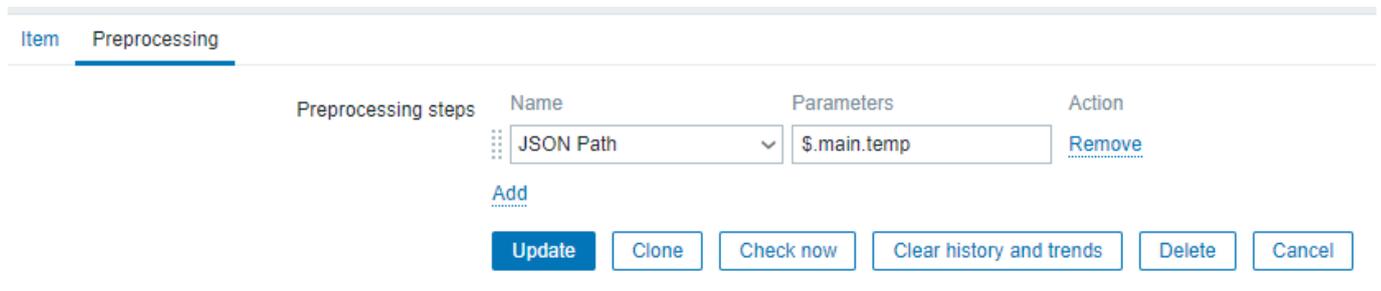
lon = 139

appid = b6907d289e10d714a6e88b30761fae22

Request type: GET

Type of information: Numeric (float)

Agora Ajustando o Preprocessing



The screenshot shows a web interface for configuring preprocessing steps. At the top, there are tabs for 'Item' and 'Preprocessing', with 'Preprocessing' selected. Below this is a table with columns: 'Preprocessing steps', 'Name', 'Parameters', and 'Action'. A single step is listed with 'JSON Path' in the Name column and '\$.main.temp' in the Parameters column. An 'Add' link is visible below the table. At the bottom of the configuration area, there are several buttons: 'Update', 'Clone', 'Check now', 'Clear history and trends', 'Delete', and 'Cancel'.

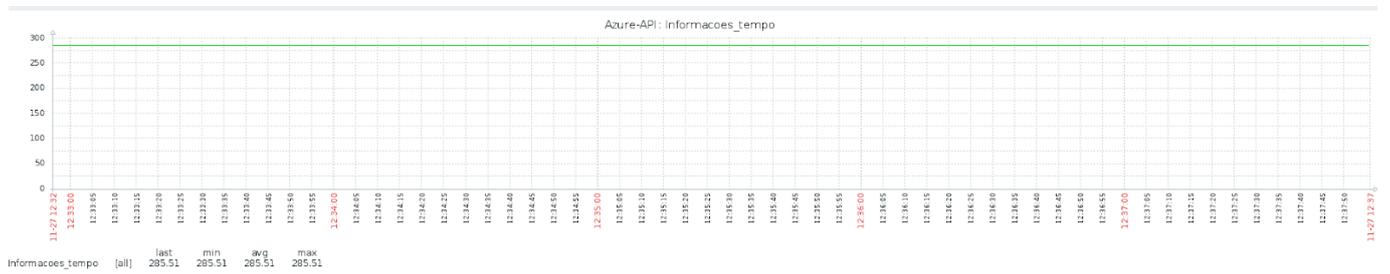
O mapeamento deve ficar dessa forma: \$.main.temp

A informação capturada será:

```
  "base": "stations",  
  "main": {  
    "temp": 285.514,  
    "pressure": 1013.75,  
    "humidity": 100,  
    "temp_min": 285.514,  
    "temp_max": 285.514,  
    "sea_level": 1023.22,  
    "grnd_level": 1013.75
```

Após finalizado clique em Add.

Alguns minutos após pode conferir em "Latest Data" e você terá todos os dados, como estamos pegando um valor numérico o gráfico já ficará configurado conforme imagem abaixo.



Bem pessoal, espero que estas informações sejam úteis e possam ajudar a todos.

E como sempre qualquer dúvida fico a disposição.

Abraço a todos.