

## 17 HTTP代理

### 概述

此监控项类型允许使用HTTP/HTTPS协议进行数据轮询。使用Zabbix sender或Zabbix sender协议也可以进行捕获。

HTTP代理同时支持HTTP和HTTPS。Zabbix可以选择跟随重定向（参考下文*Follow redirects*的选项）

了解何时使用HTTPS协议，另请参阅[已知问题](#)

Zabbix server/proxy必须首先配置cURL(libcurl)支持。

### 配置

配置HTTP监控项：

- 进入： *Configuration* → *Hosts*
- 在主机的那行点击 *Items*
- 点击 *Create item*
- 在表格中输入监控项的参数

The screenshot shows the configuration page for an HTTP agent item in Zabbix. The form is titled "Item Preprocessing". Key fields include:

- Name:** HTTP agent item
- Type:** HTTP agent
- Key:** http\_value\_search
- URL:** http://localhost:9200/str/values/\_search
- Query fields:** scroll (10s)
- Request type:** POST
- Timeout:** 3s
- Request body type:** Raw data, JSON data, XML data
- Request body:** {"query": {"bool": {"must": [{"match": {"\_id": "28275"}]}}
- Headers:** name (value)
- Required status codes:** 200
- Follow redirects:**
- Retrieve mode:** Body, Headers, Body and headers
- Convert to JSON:**
- HTTP proxy:** http://user[:password]@[proxy.example.com][:port]
- HTTP authentication:** None
- SSL verify peer:**
- SSL verify host:**
- SSL certificate file:** [empty]
- SSL key file:** [empty]
- SSL key password:** [empty]
- Host interface:** 127.0.0.1 : 10050
- Type of information:** Numeric (unsigned)
- Units:** [empty]
- Update interval:** 30s
- Custom intervals:** Flexible, Scheduling (50s, 1-7,00:00-24:00)
- History storage period:** 90d
- Trend storage period:** 365d
- Show value:** As is
- Enable trapping:**
- Allowed hosts:** 104.24.103.152
- New application:** [empty]

所有标有红色星号的为必填字段。

需要的HTTP监控项特定信息的字段是：

Type	在这里选择 <b>HTTP agent</b>
Key	输入一个唯一的监控项键值

URL	<p>连接和检索数据的URL。例如：  <a href="https://www.google.com">https://www.google.com</a>  <a href="http://www.zabbix.com/download">http://www.zabbix.com/download</a>\\可以用Unicode字符指定域名。在执行web场景步骤时，它们将自动转换为ASCII</p> <p><b>Parse</b> 可以使用Parse按钮将可选查询字段(比如?name=Admin&amp;password=mypassword)与URL分离，将属性和值移动到查询字段中，以便自动URL编码。  限制在2048个字符。  支持的宏: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, 用户宏, 低级发现宏  这是设置CURLOPT_URL cURL选项。</p>
Query fields	<p>URL的变量 (参见上文)。  指定为属性和值对。  值是自动的URL编码。从宏中解析值，然后自动编码url  支持的宏: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, 用户宏, 低级自动发现宏。  设置cURL选项 <a href="#">CURLOPT_URL</a>。</p>
Request type	选择请求方法类型: <i>GET, POST, PUT or HEAD</i>
Timeout	<p>Zabbix不会花超过设定的时间来处理URL (最大1分钟)。实际上，这个参数定义了连接URL的最大时间和执行HTTP请求的最大时间。因此Zabbix不会在一次检查中花费超过2倍的超时时间。  支持时间后缀，例如 30s, 1m。  支持的宏: 用户宏, 低级发现宏。  设置cURL选项 <a href="#">CURLOPT_TIMEOUT</a></p>
Request body type	<p>选择请求体类型:  <b>Raw data</b> - 自定义HTTP请求体，替换宏，但不执行编码。  <b>JSON data</b> - HTTP请求体是JSON格式的，宏可以用作字符串、数字、真和假;用作字符串的宏必须包含在双引号中。从宏中解析值，然后自动转义。如果没有指定header那么服务器将把默认的header值设置为"Content-Type: application/json"  <b>XML data</b> - HTTP请求体的XML格式。宏可以用作文本节点、属性或CDATA部分。从宏中解析值，然后在文本节点和属性中自动转义。如果没有指定header那么服务器将把默认的header值设置为 "Content-Type: application/xml"  注意选择 <i>XML data</i>，需要libxml2的支持。</p>
Request body	<p>输入请求体  支持的宏: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, 用户宏, 低级自动发现宏。</p>
Headers	<p>执行请求时将发送的自定义HTTP头。  指定为属性和值对。  支持的宏: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, 用户宏, 低级自动发现宏。  设置 <a href="#">CURLOPT_HTTPHEADER</a> cURL option.</p>
Required status codes	<p>期望的HTTP状态码的列表。如果Zabbix得到不在列表中的代码，那么这个项目将不受支持。如果为空，则不执行检查。  例如: 200, 201, 210-299  列表里支持的宏: 用户宏, 低级自动发现宏。  这个使用了 <a href="#">CURLINFO_RESPONSE_CODE</a> cURL option.</p>
Follow redirects	<p>标记复选框以跟随HTTP重定向。  设置 <a href="#">CURLOPT_FOLLOWLOCATION</a> cURL option.</p>
Retrieve mode	<p>选择必须检索的响应部分:  <b>Body</b> - 仅主体  <b>Headers</b> - 仅头部  <b>Body and headers</b> - 主体和头部</p>
Convert to JSON	<p>头文件作为属性和值对保存在"header" 键下。  如果遇到 'Content-Type: application/json' 主体被保存为对象，否则它被存储为string, 例如:</p> <pre>{   "header": {     "&lt;key&gt;": "&lt;value&gt;",     "&lt;key2&gt;": "&lt;value&gt;"   },   "body": &lt;body&gt; }</pre>

<p><i>HTTP proxy</i></p>	<p>可以使用格式 <code>http://[username[:password]@]proxy.mycompany.com[:port]</code> 指定要使用的HTTP代理。 默认将使用1080端口。 如果指定,代理将覆盖与代理相关的环境变量,如 <code>http_proxy</code> 或 <code>HTTPS_PROXY</code>. 如果没有指定,代理将不会覆盖与代理相关的环境变量。输入的值将被传递 "as is", 没有进行健全检查。 您还可以输入SOCKS代理地址。如果您指定了错误的协议,那么连接将失败,监控项将不受支持。由于没有指定协议,代理将被视为HTTP代理。 注意 HTTP代理只支持简单的身份验证。 支持的宏: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, 用户宏, 低级自动发现宏。 设置 <code>CURLOPT_PROXY</code> cURL option.</p>
<p><i>HTTP authentication</i></p>	<p>验证类型: <b>None</b> - 不使用身份验证。 <b>Basic authentication</b> - 使用脚本身份验证。 <b>NTLM authentication</b> - 使用NTLM (<a href="#">Windows NT LAN Manager</a>) 验证。 选择身份验证方法将为输入用户名和密码提供两个额外的字段,其中支持用户宏和低级发现宏。 设置 <code>CURLOPT_HTTPAUTH</code> cURL option.</p>
<p><i>SSL verify peer</i></p>	<p>标记复选框以验证web服务器的SSL证书。服务器证书将自动从系统范围的证书颁发机构(CA)位置获取。 可以使用Zabbix服务器或代理配置参数 <code>SSLCALocation</code> 重写CA文件的位置。 设置 <code>CURLOPT_SSL_VERIFYPEER</code> cURL option.</p>
<p><i>SSL verify host</i></p>	<p>标记复选框以验证web服务器证书的通用名称字段或主题备用名称字段是否匹配。 设置 <code>CURLOPT_SSL_VERIFYHOST</code> cURL option.</p>
<p><i>SSL certificate file</i></p>	<p>用于客户端身份验证的SSL证书文件的名称。证书文件必须是PEM<sup>1</sup>格式。如果证书文件也包含私钥,则将SSL密钥文件字段保留为空。如果密钥已加密,请在SSL密钥密码字段中指定密码。包含此文件的目录由Zabbix server或zabbix proxy配置参数 <code>SSLCertLocation</code> 指定。 支持的宏: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, 用户宏, 低级自动发现宏。 设置 <code>CURLOPT_SSLCERT</code> cURL option.</p>
<p><i>SSL key file</i></p>	<p>用于客户端身份验证的SSL私钥文件的名称。私钥文件必须是PEM<sup>1</sup>格式。包含此文件的目录由Zabbix server或zabbix proxy配置参数 <code>SSLKeyLocation</code> 指定。 支持的宏: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, 用户宏, 低级自动发现宏。 设置 <code>CURLOPT_SSLKEY</code> cURL option.</p>
<p><i>SSL key password</i></p>	<p>SSL私钥文件密码。 支持的宏: 用户宏, 低级自动发现宏 设置 <code>CURLOPT_KEYPASSWD</code> cURL option.</p>
<p><i>Enable trapping</i></p>	<p>选中此复选框后,该项目也将作为 <b>trapper</b> 监控项项发挥作用,并将接受Zabbix sender或使用Zabbix sender协议发送给该监控项的数据。</p>
<p><i>Allowed hosts</i></p>	<p>只有勾选了 <i>Enable trapping</i> 复选框才可见。 由逗号分隔的IP地址列表,可选地使用CIDR符号或主机名。\\如果指定,传入连接将仅从这里列出的主机接受。 如果启用了IPv6 <code>'127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1'</code> 这些是一样的, <code>'::/0'</code> 将允许任何IPv4或IPv6地址。 <code>'0.0.0.0/0'</code> 可用于允许任何IPv4地址。 注意, IPv4兼容的IPv6地址 (<code>0000::/96 prefix</code>) 能够被支持,但 <a href="#">RFC4291</a> 不推荐使用。 示例: <code>Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.domain</code> 在这个字段,空格和 <b>用户宏</b> 是被允许的。</p>

如果HTTP代理字段为空,则使用HTTP代理的另一种方法是设置与代理相关的环境变量。

对于HTTP - 为Zabbix server用户设置“http\_proxy”环境变量。例如:

`http_proxy=http://proxy_ip:proxy_port.`

对于HTTPS - 设置“HTTPS\_PROXY”环境变量。例如:

`HTTPS_PROXY=http://proxy_ip:proxy_port.` 可以通过运行shell命令获得更多细节: `# man curl.`

[1] Zabbix只支持PEM格式的证书和私有密钥文件。如果您的证书和私钥数据是PKCS #12格式文件(通常扩展名为 \*.p12 or \*.pfx) 您可以使用以下命令从它生成PEM文件:

```
openssl pkcs12 -in ssl-cert.p12 -clcerts -nokeys -out ssl-cert.pem
openssl pkcs12 -in ssl-cert.p12 -nocerts -nodes -out ssl-cert.key
```

## 示例

### 示例 1

发送简单的GET请求来从诸如Elasticsearch这样的服务中检索数据：

- 使用URL创建一个GET项：`localhost:9200/?pretty`
- 注意其响应

```
{
  "name" : "YQ2VAY-",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "kH4CYqh5QfqgeTsjh2F9zg",
  "version" : {
    "number" : "6.1.3",
    "build_hash" : "af51318",
    "build_date" : "2018-01-26T18:22:55.523Z",
    "build_snapshot" : false,
    "lucene_version" : "7.1.0",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You know, for search"
}
```

- 现在使用JSONPath预处理步骤提取版本号：`$.version.number`

### 示例 2

发送简单的POST请求来检索来自Elasticsearch等服务的数据：

- 使用URL创建一个POST项：[http://localhost:9200/str/values/\\_search?scroll=10s](http://localhost:9200/str/values/_search?scroll=10s)
- 配置以下POST主体以获取处理器负载（每核1分钟的平均值）

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

```
        "range": {
            "clock": {
                "gt": 1517565836,
                "lte": 1517566137
            }
        }
    }
}
```

- Received
- 接收:

```
{
  "_scroll_id":
  "DnF1ZxJ5VGhlbkZldGNobQAAAAAAAAAAkFlLRMLZBWS1UU1pxTmdEeGVwQjRBTfEAAAAAAAAAJRZ
  ZUTJWQVktVFNAcU5nRHhlcEI0QUxRAAAAAAAAAACYWWEyVkfZLVRTWnFOZ0R4ZXBCNEFMUQAAAAA
  AAAAnFlLRMLZBWS1UU1pxTmdEeGVwQjRBTfEAAAAAAAAAKBZZUTJWQVktVFNAcU5nRHhlcEI0QUx
  R",
  "took": 18,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1,
    "max_score": 1.0,
    "hits": [{
      "_index": "dbl",
      "_type": "values",
      "_id": "dqX9VWEBV6sEKSMYk6sw",
      "_score": 1.0,
      "_source": {
        "itemid": 28275,
        "value": "0.138750",
        "clock": 1517566136,
        "ns": 25388713,
        "ttl": 604800
      }
    }
  ]
}
```

- 现在使用JSONPath预处理步骤获取项值: \$.hits.hits[0].\_source.value

From:  
<https://www.zabbix.com/documentation/4.0/> - **Zabbix Documentation 4.0**

Permanent link:  
<https://www.zabbix.com/documentation/4.0/zh/manual/config/items/itemtypes/http>

Last update: **2018/11/30 06:44**

