

17 HTTP агент

Обзор

Этот тип элемента данных позволяет получать данные с использованием HTTP/HTTPS протокола. Трапы также возможны с использованием Zabbix sender или протокола Zabbix sender.

Проверка HTTP элемента данных выполняется на стороне Zabbix сервера. Однако, когда узлы сети наблюдаются через Zabbix прокси, проверки HTTP элементов данных выполняются этими прокси.

Проверки HTTP элементов данных не требуют наличия какого-либо агента, запущенного на наблюдаемом узле сети.

HTTP агент поддерживает как HTTP, так и HTTPS. Zabbix будет опционально следовать перенаправлениям (смотрите ниже опцию *Следовать перенаправлениям*). Максимальное количество перенаправлений вшито в код и равно 10 (используется cURL опция CURLOPT_MAXREDIRS).

Смотрите также [известные проблемы](#) при использовании HTTPS протокола.

Zabbix сервер/прокси должны быть изначально сконфигурированы с поддержкой cURL (libcurl).

Настройка

Для настройки HTTP элемента данных:

- Перейдите в: *Настройка* → *Узлы сети*
- Нажмите на *Элементы данных* в строке с узлом сети
- Нажмите на *Создать элемент данных*
- Введите параметры элемента данных в диалоге

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

- Name:** HTTP agent item (required, marked with a red asterisk)
- Type:** HTTP agent (dropdown menu)
- Key:** http_value_search (required, marked with a red asterisk)
- URL:** http://localhost:9200/str/values/_search (required, marked with a red asterisk)
- Query fields:** A table with columns "Name" and "Value". One entry is "scroll" with value "10s".
- Request type:** POST (dropdown menu)
- Timeout:** 3s
- Request body type:** Raw data, JSON data, XML data (radio buttons)
- Request body:** A text area containing a JSON object: {"query": {"bool": {"must": [{"match": {"_id": "28275"}]}}}}
- Headers:** A table with columns "Name" and "Value". One entry is "name" with value "value".
- Required status codes:** 200
- Follow redirects:**
- Retrieve mode:** Body, Headers, Body and headers (radio buttons)
- Convert to JSON:**
- HTTP proxy:** http://user[:password]@[proxy.example.com][:port]
- HTTP authentication:** None (dropdown menu)
- SSL verify peer:**
- SSL verify host:**
- SSL certificate file:** (text input)
- SSL key file:** (text input)
- SSL key password:** (text input)
- Host interface:** 127.0.0.1 : 10050 (required, marked with a red asterisk)
- Type of information:** Numeric (unsigned) (dropdown menu)
- Units:** (text input)
- Update interval:** 30s (required, marked with a red asterisk)
- Custom intervals:** A table with columns "Type", "Interval", "Period", and "Action". One entry is "Flexible" with interval "50s" and period "1-7,00:00-24:00".
- History storage period:** 90d (required, marked with a red asterisk)
- Trend storage period:** 365d (required, marked with a red asterisk)
- Show value:** As is (dropdown menu) with a link "show value mappings"
- Enable trapping:**
- Allowed hosts:** 104.24.103.152
- New application:** (text input)

Все обязательные поля ввода отмечены красной звёздочкой.

Поля, которые требуют специфичную информацию HTTP элементов данных:

Тип	Выберите здесь HTTP агент .
Ключ	Введите уникальный ключ элемента данных.

URL	<p>URL для подключения и получения данных. Например: https://www.google.com http://www.zabbix.com/download</p> <p>Имена доментов можно указывать Юникод символами. Они автоматически конвертируются методом punycode в ASCII при выполнении шага веб-сценария.</p> <p>Кнопку <i>Анализ</i> можно использовать чтобы отделить из URL опциональные поля запроса (например, <code>?name=Admin&password=myspassword</code>), переместив атрибуты и значения в <i>Поля запроса</i>, чтобы URL кодировка выполнялась автоматически.</p> <p>Ограничено 2048 символами.</p> <p>Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, пользовательские макросы, макросы низкоуровневого обнаружения.</p> <p>Это поле задает CURLOPT_URL [en] опцию cURL.</p>
Поля запроса	<p>Переменные для URL (см. выше).</p> <p>Задаются в виде пар атрибутов и значений.</p> <p>Значения URL кодируются автоматически. Значения с макросов раскрываются и затем URL кодируются автоматически.</p> <p>Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, пользовательские макросы, макросы низкоуровневого обнаружения.</p> <p>Это поле задает CURLOPT_URL [en] опцию cURL.</p>
Тип запроса	<p>Выберите тип метода запроса: <i>GET</i>, <i>POST</i>, <i>PUT</i> или <i>HEAD</i></p>
Время ожидания	<p>Zabbix не будет тратить более указанного времени при обработке URL (максимум 1 минута). В действительности же этот параметр определяет максимальное время содания подключения к URL и максимальное время для выполнения HTTP запроса. Следовательно, Zabbix не будет тратить более 2 x Время ожидания секунд за одну проверку.</p> <p>Поддерживаются суффиксы времени, например, 30s, 1m.</p> <p>Поддерживаемые макросы: пользовательские макросы, макросы низкоуровневого обнаружения.</p> <p>Это поле задает CURLOPT_TIMEOUT [en] опцию cURL.</p>
Типа запроса тела	<p>Выберите тип запроса тела:</p> <p>Сырые данные - пользовательское тело HTTP запроса, макросы заменяются, но кодирование не выполняется</p> <p>JSON данные - тело HTTP запроса в формате JSON. Макросы можно использовать в виде строки, числа, true или false; макросы, которые используются в виде строки должны быть заключены в двойные кавычки. Значения из макросов раскрываются и затем экранируются автоматически. Если в заголовках "Content-Type" не задан , тогда будет указано значение по умолчанию "Content-Type: application/json"</p> <p>XML данные - тело HTTP запроса в формате XML. Макросы можно использовать в виде нод текста, атрибутов или CDATA секции. Значения из макросов раскрываются и затем экранируются автоматически в нодах текста и в атрибутах. Если в заголовках "Content-Type" не задан, тогда будет указано значение по умолчанию "Content-Type: application/xml"</p> <p><i>Обратите внимание</i>, что выбор <i>XML данные</i> требует наличия поддержки libxml2.</p>
Запрос тела	<p>Введите тело запроса.</p> <p>Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, пользовательские макросы, макросы низкоуровневого обнаружения.</p>

<p><i>Заголовки</i></p>	<p>Пользовательские HTTP заголовки, которые будут отправлены при выполнении запроса. Задаются в виде пар атрибутов и значений. Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, пользовательские макросы, макросы низкоуровневого обнаружения. Это поле задает CURLOPT_HTTPHEADER [en] опцию cURL.</p>
<p><i>Требуемые коды состояния</i></p>	<p>Список ожидаемых кодов состояния HTTP. Если Zabbix получает код не из списка, то элемент данных станет неподдерживаемым. Если поле не заполнено, то проверка не производится. Например: 200,201,210-299 Поддерживаемые макросы в этом списке: пользовательские макросы, макросы низкоуровневого обнаружения. Это поле задает CURLINFO_RESPONSE_CODE [en] опцию cURL.</p>
<p><i>Следовать перенаправлениям</i></p>	<p>Отметьте для следования по HTTP перенаправлениям. Это поле задает CURLOPT_FOLLOWLOCATION [en] опцию cURL.</p>
<p><i>Режим получения</i></p>	<p>Выберите часть ответа, которую необходимо получить: Тело - только тело Заголовки - только заголовки Тело и заголовки - тело и заголовки</p>
<p><i>Конвертация в JSON</i></p>	<p>Заголовки сохраняются в виде пар атрибутов и значений под ключем "header". Если встречается 'Content-Type: application/json', тогда тело сохраняется в виде объекта, в противном случае оно сохраняется в виде строки, например:</p> <pre>{ "header": { "<key>": "<value>", "<key2>": "<value>" }, "body": <body> }</pre>
<p><i>HTTP прокси</i></p>	<p>Вы можете указать необходимый HTTP прокси, следуя следующему формату: <code>http://[имя пользователя[:пароль]]@[прокси.myscompany.com[:порт]]</code> По умолчанию будет использоваться порт 1080. Если указан, прокси заменит переменные окружения связанные с прокси такие как http_proxy, HTTPS_PROXY. Если не указан, переменные окружения не будут заменены. Введённое значение передается "как есть", проверка правильности не производится. Вы также можете указать адрес SOCKS прокси. Если вы укажете ошибочный протокол, подключение провалится и элемент данных станет неподдерживаемым. Если протокол не указан, прокси будет считаться HTTP прокси. <i>Примечание:</i> Для HTTP прокси поддерживается только простая аутентификация. Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, пользовательские макросы, макросы низкоуровневого обнаружения. Это поле задает CURLOPT_PROXY [en] опцию cURL.</p>

<p>HTTP Аутентификация</p>	<p>Тип аутентификации: Пусто - без использования аутентификации. Простая аутентификация - с использованием простой аутентификации. NTLM аутентификация - с использованием NTLM (Windows NT LAN Manager) аутентификации. Выбрав какой-либо метод аутентификации, будут доступны доступны два дополнительных поля для ввода имени пользователя и пароля, в которых поддерживаются пользовательские макросы и макросы низкоуровневого обнаружения. Это поле задает CURLOPT_HTTPAUTH [en] опцию cURL.</p>
<p>Проверка SSL узла</p>	<p>Отметьте для верификации SSL сертификата веб-сервера. Сертификат сервера будет автоматически взят из места центра сертификации (CA) всей системы. Вы можете перезаписать расположение CA файлов, используя параметр конфигурации SSLCAlocation в Zabbix сервере или прокси. Это поле задает CURLOPT_SSL_VERIFYPEER [en] опцию cURL.</p>
<p>Проверка SSL хоста</p>	<p>Отметьте для верификации, что поле <i>Common Name</i> или поле <i>Subject Alternate Name</i> сертификата веб-сервера совпадают. Это поле задает CURLOPT_SSL_VERIFYHOST [en] опцию cURL.</p>
<p>Файл SSL сертификата</p>	<p>Имя файла SSL сертификата для аутентификации клиента. Файл сертификата должен быть в формате PEM¹. Если файл сертификата также содержит и приватный ключ, оставьте поле Файл SSL ключа пустым. Если ключ зашифрован, укажите пароль в поле Пароль к SSL ключу. Папка, содержащая этот файл указывается в параметре конфигурации SSLCertLocation Zabbix сервера или прокси. Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, пользовательские макросы, макросы низкоуровневого обнаружения. Это поле задает CURLOPT_SSLCERT [en] опцию cURL.</p>
<p>Файл SSL ключа</p>	<p>Имя файла приватного SSL ключа, который используется для аутентификации клиента. Файл приватного ключа должен быть в формате PEM¹. Папка, содержащая этот файл указывается в параметре конфигурации SSLKeyLocation Zabbix сервера или прокси. Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, пользовательские макросы, макросы низкоуровневого обнаружения. Это поле задает CURLOPT_SSLKEY [en] опцию cURL.</p>
<p>Пароль к SSL ключу</p>	<p>Пароль к файлу приватного ключа. Поддерживаемые макросы: пользовательские макросы, макросы низкоуровневого обнаружения. Это поле задает CURLOPT_KEYPASSWD [en] опцию cURL.</p>
<p>Активировать трапы</p>	<p>При наличии этой опции, элемент данных будет работать как траппер элемент данных и будет принимать данные, которые отправляются на этот элемент данных при помощи Zabbix sender или с использованием протокола Zabbix sender.</p>

<i>Разрешенные хосты</i>	<p>Видимо только, если выбрана опция <i>Активировать трапы</i>. Список разделенных запятыми IP адресов, опционально в CIDR представлении, или имена узлов сети. Если задано, входящие подключения будут приниматься только с перечисленных здесь узлов сети. Если включена поддержка IPv6, тогда '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' обрабатываются одинаково и '::/0' будет разрешать любые IPv4 или IPv6 адреса. '0.0.0.0/0' можно использовать, чтобы разрешить любые IPv4 адреса. Обратите внимание, что "IPv4-совместимые IPv6 адреса" (0000::/96 префикс) поддерживаются, но являются устаревшими согласно RFC4291.</p> <p>Пример: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.domain В этом поле разрешены пробелы и пользовательские макросы.</p>
--------------------------	---

Если поле *HTTP прокси* оставить пустым, можно воспользоваться другим способом указать HTTP прокси, для этого необходимо задать переменные окружения.

Для HTTP проверок - укажите переменную окружения **http_proxy** для пользователя Zabbix сервера. Например, `http_proxy=http://proxy_ip:proxy_port`.

Для HTTPS проверок - укажите переменную окружения **HTTPS_PROXY**. Например, `HTTPS_PROXY=http://proxy_ip:proxy_port`. Более подробную информацию можно получить, выполнив в shell команду `# man curl`.

[1] Zabbix поддерживает файлы сертификатов и приватных ключей только в PEM формате. В случае, если у вас имеются данные сертификата и приватного ключа в формате файла PKCS #12 (обычно используется с расширением *.p12 или *.pfx), вы можете сгенерировать из них PEM файл, используя следующие команды:

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

Примеры

Пример 1

Отправка простых GET запросов на получение данных с таких сервисов, как Elasticsearch:

- Создайте GET элемент данных с URL: `localhost:9200/?pretty`
- Обратите внимание на ответ:

```
{
  "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"
}

```

- Теперь извлеките номер версии, используя шаг предварительной обработки JSONPath: `$.version.number`

Пример 2

Отправка простых POST запросов на получение данных с таких сервисов, как Elasticsearch:

- Создайте POST элемент данных с URL:
http://localhost:9200/str/values/_search?scroll=10s
- Настройте следующее тело POST, чтобы получать загрузку процессора (среднее значение за 1 по каждому ядру)

```

{
  "query": {
    "bool": {
      "must": [{
        "match": {
          "itemid": 28275
        }
      }],
      "filter": [{
        "range": {
          "clock": {
            "gt": 1517565836,
            "lte": 1517566137
          }
        }
      ]
    }
  }
}

```

- Получено:

```

{
  "_scroll_id":
  "DnF1ZXJ5VGhlbkZldGNoBQAAAAAAAAAAkFlRMlZBWS1UU1pxTmdEeGVwQjRBTFEAAAAAAAAAAJRZ
  ZUTJWQVktVFNaCU5nRHhlcEI0QUxRAAAAAAAAAACYWwVEyVkFZLVRTWnFOZ0R4ZXBCNEFMUQAAAAA
  AAAAAFlRMlZBWS1UU1pxTmdEeGVwQjRBTFEAAAAAAAAAAKBZZUTJWQVktVFNaCU5nRHhlcEI0QUx

```

```
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
      }
    }]
  }
}
```

- Теперь используйте шаг предварительной обработки JSONPath, чтобы получить значение элемента данных: `$.hits.hits[0]._source.value`

Пример 3

Проверка, доступен ли Zabbix API с использованием [apiinfo.version](#).

- Настройка элемента данных:

Item Preprocessing

* Name

Type

* Key

* URL

Query fields

Name	Value
Add	

Request type

Timeout

Request body type

Request body

```
{
  "jsonrpc": "2.0",
  "method": "apiinfo.version",
  "params": [],
  "id": 1
}
```

Headers

Name	Value
<input type="text" value="Content-Type"/>	<input type="text" value="application/json-rpc"/>
Add	

Required status codes

Follow redirects

Retrieve mode

Обратите внимание, на использование POST метода с JSON данными, с настройкой заголовков в запросе и получением только заголовков:

- Предварительная обработка значений элемента данных при помощи регулярного выражения, чтобы получить HTTP код:

Item Preprocessing

Preprocessing steps

Name	Parameters
<input type="text" value="Regular expression"/>	<input type="text" value="HTTPV1.1 ([0-9]+)"/> <input type="text" value="1"/>
Add	

- Проверка результата в *Последние данные*:

The screenshot shows the 'Latest data' interface in Zabbix. At the top, there is a 'Filter' section with three search boxes: 'Host groups' (containing 'type here to search'), 'Hosts' (containing 'Zabbix server' and 'nginx'), and 'Application' (containing 'type here to search'). There are 'Select' buttons for each. To the right, the 'Name' field contains 'Check Zabbix API'. Below the filters are checkboxes for 'Show items without data' and 'Show details', and 'Apply' and 'Reset' buttons.

<input type="checkbox"/>	Host	Name	Last check	Last value	Change
<input type="checkbox"/>	Zabbix server	- other - (1 item)			
<input type="checkbox"/>		Check Zabbix API version	2018-05-16 23:50:34	OK (200)	Graph

Пример 4

Получение информации о погоде при помощи подключения к публичному сервису Openweathermap.

- Настройка основного элемента данных для массового сбора данных в одном JSON:

Item
Preprocessing

Parent items [Template Weather](#)

* Name

Type

* Key

* URL Pa

Name	Value
<input type="text" value="units"/>	<input type="text" value="metric"/>
<input type="text" value="lat"/>	<input type="text" value="{SLAT}"/>
<input type="text" value="lon"/>	<input type="text" value="{SLON}"/>
<input type="text" value="APPID"/>	<input type="text" value="{SWEATHER_APIKEY}"/>
<input type="text" value="lang"/>	<input type="text" value="{SWEATHER_LANG}"/>

[Add](#)

Request type ▾

Timeout

Request body type

Request body

Обратите внимание на использование макросов в полях запроса. Обратитесь к [Openweathermap API](#) для получения подробных сведений о том как заполнять их.

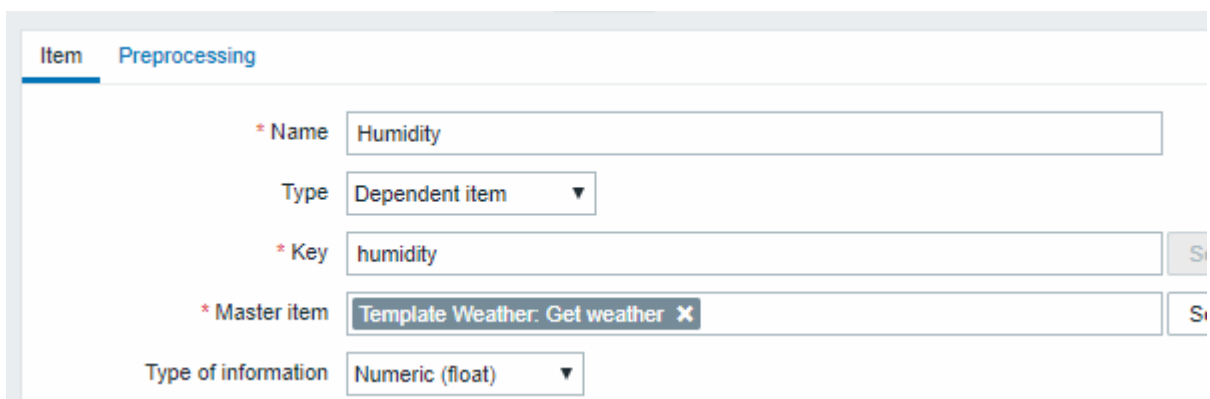
Пример JSON, который вернулся в ответе на HTTP агент:

```
{
  "body": {
    "coord": {
      "lon": 40.01,
      "lat": 56.11
    },
    "weather": [{
      "id": 801,
      "main": "Clouds",
      "description": "few clouds",
      "icon": "02n"
    }],
    "base": "stations",
```

```
"main": {
  "temp": 15.14,
  "pressure": 1012.6,
  "humidity": 66,
  "temp_min": 15.14,
  "temp_max": 15.14,
  "sea_level": 1030.91,
  "grnd_level": 1012.6
},
"wind": {
  "speed": 1.86,
  "deg": 246.001
},
"clouds": {
  "all": 20
},
"dt": 1526509427,
"sys": {
  "message": 0.0035,
  "country": "RU",
  "sunrise": 1526432608,
  "sunset": 1526491828
},
"id": 487837,
"name": "Stavrovo",
"cod": 200
}
}
```

Следующая задача - настройка зависимых элементов данных, которые извлекут данные с JSON.

- Настройка примера зависимого элемента данных для влажности:

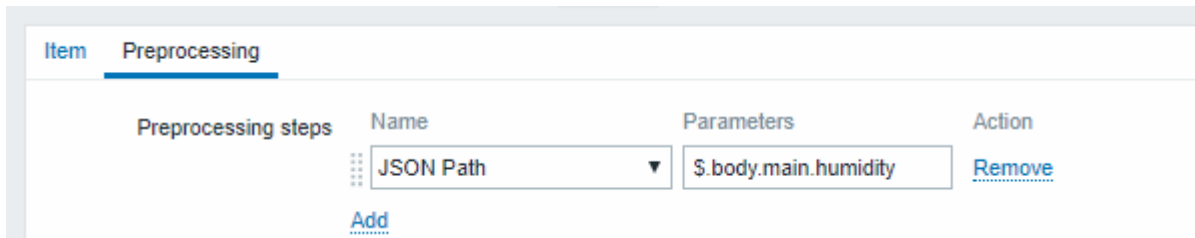


The screenshot shows the 'Preprocessing' tab for an item configuration. The fields are as follows:

- Name:** Humidity
- Type:** Dependent item
- Key:** humidity
- Master item:** Template Weather: Get weather
- Type of information:** Numeric (float)

Другие метрики погоды, такие как 'Температура', добавляются тем же способом.

- Пример предобработки значений зависимого элемента данных с использованием JSONPath:



- Проверка результата данных о погоде в *Последние данные*:

Host	Name	Inter...	History	Trends	Type	Last check	Last value
Weather (8 items)							
<input type="checkbox"/>	Get weather get_weather.http	10m	1d		HTTP agent	2018-05-17 01:23:45	{'body': {'coord': {'lon...
<input type="checkbox"/>	Get weather HTTP response code get_weather.http_code		7d	0	Depende...	2018-05-17 01:23:45	OK (200)
<input type="checkbox"/>	Humidity humidity		90d	365d	Depende...	2018-05-17 01:23:45	66 %
<input type="checkbox"/>	Temperature temp		90d	365d	Depende...	2018-05-17 01:23:45	15.14 C
<input type="checkbox"/>	Weather weather		90d		Depende...	2018-05-17 01:23:45	Clouds
<input type="checkbox"/>	Weather condition id weather.condition.id		7d	0	Depende...	2018-05-17 01:23:45	801
<input type="checkbox"/>	Weather description weather.description		90d		Depende...	2018-05-17 01:23:45	few clouds
<input type="checkbox"/>	Wind speed wind.speed		90d	365d	Depende...	2018-05-17 01:23:45	1.86 m/s

Пример 5

Подключение в странице состояния Nginx и получения его метрик за один запрос.

- Настройте Nginx, следуя [официальному руководству](#).
- Настройте основной элемент данных для массового сбора данных:

The screenshot shows the configuration for an item named 'Get NGINX status page'. The configuration includes:

- Name:** Get NGINX status page
- Type:** HTTP agent
- Key:** get_nginx
- URL:** http://{HOST.CONN}/nginx_status
- Query fields:** A table with columns 'Name' and 'Value'. One field is defined with 'name' as the key and 'value' as the value.
- Request type:** GET
- Timeout:** 3s
- Request body type:** Raw data (selected), JSON data, XML data
- Request body:** (empty)

Пример вывода состояния Nginx stub:

```
Active connections: 1 Active connections:  
server accepts handled requests  
52 52 52  
Reading: 0 Writing: 1 Waiting: 0
```

Следующая задача - настроить зависимые элементы данных, которые извлекают необходимые данные.

- Настройка примера зависимого элемента данных для запросов в секунду:

The screenshot shows the configuration for a dependent item named 'Client requests per second'. The configuration includes:

- Name:** Client requests per second
- Type:** Dependent item
- Key:** nginx_requests_rps
- Master item:** Template App Nginx HTTP: Get Nginx stub status
- Type of information:** Numeric (unsigned)

- Настройка предобработки значений зависимого элемента данных при помощи регулярного выражения:

Item Preprocessing

Preprocessing steps	Name	Parameters
<input type="checkbox"/>	Regular expression	server accepts handled requests's+([0-9]+) ([0-9]+) ([0-9]+)
<input type="checkbox"/>	Change per second	13

[Add](#)

- Проверьте полученный результат от модуля stub в *Последние данные*:

Host	Name	Last check	Last value
nginx	Nginx (8 Items)		
<input type="checkbox"/>	Accepted client connections	2018-05-18 17:54:53	568
<input type="checkbox"/>	Active connections	2018-05-18 17:54:53	1
<input type="checkbox"/>	Client requests per second	2018-05-18 17:54:53	0 rps
<input checked="" type="checkbox"/>	Get Nginx stub status	2018-05-18 17:54:53	HTTP/1.1 200 OK Se...
<input type="checkbox"/>	Handled connections per second	2018-05-18 17:54:53	0
<input type="checkbox"/>	Reading	2018-05-18 17:54:53	0
<input type="checkbox"/>	Waiting	2018-05-18 17:54:53	0
<input type="checkbox"/>	Writing	2018-05-18 17:54:53	1

From: <https://www.zabbix.com/documentation/4.2/> - **Zabbix Documentation 4.2**

Permanent link: <https://www.zabbix.com/documentation/4.2/ru/manual/config/items/itemtypes/http>

Last update: **2018/10/01 09:42**

