



ZABBIX 2020
Conference
CHINA

演讲主题

通过Zabbix中的数据预处理 提高效率和灵活性

演讲嘉宾

米宏 宏时数据 项目经理

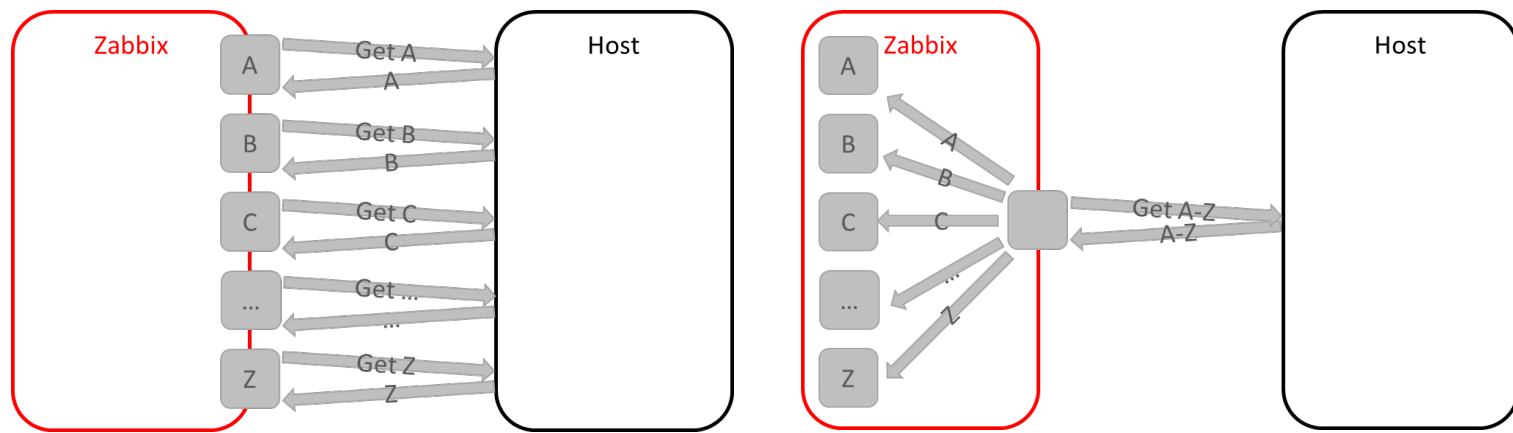
什么是预处理？



预处理允许为接收的监控值定义转换规则。在将值保存到数据库之前，可以进行一个或多个转换。转换按照定义它们的顺序执行。

概述

- ✓ Zabbix支持依赖项，允许在监控项中批量收集和使用采集到的数据
- ✓ 主监控项数据获取成功后，会自动填充从属监控项的值
- ✓ Zabbix预处理选项可以从主监控项中提取相关的监控数据
- ✓ 只有Zabbix Server，Zabbix Proxy才能处理依赖项



预处理相关配置及进程

```
#ps -ef | grep [p]reprocess
zabbix    5537    5489  0 Sep24 ?        00:00:13 /usr/sbin/zabbix_server: preprocessing manager #1 [queued 0, 1
  sec during 5.006046 sec]
zabbix    5538    5489  0 Sep24 ?        00:00:05 /usr/sbin/zabbix_server: preprocessing worker #1 started
zabbix    5539    5489  0 Sep24 ?        00:00:00 /usr/sbin/zabbix_server: preprocessing worker #2 started
zabbix    5540    5489  0 Sep24 ?        00:00:00 /usr/sbin/zabbix_server: preprocessing worker #3 started
```

```
### Option: StartPreprocessors
#       Number of pre-forked instances of preprocessing workers.
#       The preprocessing manager process is automatically started when preprocessor worker is started.
#
# Mandatory: no
# Range: 1-1000
# Default:
# StartPreprocessors=3
```

多种处理方式

TEXT : *Regular expression、Replace、Trim、Right trim、Left Trim*

Structured data : *XML、JSON、CSV*

Arithmetic : *Custom multiplier*

Change : *Simple change、Change per second*

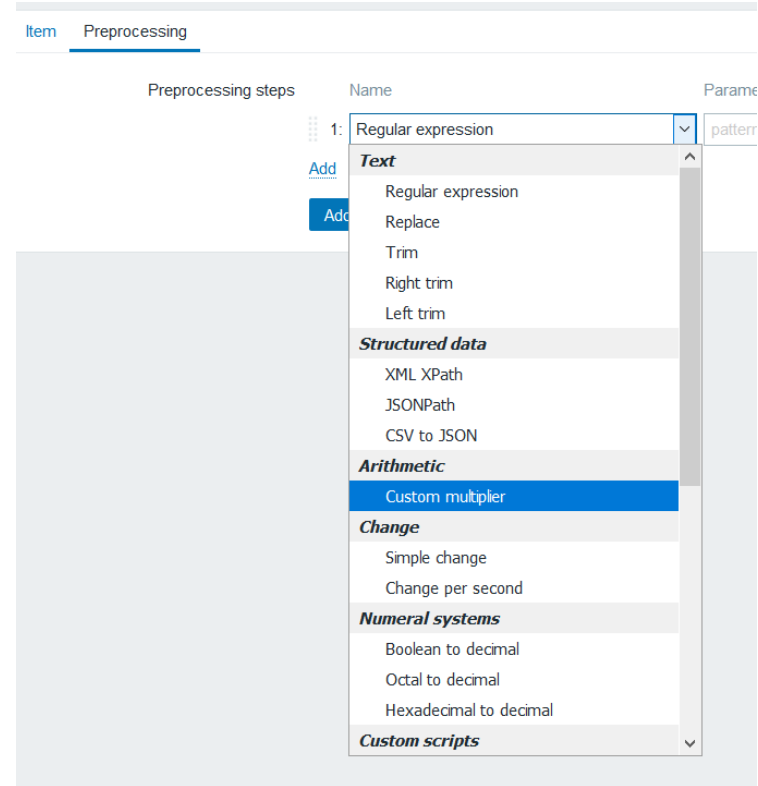
Numeral systems : *Boolean to decimal、Octal to decimal、Hexadecimal to decimal*

Custom scripts : *Javascript*

Validation : *In range、Matches regular expression*

Throttling : *Discard unchanged、Discard unchanged with heartbeat*

Prometheus : *Prometheus pattern、Prometheus to JSON*



兼容正则表达式(PCRE)

PCRE: 将监控项值与正则表达式 <pattern> 匹配并替换输出 <output>

PCRE pattern	Output	Result
Threads_connected.(\d*)	\1	Extracts number of connected threads (e.g. 39)
version\s+ \s+([\d+\.]{2,}\.\d+)\D	\1	Extracts MySQL version number (e.g. 5.5.52)
674.10893.2.30.1.1.8.*STRING: "(.*)"	\1	Extracts part of SNMP trap (e.g. enclosure path redundancy lost)

PCRE 示例

`Threads_connected.(\d*)`

```
Threads_cached 0  
Threads_connected 39  
Threads_created 327561
```



```
Threads_cached 0  
Threads_connected 39  
Threads_created 327561
```

`version\s+|\s+([\d+\.]{2,}\.\d+)\D`

```
| userstat      | OFF      |  
| version       | 5.5.52-MariaDB|  
| version_comment | MariaDB Server|
```



```
| userstat      | OFF      |  
| version       | 5.5.52-MariaDB|  
| version_comment | MariaDB Server|
```

`674.10893.2.30.1.1.8.*STRING: "(.*)"`

```
DISMAN-EVENT-MIB::sysUpTimeInstance type=67  
value=Timeticks: (1002197237) 115 days, 23:52:52.37  
SNMPv2-MIB::snmpTrapOID.0 type=6 value=OID:  
SNMPv2-SMI::enterprises.674.10893.2.30.0.2  
SNMPv2-SMI::enterprises.674.10893.2.30.1.1.4  
type=4 value=STRING: "iscsi01"  
SNMPv2-SMI::enterprises.674.10893.2.30.1.1.6  
type=4 value=STRING: "Mar 14, 2018 3:12:57 AM"  
SNMPv2-SMI::enterprises.674.10893.2.30.1.1.7  
type=4 value=STRING: "Enclosure path redundancy lost"  
SNMP-COMMUNITY-MIB::snmpTrapCommunity.0  
type=4 value=STRING: "IKhkd^%^fds3"
```



```
DISMAN-EVENT-MIB::sysUpTimeInstance type=67  
value=Timeticks: (1002197237) 115 days, 23:52:52.37  
SNMPv2-MIB::snmpTrapOID.0 type=6 value=OID:  
SNMPv2-SMI::enterprises.674.10893.2.30.0.2  
SNMPv2-SMI::enterprises.674.10893.2.30.1.1.4  
type=4 value=STRING: "iscsi01"  
SNMPv2-SMI::enterprises.674.10893.2.30.1.1.6  
type=4 value=STRING: "Mar 14, 2018 3:12:57 AM"  
SNMPv2-SMI::enterprises.674.10893.2.30.1.1.7  
type=4 value=STRING: "Enclosure path redundancy lost"  
SNMP-COMMUNITY-MIB::snmpTrapCommunity.0  
type=4 value=STRING: "IKhkd^%^fds3"
```

JSON PATH

JSON Path: 支持在不使用定制脚本的情况下从JSON中提取数据

示例：

JSON PATH	结果
<code>\$.store.book[0].price</code>	在JSON数据中选择第一本书的价格
<code>\$.store.book[2].title</code>	在JSON数据中选择第三本书的标题
<code>\$.store.bicycle.price</code>	在JSON数据中选择自行车的价格

JSON PATH

示例：

```
$.store.book[0].price
```

```
{ "store": {  
  "book": [  
    { "category": "reference",  
      "author": "Nigel Rees",  
      "title": "Sayings of the Century",  
      "price": 8.95  
    },  
    { "category": "fiction",  
      "author": "Evelyn Waugh",  
      "title": "Sword of Honour",  
      "price": 12.99  
    },  
    { "category": "fiction",  
      "author": "J. R. R. Tolkien",  
      "title": "The Lord of the Rings",  
      "isbn": "0-395-19395-8",  
      "price": 22.99  
    }  
  ],  
  "bicycle": {  
    "color": "red",  
    "price": 19.95  
  }  
}
```



```
{ "store": {  
  "book": [  
    { "category": "reference",  
      "author": "Nigel Rees",  
      "title": "Sayings of the Century",  
      "price": 8.95  
    },  
    { "category": "fiction",  
      "author": "Evelyn Waugh",  
      "title": "Sword of Honour",  
      "price": 12.99  
    },  
    { "category": "fiction",  
      "author": "J. R. R. Tolkien",  
      "title": "The Lord of the Rings",  
      "isbn": "0-395-19395-8",  
      "price": 22.99  
    }  
  ],  
  "bicycle": {  
    "color": "red",  
    "price": 19.95  
  }  
}
```

创建一个HTTP agent监控项

Items

All hosts / test Enabled ZBX SNMP JMX IPMI Applications Items 2 Triggers Graphs Discovery rules Web scenarios

Item Preprocessing

* Name

Type

* Key

* URL

Query fields

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

[Add](#)

Request type

Timeout

Request body type Raw data JSON data XML data

Request body

Headers

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

[Add](#)

* Host interface

Type of information

* Update interval

主监控项获取得监控内容

Value

Log Type : Security

Event Type : Audit Success

Time : 10.12.2012 19:33:24

Event ID : 680

User Name : SYSTEM

Computer : YYY

Event Description : Logon attempt by: MICROSOFT_AUTHENTICATION_PACKAGE_V1_0 Logon account: XXX Source Workstation: YYY Error Code: 0x0

Log Type : Security

Event Type : Audit Success

Time : 10.12.2013 18:33:24

Event ID : 679

User Name : ADMIN

Computer : YYY

Event Description : Logon attempt by: MICROSOFT_AUTHENTICATION_PACKAGE_V1_0 Logon account: XXX Source Workstation: YYY Error Code: 0x0

使用正则工具进行测试

正则测试网站：<https://regex101.com/>

The screenshot displays the regex101.com interface. The regular expression `/.*/gm` is entered in the 'REGULAR EXPRESSION' field. The 'TEST STRING' field contains a multi-line log entry. The results show 41 matches across 272 steps. The 'EXPLANATION' panel details the components of the regex: `/` for the forward slash, `(.*)` for the capturing group, and `gm` for the global and multiline flags. The 'MATCH INFORMATION' panel shows two matches, both with a full match of `Log Type : Security`. The 'QUICK REFERENCE' panel provides a list of common tokens and their corresponding regex patterns.

regular expressions 101

@regex101 donate sponsor contact bug reports & feedback wiki

SAVE & SHARE

Save Regex `ctrl+s`

FLAVOR

- PCRE (PHP) ✓
- ECMAScript (JavaScript)
- Python
- Golang

FUNCTION

- ★ Match ✓
- Substitution
- Unit Tests

TOOLS

- Code Generator
- Regex Debugger

REGULAR EXPRESSION `/.*/gm` 41 matches, 272 steps (~1ms)

TEST STRING

```
Log Type : Security
Event Type : Audit Success
Time : 10.12.2012 19:33:24
Event ID : 680
User Name : SYSTEM
Computer : YYY
Event Description : Logon attempt by: MICROSOFT_AUTHENTICATION_PACKAGE_V1_0 Logon
account: XXX Source Workstation: YYY Error Code: 0x0
Log Type : Security
Event Type : Audit Success
Time : 10.12.2013 18:33:24
Event ID : 679
User Name : ADMIN
```

EXPLANATION

- `/` matches any character (except for line terminators)
- `(.*)` Quantifier — Matches between zero and unlimited times, as many times as possible, giving back as needed (greedy)
- Global pattern flags**
 - `g` modifier: global. All matches (don't return after first match)

MATCH INFORMATION

Match 1

- Full match 0-19 Log Type : Security
- Group 1. 0-19 Log Type : Security

Match 2

- Full match 19-19

QUICK REFERENCE

Search reference

- All Tokens
- ★ Common Tokens ✓
- General Tokens
- Anchors
- Meta Sequences

- A single character of: a, b or c `[abc]`
- A character except: a, b or c `[^abc]`
- A character in the range: a-z `[a-z]`
- A character not in the range... `[^a-z]`
- A character in the range: ... `[a-zA-Z]`
- Any single character `.`

SPONSOR

获取第一行数据

Item Preprocessing

* Name

Type

* Key

* Master item

Type of information

* History storage period

Item Preprocessing

Preprocessing steps	Name	Parameters	Custom on fail	Actions
1:	<input type="text" value="Regular expression"/>	<input type="text" value="(.*)"/>	<input type="text" value="10"/>	<input type="checkbox"/> Test Remove

[Add](#) [Test all steps](#)

测试正则匹配

The screenshot displays a web-based regex testing tool. The interface is divided into several sections:

- REGULAR EXPRESSION:** The input field contains the pattern `/ User Name : (.*) / gm`. A red box highlights this field.
- TEST STRING:** A list of log entries is shown. The entry `User Name : SYSTEM` is highlighted with a red box, indicating a successful match.
- EXPLANATION:** A detailed breakdown of the regex components:
 - `/ User Name : (.*) / gm`
 - `User Name :` matches the characters `User Name :` literally (case sensitive)
 - 1st Capturing Group `(.*)`**
 - `.` matches any character (except for line terminators)
 - Quantifier** — Matches between **zero** and **unlimited** times, as many times as possible, giving back as needed (greedy)
 - Global pattern flags**
- MATCH INFORMATION:** Shows two matches:
 - Match 1:** Full match: 93-111, User Name : SYSTEM. Group 1: 105-111, SYSTEM. A red box highlights the group 1 match.
 - Match 2:** Full match: 358-375, User Name : ADMIN.
- QUICK REFERENCE:** A list of common regex tokens and their meanings, such as `[abc]` for a single character, `[^abc]` for a character except, `[a-z]` for a character in a range, `[^a-z]` for a character not in a range, `[a-zA-Z]` for a character in a range, and `.` for any single character.

获取登录用户名

Item Preprocessing

* Name

Type

* Key

* Master item

Type of information

* History storage period

Item Preprocessing

Preprocessing steps	Name	Parameters	Custom on fail	Actions
1:	<input type="text" value="Regular expression"/>	<input type="text" value="User Name : (*)"/>	<input type="text" value="1"/>	<input type="checkbox"/> Test Remove

[Add](#) [Test all steps](#)

使用正则工具进行测试

regular-expressions.com

@regex101 donate sponsor contact bug reports & feedback wiki

1 match, 1241 steps (~6ms)

REGULAR EXPRESSION: `/User Name(.|\n)*User Name : (.*)/gm`

TEST STRING

```
Time : 10.12.2012 19:33:24
Event ID : 680
User Name : SYSTEM
Computer : YYY
Event Description : Logon attempt by: MICROSOFT_AUTHENTICATION_PACKAGE_V1_0 Logon
account: XXX Source Workstation: YYY Error Code: 0x0
Log Type : Security
Event Type : Audit Success
Time : 10.12.2013 18:33:24
Event ID : 679
User Name : ADMIN
Computer : YYY
Event Description : Logon attempt by: MICROSOFT_AUTHENTICATION_PACKAGE_V1_0 Logon
```

EXPLANATION

▼ /User Name(.|\n)*User Name : (.*)/gm
User Name matches the characters `User Name` literally (case sensitive)

▼ 1st Capturing Group `(.|\n)*`
Quantifier — Matches between **zero** and **unlimited** times, as many times as possible, giving back as needed (*greedy*)
A repeated capturing group will only capture the last iteration. Put a capturing group around the repeated group to capture all iterations or use a non-capturing group instead if you're not interested in the data

MATCH INFORMATION

Group	Start	End	Match
Group 1.	357	358	
Group 2.	370	375	ADMIN

QUICK REFERENCE

Search reference

- All Tokens
- ★ Common Tokens ✓
- General Tokens
- ⚓ Anchors
- Meta Sequences

- A single character of: a, b or c `[abc]`
- A character except: a, b or c `[^abc]`
- A character in the range: a-z `[a-z]`
- A character not in the range... `[^a-z]`
- A character in the range: ... `[a-zA-Z]`
- Any single character `.`

SPONSOR

自帶測試工具

Item Preprocessing

Preprocessing steps	Name	Parameters	Custom on fail	Actions	
1:	Regular expression	User Name(\n)*User Name : (*)	12	<input type="checkbox"/>	Test Remove

[Add](#)

[Add](#) [Test](#) [Cancel](#)

[Test all steps](#)

Test item

Value [✎](#) Time

Previous value [✎](#) Prev. time

End of line sequence

Preprocessing steps

Name	Result
1: Regular expression	ADMIN

[Test](#) [Cancel](#)

创建子监控项

Item Preprocessing

* Name

Type

* Key

* Master item

Type of information

* History storage period

Show value

New application

Applications

创建一个获取JSON监控项

Item Preprocessing

* Name

Type

* Key

* URL

Query fields

Name	Value
------	-------

Value

```
{
  "books": [
    {
      "category": "reference",
      "author": "Nigel Rees",
      "title": "Sayings of the Century",
      "price": 8.95,
      "id": 1
    },
    {
      "category": "fiction",
      "author": "Evelyn Waugh",
      "title": "Sword of Honour",
      "price": 12.99,
      "id": 2
    }
  ],
}
```

解析JSON

JSON在线编辑器

```
{
  "books": [
    {
      "category": "reference",
      "author": "Nigel Rees",
      "title": "Sayings of the Century",
      "price": 8.95,
      "id": 1
    },
    {
      "category": "fiction",
      "author": "Evelyn Waugh",
      "title": "Sword of Honour",
      "price": 12.99,
      "id": 2
    },
    {
      "category": "fiction",
      "author": "Herman Melville",
```

object (5)

- books [4]
 - 0 {5}
 - category: reference
 - author: Nigel Rees
 - title: Sayings of the Century
 - price: 8.95
 - id: 1
 - 1 {5}
 - 2 {6}
 - 3 {6}
 - services {3}

创建JSON子监控项

Item Preprocessing

* Name

Type

* Key

* Master item

Type of information

* History storage period

Item Preprocessing

Preprocessing steps	Name	Parameters	Custom on fail	Actions
1:	<input type="text" value="JSONPath"/>	<input type="text" value="\$..books[0].title"/>	<input type="checkbox"/>	Test Remove

[Add](#)

[Test all steps](#)

解析JSON

Item Preprocessing

* Name

Type

* Key

* Master item

Type of information

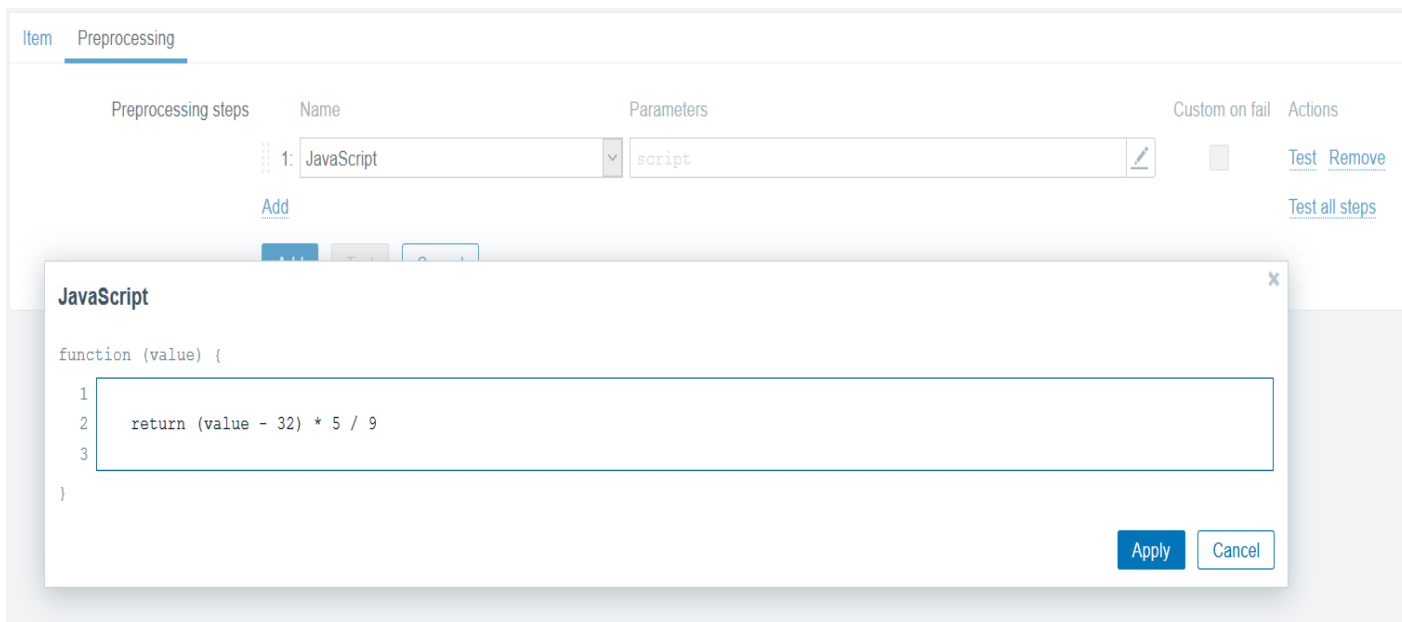
Item Preprocessing

Preprocessing steps	Name	Parameters
1:	<input type="text" value="JSONPath"/>	<input type="text" value="\$services.bookbinding.price"/>

[Add](#)

Javascript预处理

Javascript预处理是通过调用带有单一参数“value”和用户提供的函数体的Javascript函数来完成的。预处理步骤的结果是从这个函数返回的值。



The screenshot shows the Zabbix Preprocessing configuration interface. A modal window titled "JavaScript" is open, allowing the user to define a function. The function body contains the following code:

```
function (value) {  
1  
2   return (value - 32) * 5 / 9  
3  
}
```

The background interface shows a table of preprocessing steps with columns for Name, Parameters, Custom on fail, and Actions. The first step is named "JavaScript" and has the parameter "script".

Javascript测试数据

Timestamp	Value
2020-09-25 04:08:02	AAA 000
	BBB 111
	CCC 222
	DDD 333
2020-09-25 04:07:57	AAA 000
	BBB 111
	CCC 222
	DDD 333

配置依赖项

Item Preprocessing

* Name

Type

* Key

* Master item

Type of information

* History storage period Storage period

Item Preprocessing

Preprocessing steps	Name	Parameters	Custom on fail	Actions
1:	<input type="text" value="JavaScript"/>	<input type="text" value="script"/>	<input type="checkbox"/>	Test Remove

[Add](#)

[Test all steps](#)

生成JSON

```
lines = value.split("\n")
var result_array = new Array()
var i = 0
lines.forEach(function(line)
{
    if (line){
        result_array[i] = '{"NAME":'+'''+line.split(' ')[0]+'''+', "NUMBER":'+'''+line.split('
')[1]+'''+'}'
        i++
    }
})
return '{"data":['+result_array+'}]'
```

处理后生成的值

Value

```
{"data":[{"NAME":"AAA","NUMBER":"000"}, {"NAME":"BBB","NUMBER":"111"}, {"NAME":"CCC","NUMBER":"222"}, {"NAME":"DDD","NUMBER":"333"}]}
```

```
{"data":[{"NAME":"AAA","NUMBER":"000"}, {"NAME":"BBB","NUMBER":"111"}, {"NAME":"CCC","NUMBER":"222"}, {"NAME":"DDD","NUMBER":"333"}]}
```

```
{"data":[{"NAME":"AAA","NUMBER":"000"}, {"NAME":"BBB","NUMBER":"111"}, {"NAME":"CCC","NUMBER":"222"}, {"NAME":"DDD","NUMBER":"333"}]}
```

```
{"data":[{"NAME":"AAA","NUMBER":"000"}, {"NAME":"BBB","NUMBER":"111"}, {"NAME":"CCC","NUMBER":"222"}, {"NAME":"DDD","NUMBER":"333"}]}
```

```
{"data":[{"NAME":"AAA","NUMBER":"000"}, {"NAME":"BBB","NUMBER":"111"}, {"NAME":"CCC","NUMBER":"222"}, {"NAME":"DDD","NUMBER":"333"}]}
```

```
{"data":[{"NAME":"AAA","NUMBER":"000"}, {"NAME":"BBB","NUMBER":"111"}, {"NAME":"CCC","NUMBER":"222"}, {"NAME":"DDD","NUMBER":"333"}]}
```

```
{"data":[{"NAME":"AAA","NUMBER":"000"}, {"NAME":"BBB","NUMBER":"111"}, {"NAME":"CCC","NUMBER":"222"}, {"NAME":"DDD","NUMBER":"333"}]}
```

```
{"data":[{"NAME":"AAA","NUMBER":"000"}, {"NAME":"BBB","NUMBER":"111"}, {"NAME":"CCC","NUMBER":"222"}, {"NAME":"DDD","NUMBER":"333"}]}
```

```
{"data":[{"NAME":"AAA","NUMBER":"000"}, {"NAME":"BBB","NUMBER":"111"}, {"NAME":"CCC","NUMBER":"222"}, {"NAME":"DDD","NUMBER":"333"}]}
```

```
{"data":[{"NAME":"AAA","NUMBER":"000"}, {"NAME":"BBB","NUMBER":"111"}, {"NAME":"CCC","NUMBER":"222"}, {"NAME":"DDD","NUMBER":"333"}]}
```

```
{"data":[{"NAME":"AAA","NUMBER":"000"}, {"NAME":"BBB","NUMBER":"111"}, {"NAME":"CCC","NUMBER":"222"}, {"NAME":"DDD","NUMBER":"333"}]}
```

```
{"data":[{"NAME":"AAA","NUMBER":"000"}, {"NAME":"BBB","NUMBER":"111"}, {"NAME":"CCC","NUMBER":"222"}, {"NAME":"DDD","NUMBER":"333"}]}
```

```
{"data":[{"NAME":"AAA","NUMBER":"000"}, {"NAME":"BBB","NUMBER":"111"}, {"NAME":"CCC","NUMBER":"222"}, {"NAME":"DDD","NUMBER":"333"}]}
```

QUESTIONS?

ZABBIX

The Enterprise class Monitoring Solution for Everyone

www.zabbix.com

ZABBIX 2020
Conference
CHINA

联系我们

Contact us

Zabbix 中国致力于为国内用户提供培训、咨询、以及其他的专业技术支持。也为国内的用户搭建交流学习的平台。



138-1772-0274



china@zabbix.com



www.grandage.cn
www.zabbix.com/cn



上海市徐汇区虹梅路1905号



Zabbix开源社区



Zabbix中国



Zabbix_China



Zabbix_team



Zabbix 开源社区



加入技术交流群

ZABBIX 2020
Conference
CHINA

THANK YOU 😊

