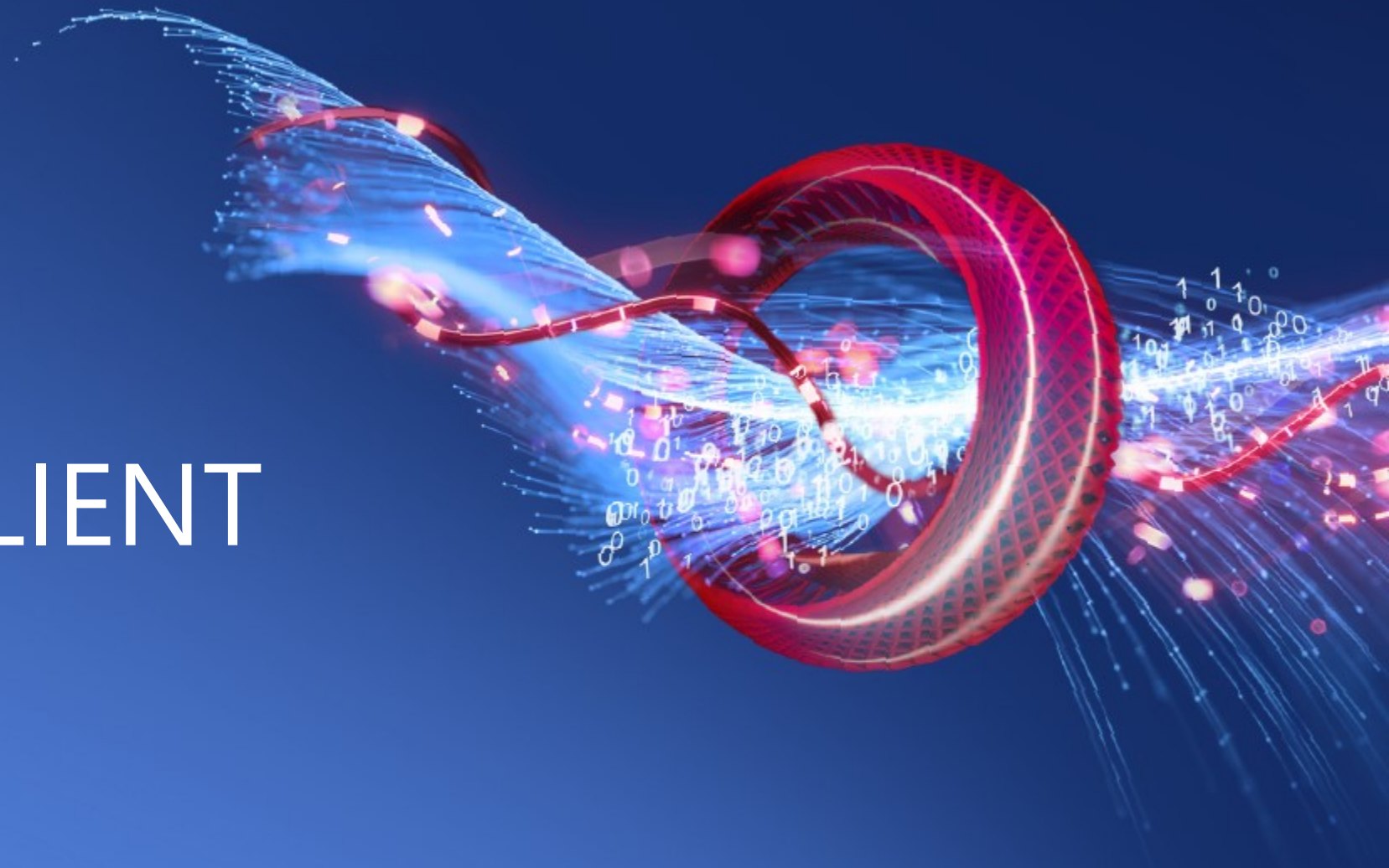
ZABBIX 5.0

ZABBIX API – LEVEL 1

MESSAGE TO SPREAD

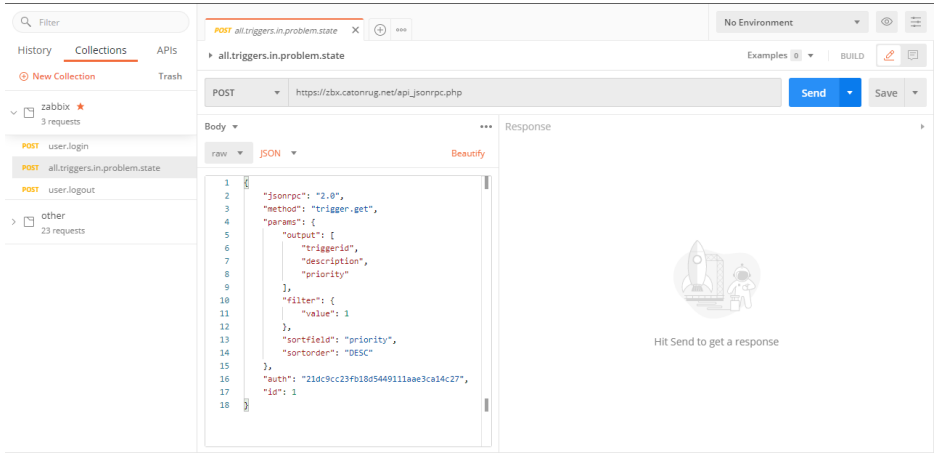
- Everything is easy like copy and paste
- Documentation page is your friend
- Introduce what is possible

REST API CLIENT



GUI to perform API calls

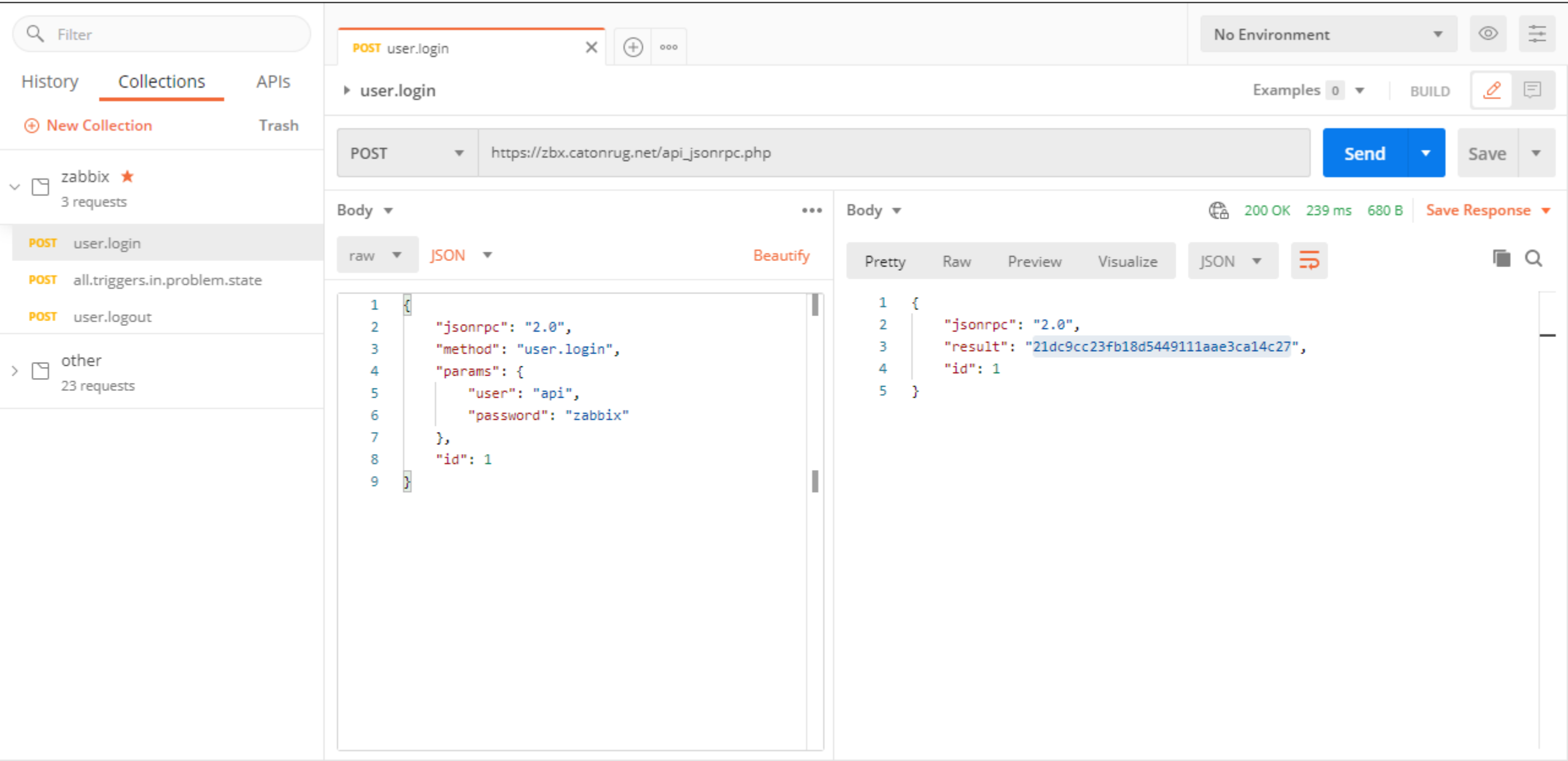
POSTMAN



https://www.postman.com/downloads/

- Available on Windows, Linux, Mac
- Store all your snippets to Google account
- Use official Zabbix documentation page

OBTAIN SESSION TOKEN



PICK A FLAVOR FROM DOCUMENTATION PAGE

Filter

History Collections APIs

New Collection Trash

zabbix ★
3 requests

POST user.login
POST all.triggers.in.problem.state
POST user.logout

other
23 requests

POST all.triggers.in.problem.state

No Environment

all.triggers.in.problem.state

Examples 0 BUILD

POST https://zbx.catonrug.net/api_jsonrpc.php

Send Save

Body

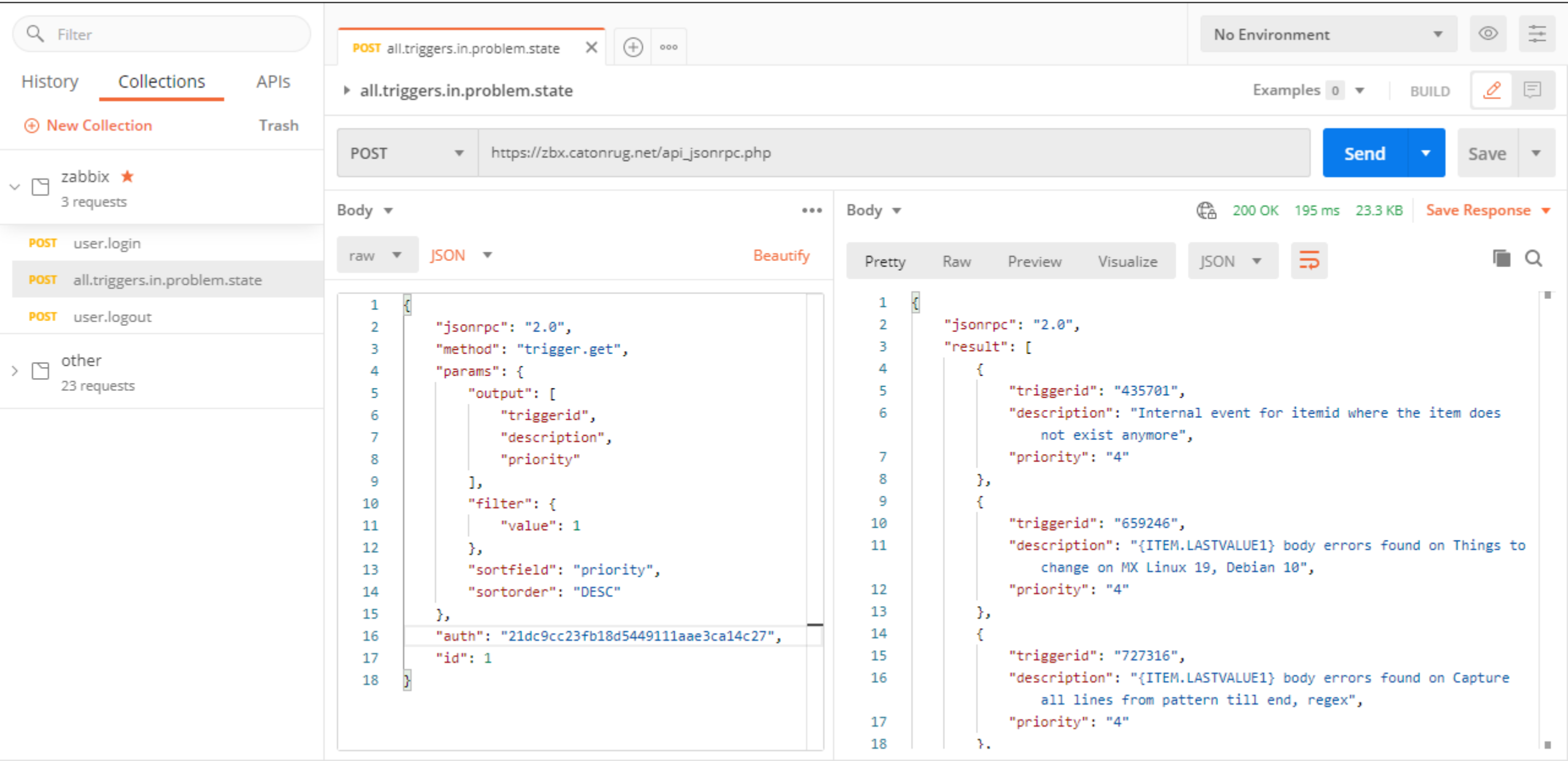
raw JSON Beautify

```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
    "params": {
        "output": [
            "triggerid",
            "description",
            "priority"
        ],
        "filter": {
            "value": 1
        },
        "sortfield": "priority",
        "sortorder": "DESC"
    },
    "auth": "21dc9cc23fb18d5449111aae3ca14c27",
    "id": 1
}
```

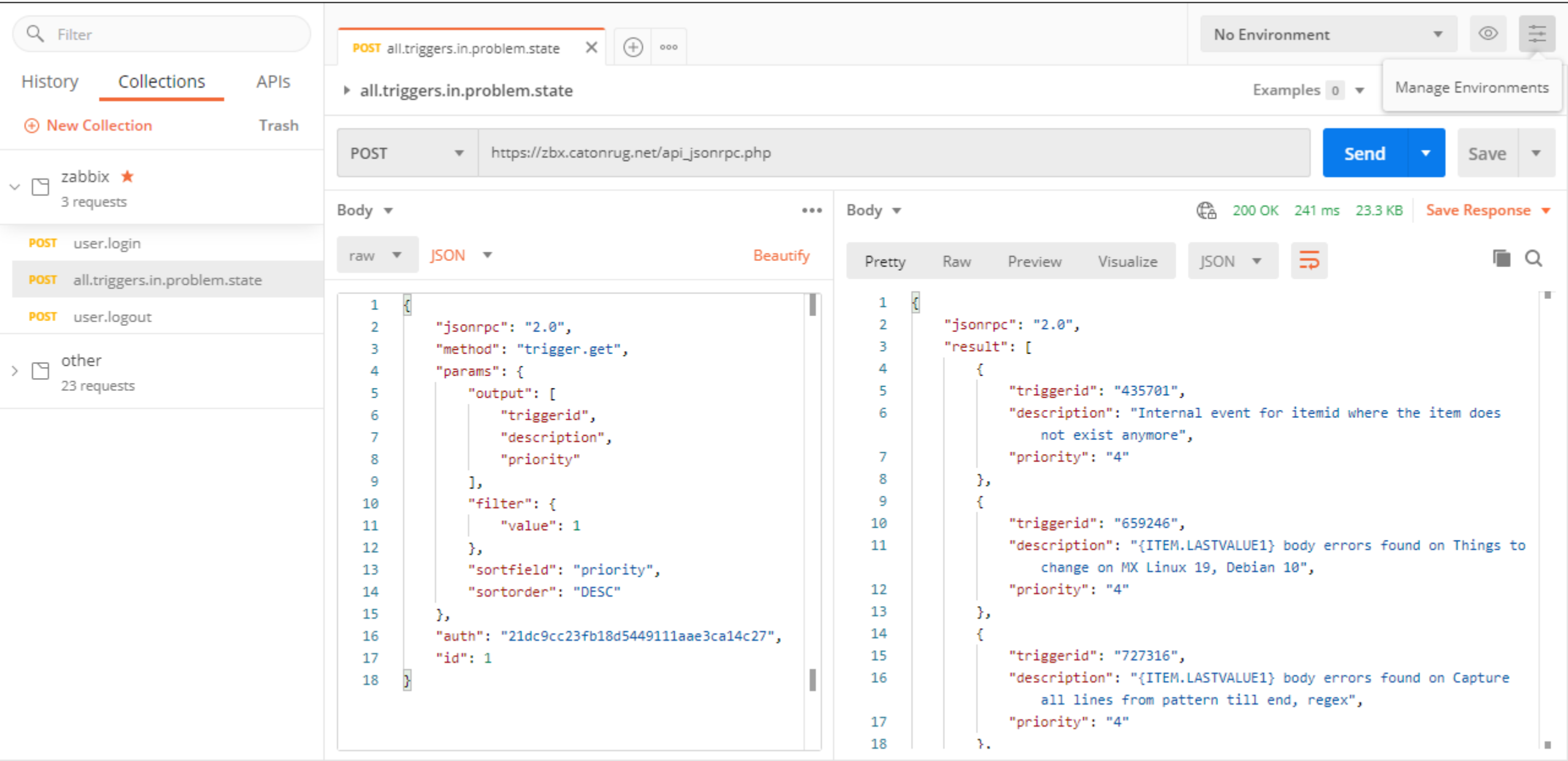
Response

Hit Send to get a response

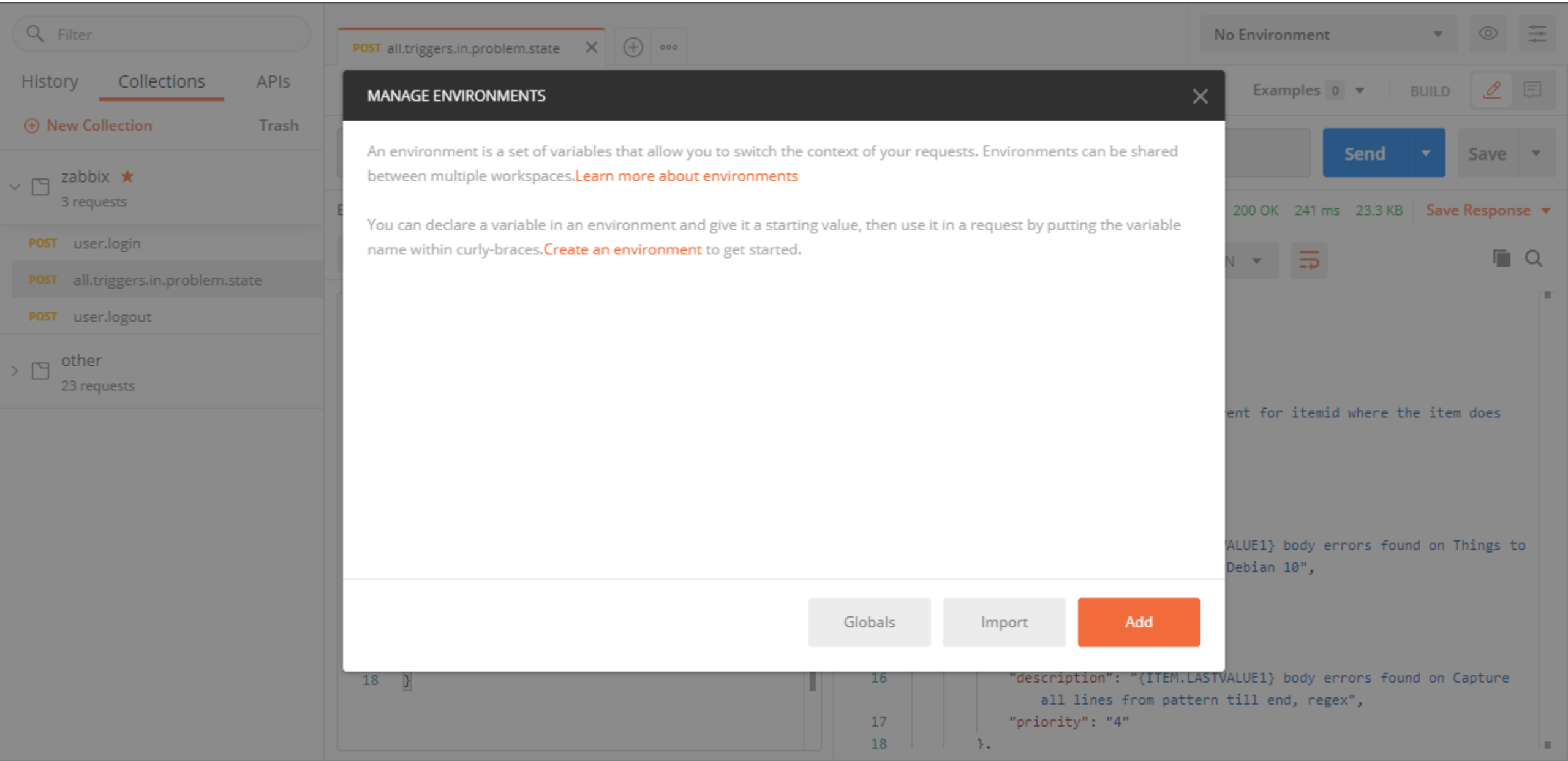
RESPONSE


Filter
History
Collections
APIs
New Collection
Trash
zabbix
3 requests
POST user.login
POST all.triggers.in.problem.state
POST user.logout
other
23 requests
POST all.triggers.in.problem.state
No Environment
all.triggers.in.problem.state
Examples 0
BUILD
POST
https://zbx.catonrug.net/api_jsonrpc.php
Send
Save
Body
raw
JSON
Beautify
{
"jsonrpc": "2.0",
"method": "trigger.get",
"params": {
"output": [
"triggerid",
"description",
"priority"
],
"filter": {
"value": 1
},
"sortfield": "priority",
"sortorder": "DESC"
},
"auth": "21dc9cc23fb18d5449111aae3ca14c27",
"id": 1
}
Body
200 OK
195 ms
23.3 KB
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"jsonrpc": "2.0",
"result": [
{
"triggerid": "435701",
"description": "Internal event for itemid where the item does not exist anymore",
"priority": "4"
},
{
"triggerid": "659246",
"description": "{ITEM.LASTVALUE1} body errors found on Things to change on MX Linux 19, Debian 10",
"priority": "4"
},
{
"triggerid": "727316",
"description": "{ITEM.LASTVALUE1} body errors found on Capture all lines from pattern till end, regex",
"priority": "4"
},

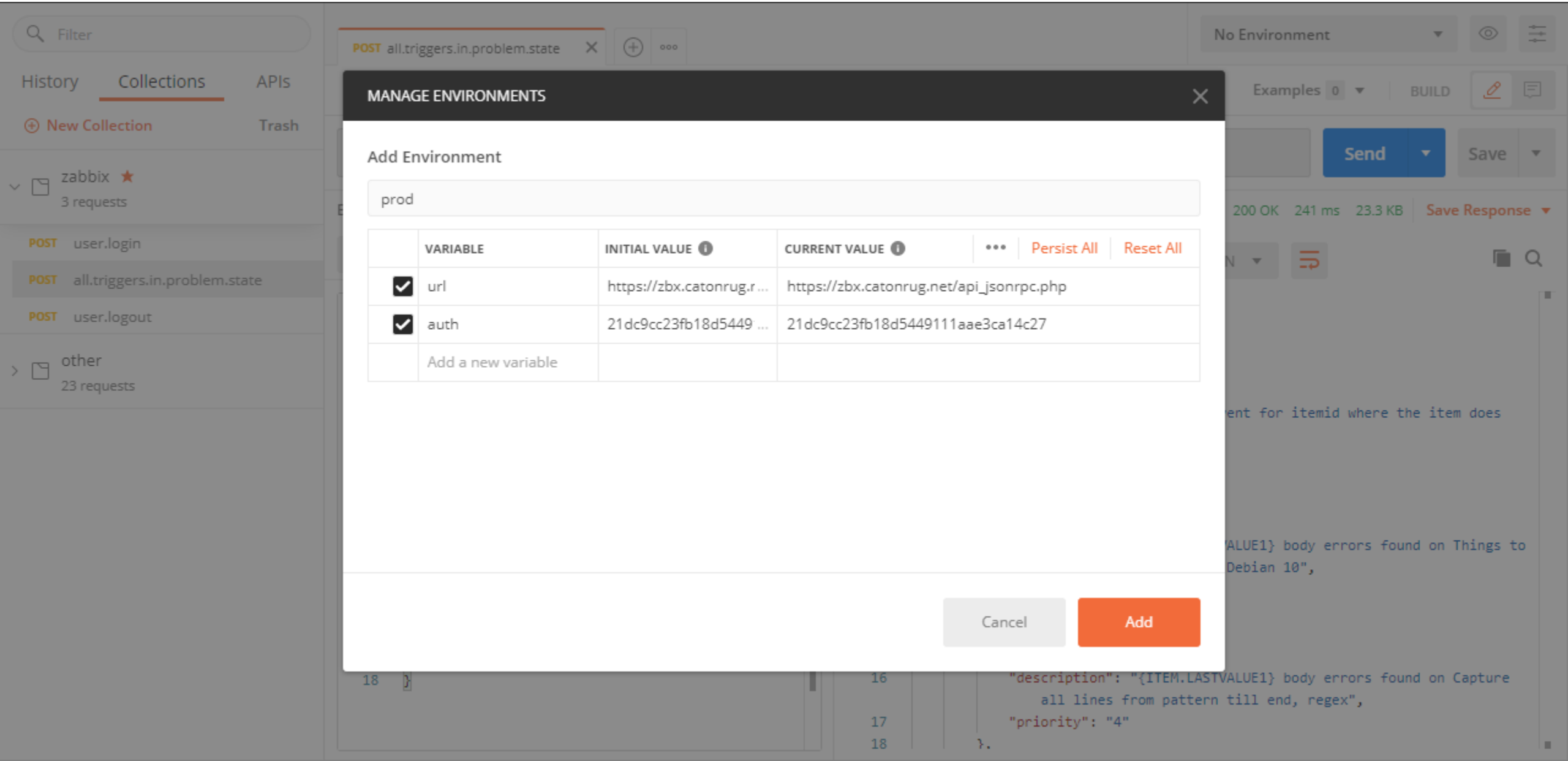
MANAGE ENVIRONMENTS



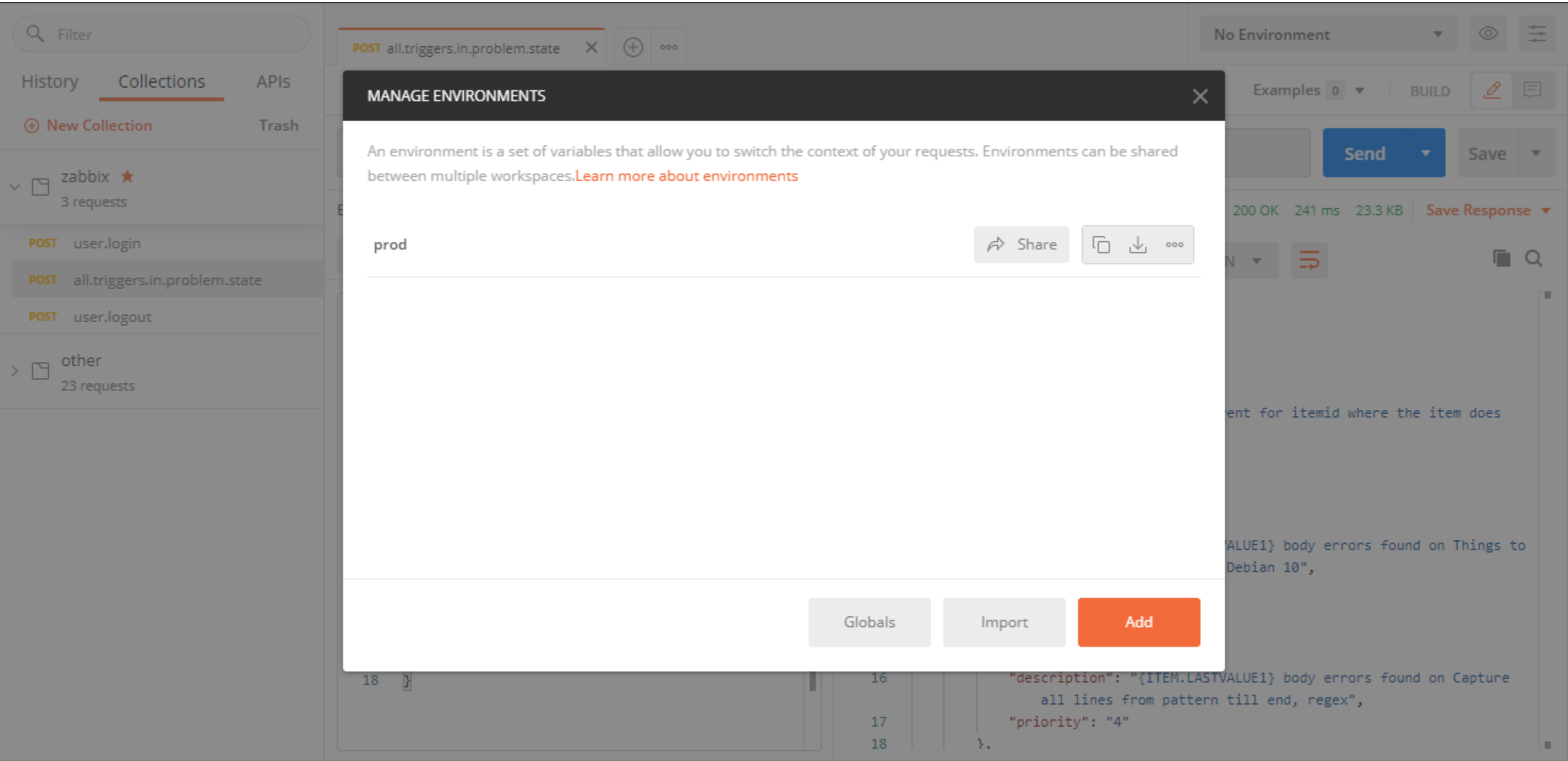
CREATE AN ENVIRONMENT



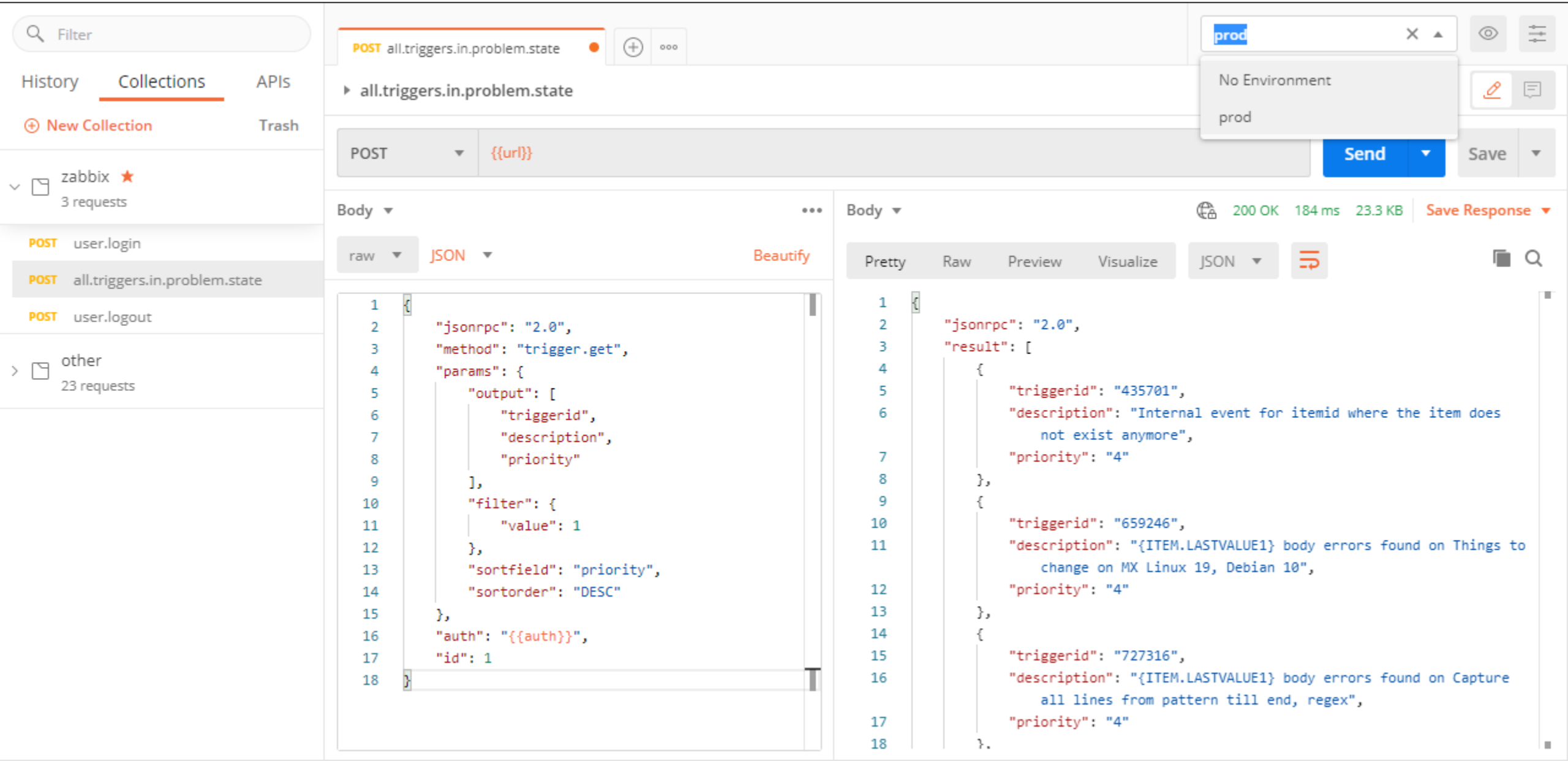
CREATE ENVIRONMENT CHARACTERISTICS



PROD, TEST, 4.0, 5.0



SET ENVIRONMENT, CONFIGURE DYNAMICS



API scripting

via «bash»

Samples = cheat sheet

PREREQUSITES

«curl» available out from box for most of platforms

«jq» provides conveniences to locate element or just beautifuly output. Install on:

CentOS7/RHEL7

```
yum install epel-release && yum install jq
```

CentOS8/RHEL8

```
dnf install jq
```

Ubuntu/Debian

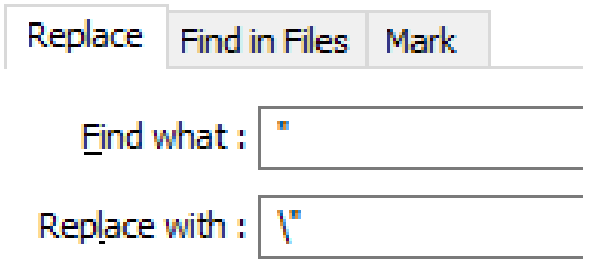
```
apt install jq
```

Standalone 64-bit binary on any Linux:

```
curl -skL "https://github.com/stedolan/jq/releases/download/jq-
1.5/jq-linux64" -o /usr/bin/jq && chmod +x /usr/bin/jq
```

CREATE A QUERY

Visit [documentation page](). Select API flavor.	Escape double quotes. Do even think to do it manually!	Replace: 038e1d7b1735c6a5436ee9eae095879e with: $auth

<pre>{
"jsonrpc": "2.0",
"method": "alert.get",
"params": {
"output": "extend",
"actionids": "3"
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}</pre>
		<pre>curl -s -X POST \
		-H 'Content-Type: application/json-rpc' \
		-d " \
		{
		\"jsonrpc\": \"2.0\",
		\"method\": \"alert.get\",
		\"params\": {
		\"output\": \"extend\",
		\"actionids\": \"3\"
		},
		\"auth\": \"$auth\",
		\"id\": 1
		}
		" $url</pre>

Always execute step 2 at first and only then: step 3

Notice 2 double quotes are not escaped

BASH EXAMPLE

```
#!/bin/bash

# 1. set connection details
url=http://127.0.0.1/api_jsonrpc.php
user=api
password=zabbix

# 2. get authorization tokken
auth=$(curl -s -X POST \
-H 'Content-Type: application/json-rpc' \
-d " \
{
 \"jsonrpc\": \"2.0\",
 \"method\": \"user.login\",
 \"params\": {
  \"user\": \"$user\",
  \"password\": \"$password\"
 },
 \"id\": 1,
 \"auth\": null
}
" $url | \
jq -r '.result'
)
```

```
# 3. show triggers in problem state

curl -s -X POST \
-H 'Content-Type: application/json-rpc' \
-d " \
{
 \"jsonrpc\": \"2.0\",
    \"method\": \"trigger.get\",
    \"params\": {
        \"output\": \"extend\",
        \"selectHosts\": \"extend\",
        \"filter\": {
            \"value\": 1
        },
        \"sortfield\": \"priority\",
        \"sortorder\": \"DESC\"
    },
    \"auth\": \"$auth\",
    \"id\": 1
}
" $url | \
jq -r '.result'
```

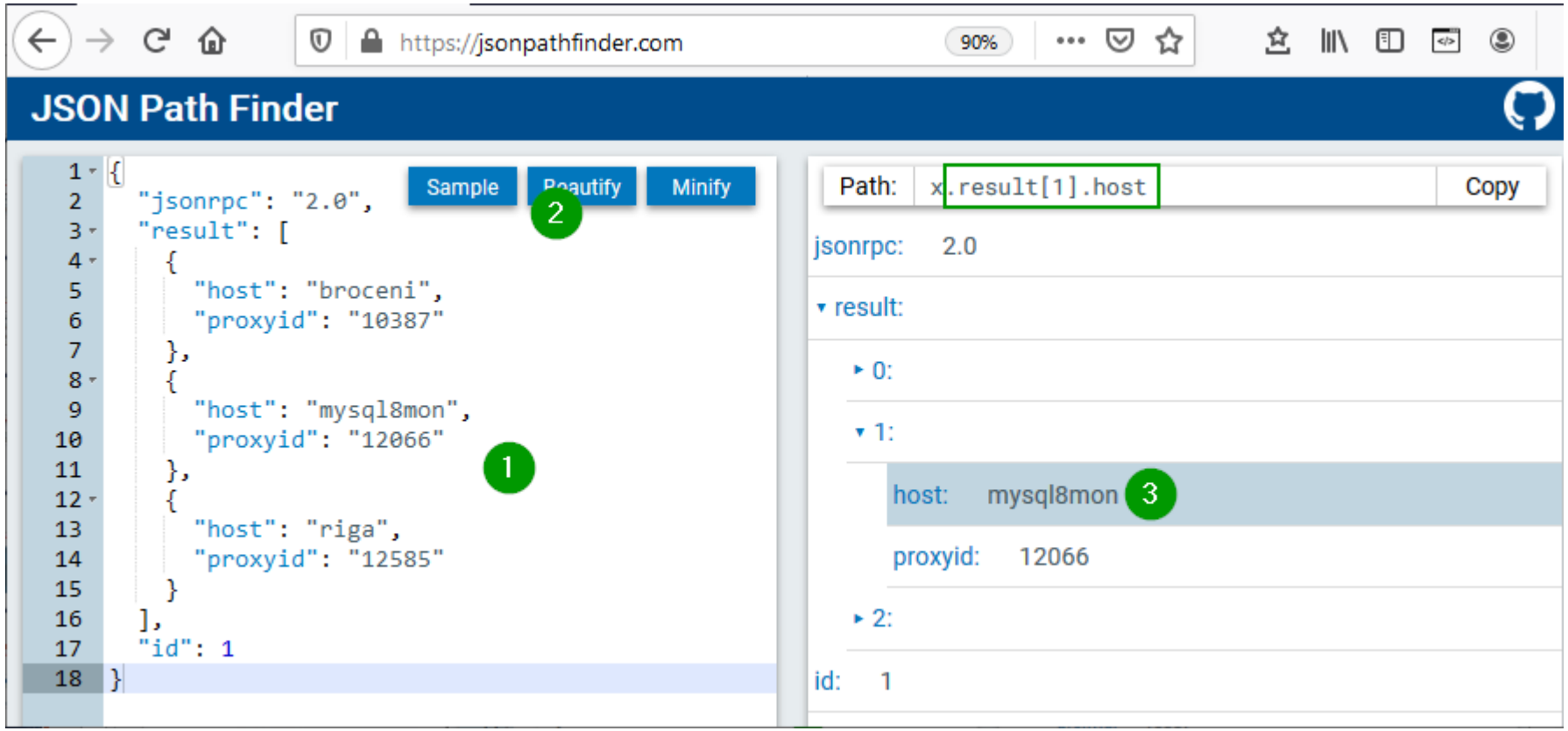
```
# 4. logout user

curl -s -X POST \
-H 'Content-Type: application/json-rpc' \
-d " \
{
    \"jsonrpc\": \"2.0\",
    \"method\": \"user.logout\",
    \"params\": [],
    \"id\": 1,
    \"auth\": \"$auth\"
}
" $url
```

API task starts with ‘user.login’ procedure and ends with ‘user.logout’ procedure

LOCATE ELEMENTS - JSONPATHFINDER.COM

```
# input
curl -s -X POST \
-H 'Content-Type: application/json-rpc' \
-d " \
{
    \"jsonrpc\": \"2.0\",
    \"method\": \"proxy.get\",
    \"params\": {
        \"output\": [\"host\"]
    },
    \"auth\": \"$auth\",
    \"id\": 1
}
" $url
```

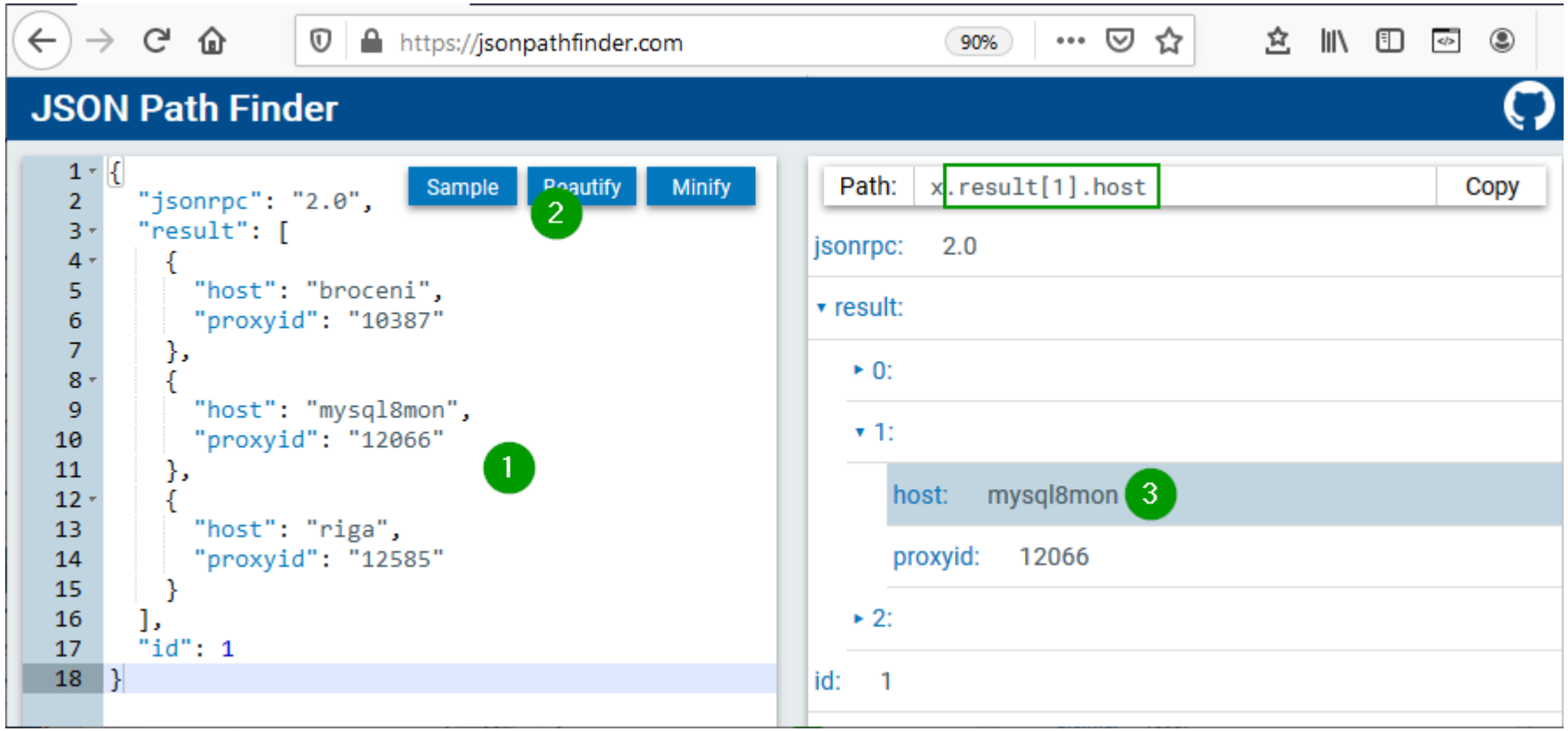
```
# output
{"jsonrpc":"2.0","result":[{"host":"broceni","proxyid":"10387
"},{"host":"mysql8mon","proxyid":"12066"},{"host":"riga","pro
xyid":"12585"}],"id":1}
```



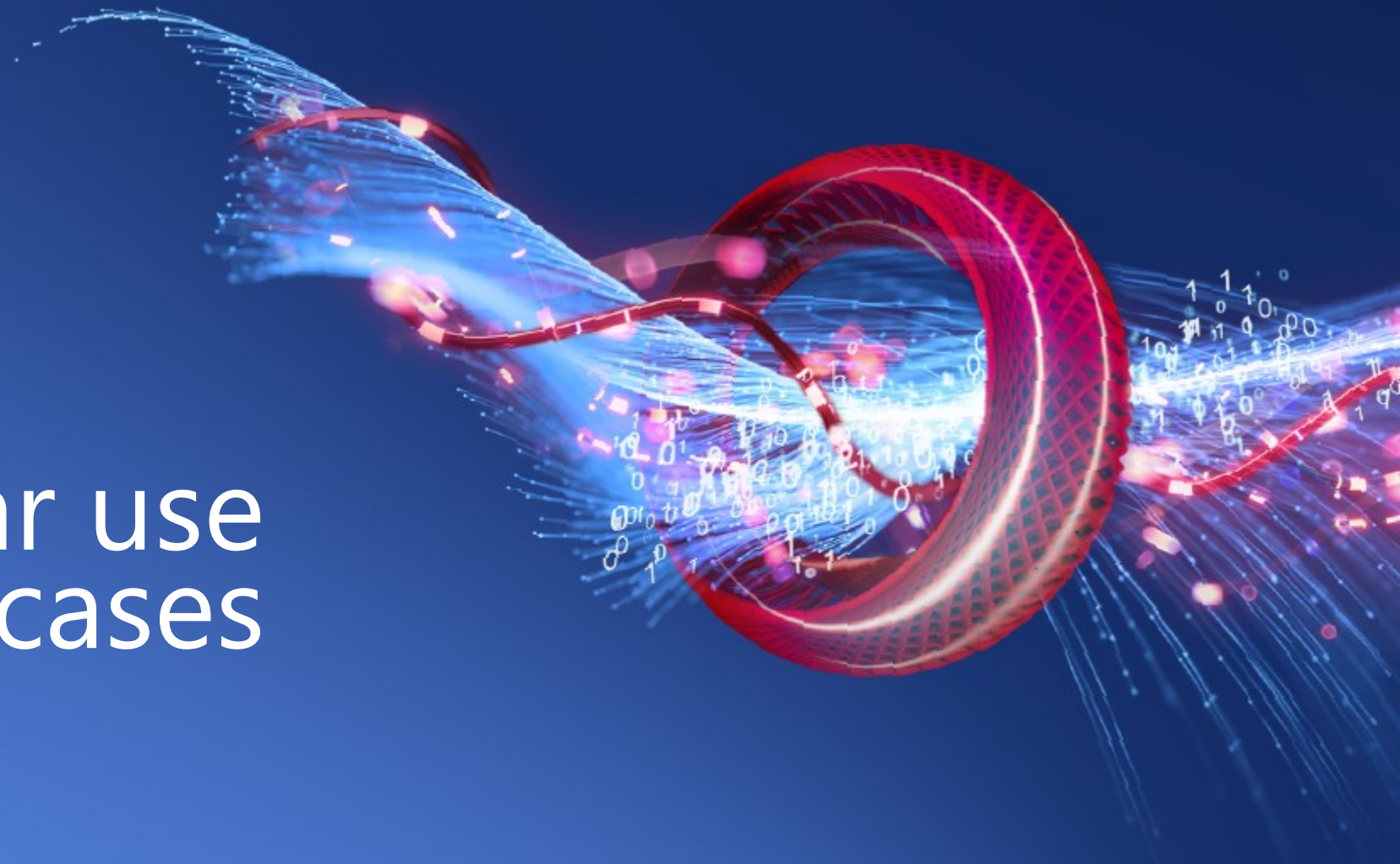
LOCATE ELEMENTS - JSONPATHFINDER.COM

```
# input
curl -s -X POST \
-H 'Content-Type: application/json-rpc' \
-d " \
{
    \"jsonrpc\": \"2.0\",
    \"method\": \"proxy.get\",
    \"params\": {
        \"output\": [\"host\"]
    },
    \"auth\": \"$auth\",
    \"id\": 1
}
" $url | jq -r '.result[].host'
```

```
# output
broceni
mysql8mon
riga
```



Popular use cases



Use case 1: Log file monitoring SNMP traps

CHARACTERISTICS

Over 600 patterns to detect

Everything is stored in one item

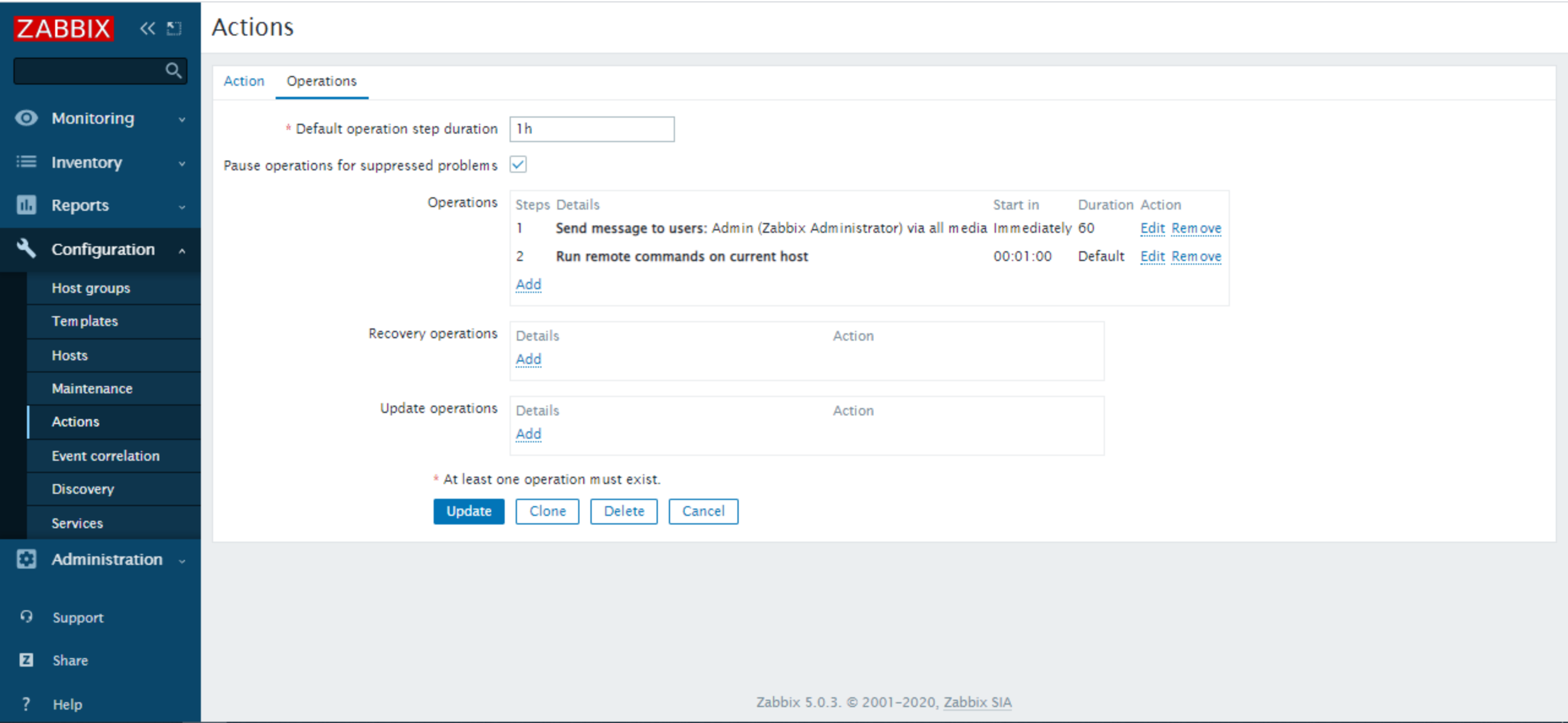
Event generation mode: Multiple

No recovery logic

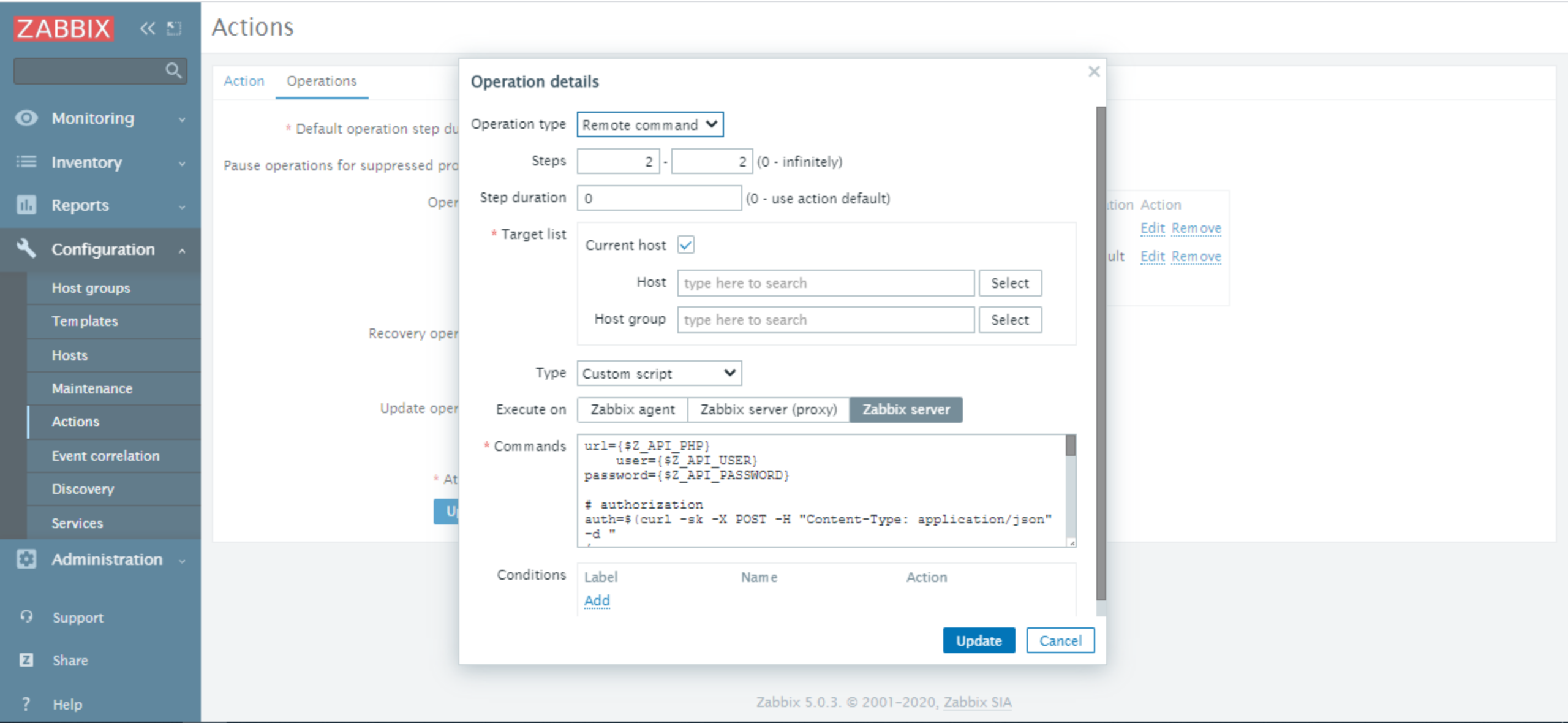
Ambition:

Notify person and close event

OPERATIONS


ZABBIX
Monitoring
Inventory
Reports
Configuration
Host groups
Templates
Hosts
Maintenance
Actions
Event correlation
Discovery
Services
Administration
Support
Share
Help
Actions
Action
Operations
* Default operation step duration
1h
Pause operations for suppressed problems
Operations
Steps Details
Start in
Duration
Action
1 Send message to users: Admin (Zabbix Administrator) via all media
Immediately
60
Edit Remove
2 Run remote commands on current host
00:01:00
Default
Edit Remove
Add
Recovery operations
Details
Action
Add
Update operations
Details
Action
Add
* At least one operation must exist.
Update
Clone
Delete
Cancel
Zabbix 5.0.3. © 2001–2020, Zabbix SIA

REMOTE COMMAND



SOLUTION IN DETAIL

Macro	Value
{$Z_API_PHP}	http://127.0.0.1/api_jsonrpc.php
{$Z_API_PASSWORD}	zabbix
{$Z_API_USER}	api

```
# characteristics to access API
     url={$Z_API_PHP}
    user={$Z_API_USER}
password={$Z_API_PASSWORD}

# authorization
auth=$(curl -sk -X POST -H "Content-Type:
application/json" -d "
{
       \"jsonrpc\": \"2.0\",
       \"method\": \"user.login\",
       \"params\": {
              \"user\": \"$user\",
              \"password\": \"$password\"
       },
       \"id\": 1,
       \"auth\": null
}
" $url | \
grep -E -o "([0-9a-f]{32,32})")
```

```
# achknowledge and close event
curl -sk -X POST -H "Content-Type: application/json" -d "
{
       \"jsonrpc\": \"2.0\",
       \"method\": \"event.acknowledge\",
       \"params\": {
              \"eventids\": \"{EVENT.ID}\",
              \"action\": 1,
              \"message\": \"Problem resolved.\"
       },
       \"auth\": \"$auth\",
       \"id\": 1
}
" $url

# close api key
curl -sk -X POST -H "Content-Type: application/json" -d "
{
    \"jsonrpc\": \"2.0\",
    \"method\": \"user.logout\",
    \"params\": [],
    \"id\": 1,
    \"auth\": \"$auth\"
}
" $url
```

Use case 2: Maintain global macro via API

CHARACTERISTICS

Have managed a way to obtain session token

Will use it in “HTTP agent” item

Ambition:

Maintain session token in global macro

SOLUTION: ZABBIX_GLOBAL_MACRO_UPDATE.SH

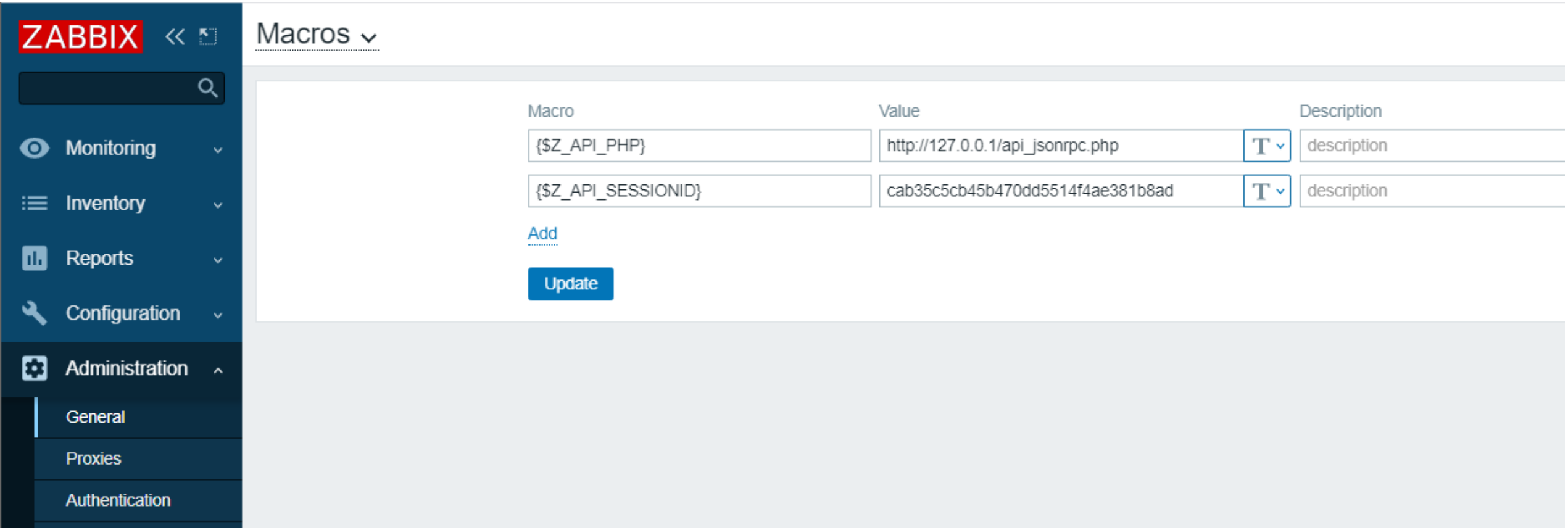
```
#!/bin/bash

# characteristics to access API
url=http://127.0.0.1/api_jsonrpc.php
user=Admin
password=zabbix

# argument 1
macro=$1

# argument 2
value=$2

# get authorization token
auth=$(curl -s -X POST \
-H 'Content-Type: application/json-rpc' \
-d " \
{
 \"jsonrpc\": \"2.0\",
 \"method\": \"user.login\",
 \"params\": {
  \"user\": \"$user\",
  \"password\": \"$password\"
 },
 \"id\": 1,
 \"auth\": null
}
" $url | grep -E -o "([0-9a-f]{32,32})")

# get global user macro id
id=$(curl -s -X POST \
-H 'Content-Type: application/json-rpc' \
-d " \
{
    \"jsonrpc\": \"2.0\",
    \"method\": \"usermacro.get\",
    \"params\": {
        \"output\": [\"globalmacroid\"],
        \"globalmacro\": true,
        \"filter\":{\"macro\": \"$macro\"}
    },
    \"auth\": \"$auth\",
    \"id\": 1
}
" $url | jq -r ".result[].globalmacroid")

# update
curl -s -X POST \
-H 'Content-Type: application/json-rpc' \
-d " \
{
    \"jsonrpc\": \"2.0\",
    \"method\": \"usermacro.updateglobal\",
    \"params\": {
        \"globalmacroid\": \"$id\",
        \"value\": \"$value\"
    },
    \"auth\": \"$auth\",
    \"id\": 1
}
" $url

# logout user
curl -s -X POST \
-H 'Content-Type: application/json-rpc' \
-d " \
{
    \"jsonrpc\": \"2.0\",
    \"method\": \"user.logout\",
    \"params\": [],
    \"id\": 1,
    \"auth\": \"$auth\"
}
" $url
```

ZABBIX_GLOBAL_MACRO_UPDATE.SH: SAMPLE

```
zabbix_global_macro_update.sh '{$TIME.ZONE.VAR}' "Europe/Riga (EEST, +0300)"

zabbix_global_macro_update.sh '{$TIME.ZONE.VAR}' "$(timedatectl | grep -oP "(Time zone: \K.*$)")"

zabbix_global_macro_update.sh '{$TIME.ZONE.VAR}' "$(zabbix_show_local_timezone.sh)"
```

Use case 3: Zabbix API key in a global level

DEFINE MACRO AT GLOBAL LEVEL



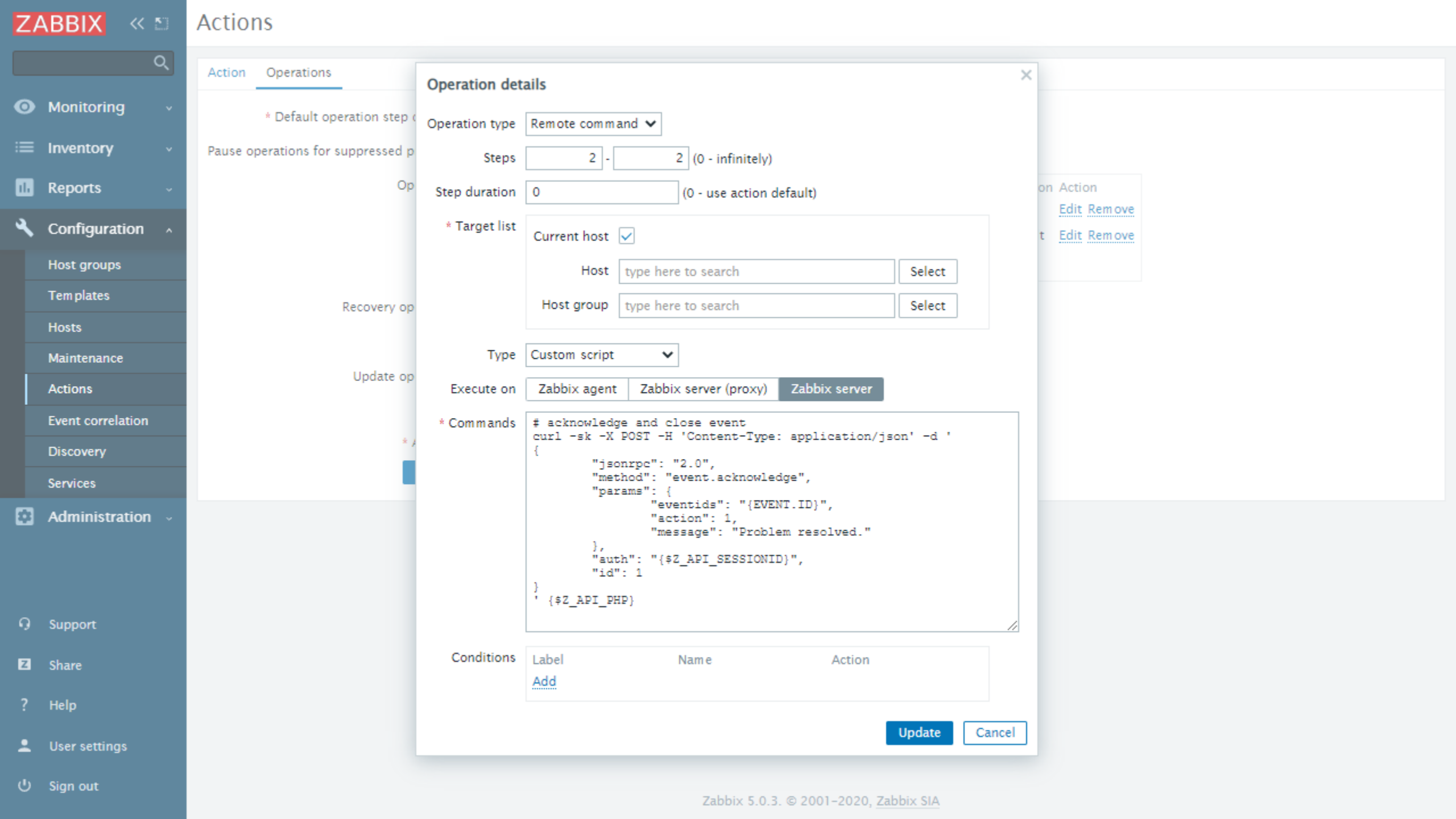

```
curl -s -X POST -H 'Content-Type: application/json' http://127.0.0.1/api_jsonrpc.php -d \
'{"jsonrpc":"2.0","method":"user.login","params":{"user":"api","password":"zabbix"},"id":1,"auth":null}' | \
grep -E -o "([0-9a-f]{32,32})"
```

CASE 1: IMPROVED

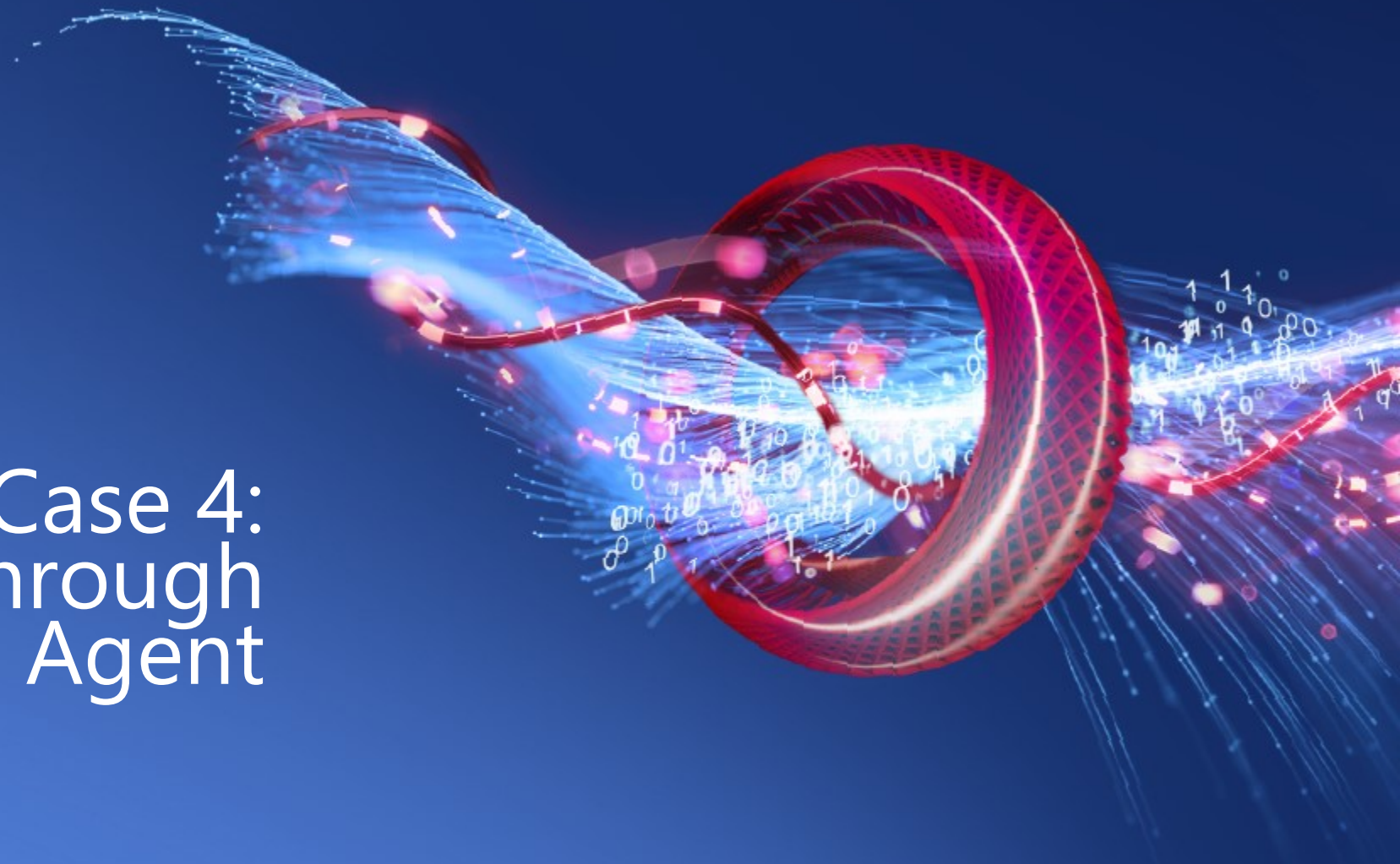
```
# characteristics to access API
     url={$Z_API_PHP}
    user={$Z_API_USER}
password={$Z_API_PASSWORD}

# authorization
auth=$(curl -sk -X POST -H "Content-Type:
application/json" -d "
{
       \"jsonrpc\": \"2.0\",
       \"method\": \"user.login\",
       \"params\": {
              \"user\": \"$user\",
              \"password\": \"$password\"
       },
       \"id\": 1,
       \"auth\": null
}
" $url | \
grep -E -o "([0-9a-f]{32,32})")
```

```
# acknowledge and close event
curl -sk -X POST -H 'Content-Type: application/json' -d '
{
       "jsonrpc": "2.0",
       "method": "event.acknowledge",
       "params": {
              "eventids": "{EVENT.ID}",
              "action": 1,
              "message": "Problem resolved."
       },
       "auth": "{$Z_API_SESSIONID}",
       "id": 1
}
' {$Z_API_PHP}

# close api key
curl -sk -X POST -H "Content-Type: application/json" -d "
{
    \"jsonrpc\": \"2.0\",
    \"method\": \"user.logout\",
    \"params\": [],
    \"id\": 1,
    \"auth\": \"$auth\"
}
" $url
```

No need to escape double quotes anymore
Original sample still is more bulletproof


ZABBIX
Actions
Monitoring
Inventory
Reports
Configuration
Host groups
Templates
Hosts
Maintenance
Actions
Event correlation
Discovery
Services
Administration
Support
Share
Help
User settings
Sign out
Action
Operations
Operation details
Operation type
Remote command
Steps
2
2
(0 - infinitely)
Step duration
0
(0 - use action default)
Target list
Current host
Host
type here to search
Select
Host group
type here to search
Select
Type
Custom script
Execute on
Zabbix agent
Zabbix server (proxy)
Zabbix server
Commands
acknowledge and close event
curl -sk -X POST -H 'Content-Type: application/json' -d '
{
"jsonrpc": "2.0",
"method": "event.acknowledge",
"params": {
"eventids": "{EVENT.ID}",
"action": 1,
"message": "Problem resolved."
},
"auth": "{$Z_API_SESSIONID}",
"id": 1
}
' {$Z_API_PHP}
Conditions
Label
Name
Action
Add
Update
Cancel
Edit
Remove
Edit
Remove
Zabbix 5.0.3. © 2001–2020, Zabbix SIA

USE Case 4: API through HTTP Agent



DISCOVERY RULE

Discovery rule | Preprocessing | LLD macros | Filters | Overrides

* Name: Discover all proxies

Type: HTTP agent

* Key: get.proxy.list

* URL: {$Z_API_PHP} Parse

Request type: POST

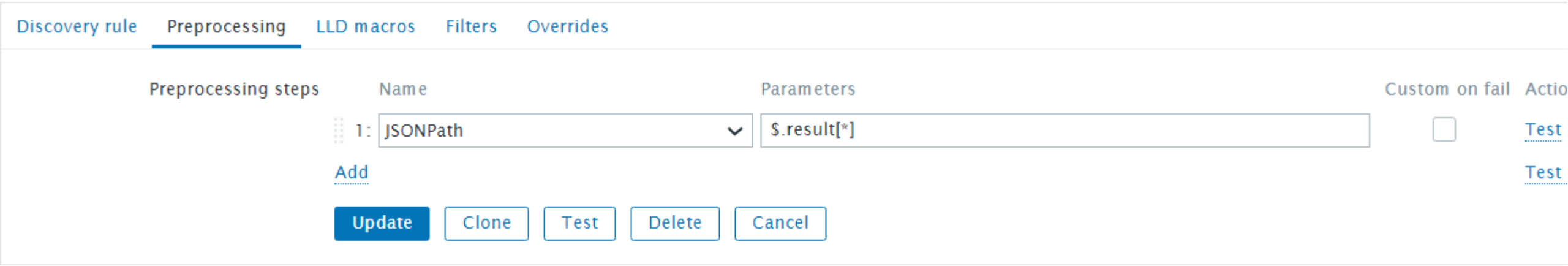
Timeout: 8s

Request body type: Raw data | JSON data | XML data

Request body:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.get",
    "params": {
        "output": "extend",
        "selectInterface": "extend"
    },
    "auth": "{$Z_API_SESSIONID}",
    "id": 1
}
```

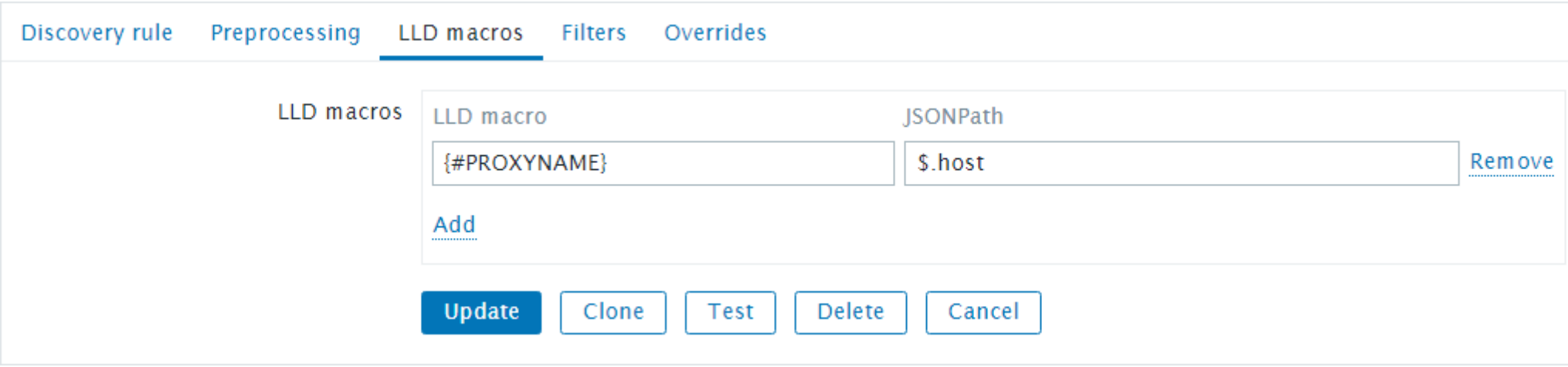
PREPROCESSING




```
{
        "jsonrpc": "2.0",
        "result": [{
                "host": "broceni",
                "proxyid": "10387"
        }, {
                "host": "mysql8mon",
                "proxyid": "12066"
        }, {
                "host": "riga",
                "proxyid": "12585"
        }],
        "id": 1
}
```

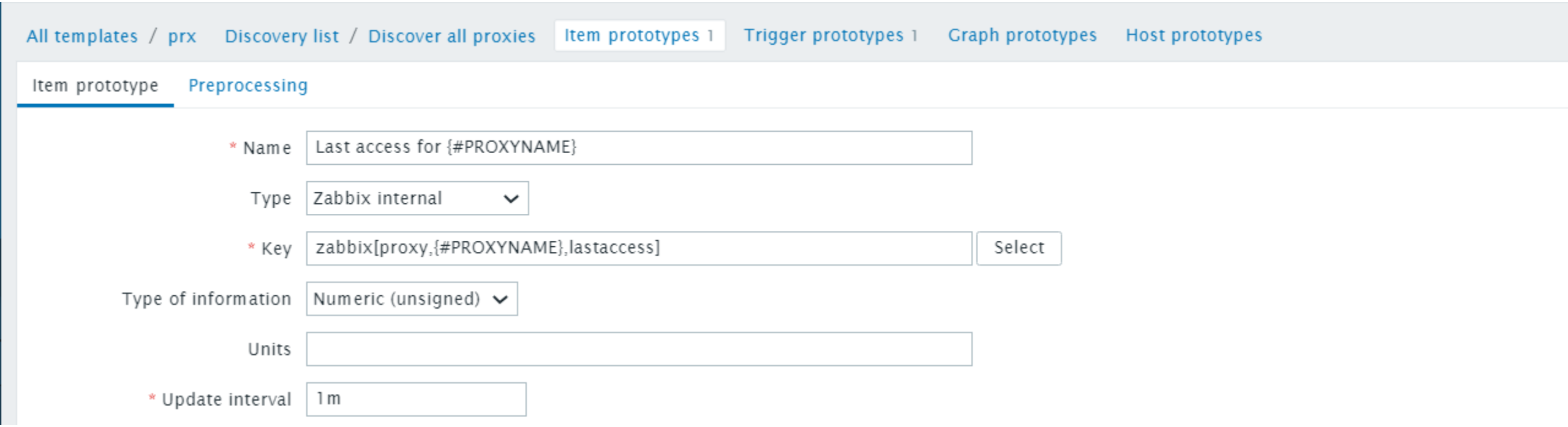
```
[{
        "host": "broceni",
        "proxyid": "10387"
}, {
        "host": "mysql8mon",
        "proxyid": "12066"
}, {
        "host": "riga",
        "proxyid": "12585"
}]
```

MAPPING

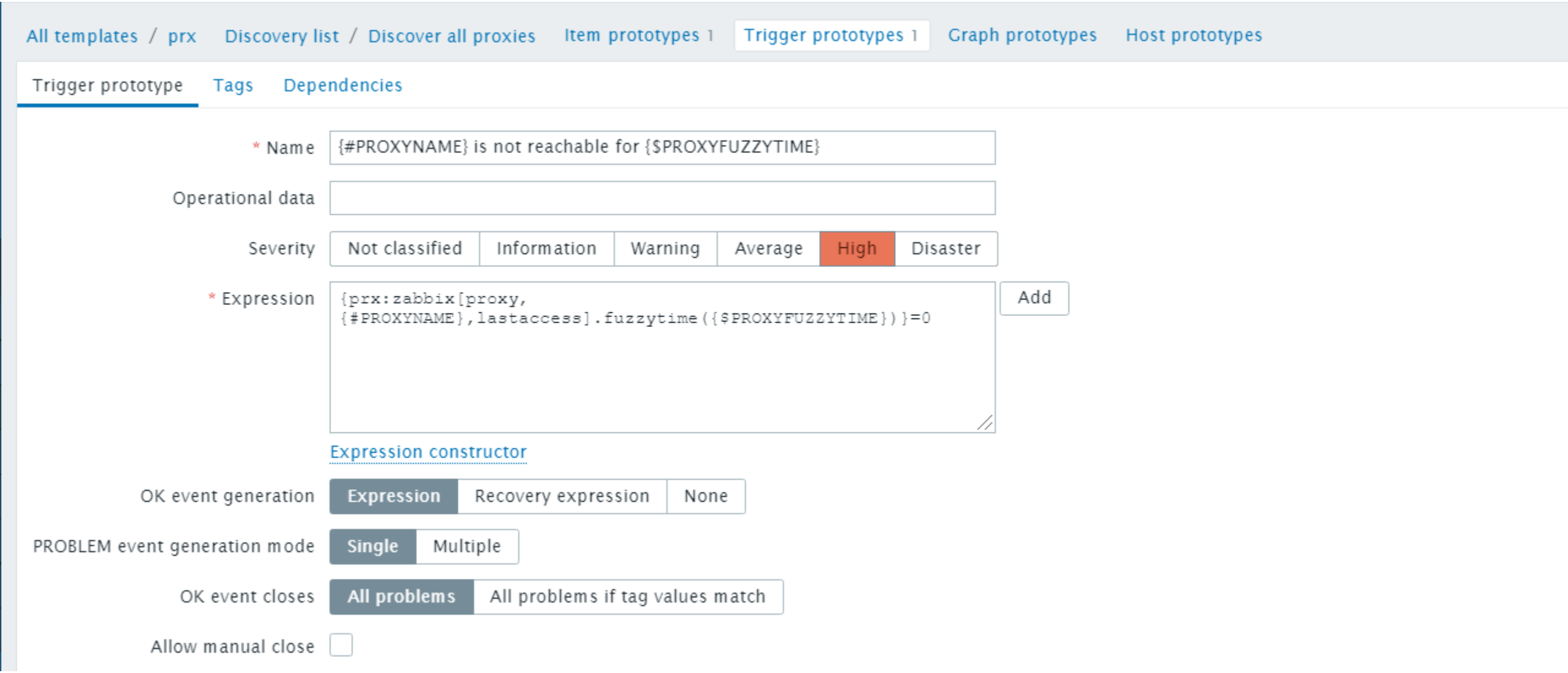



```
[{
        "host": "broceni",
        "proxyid": "10387"
}, {
        "host": "mysql8mon",
        "proxyid": "12066"
}, {
        "host": "riga",
        "proxyid": "12585"
}]
```

ITEM PROTOTYPE

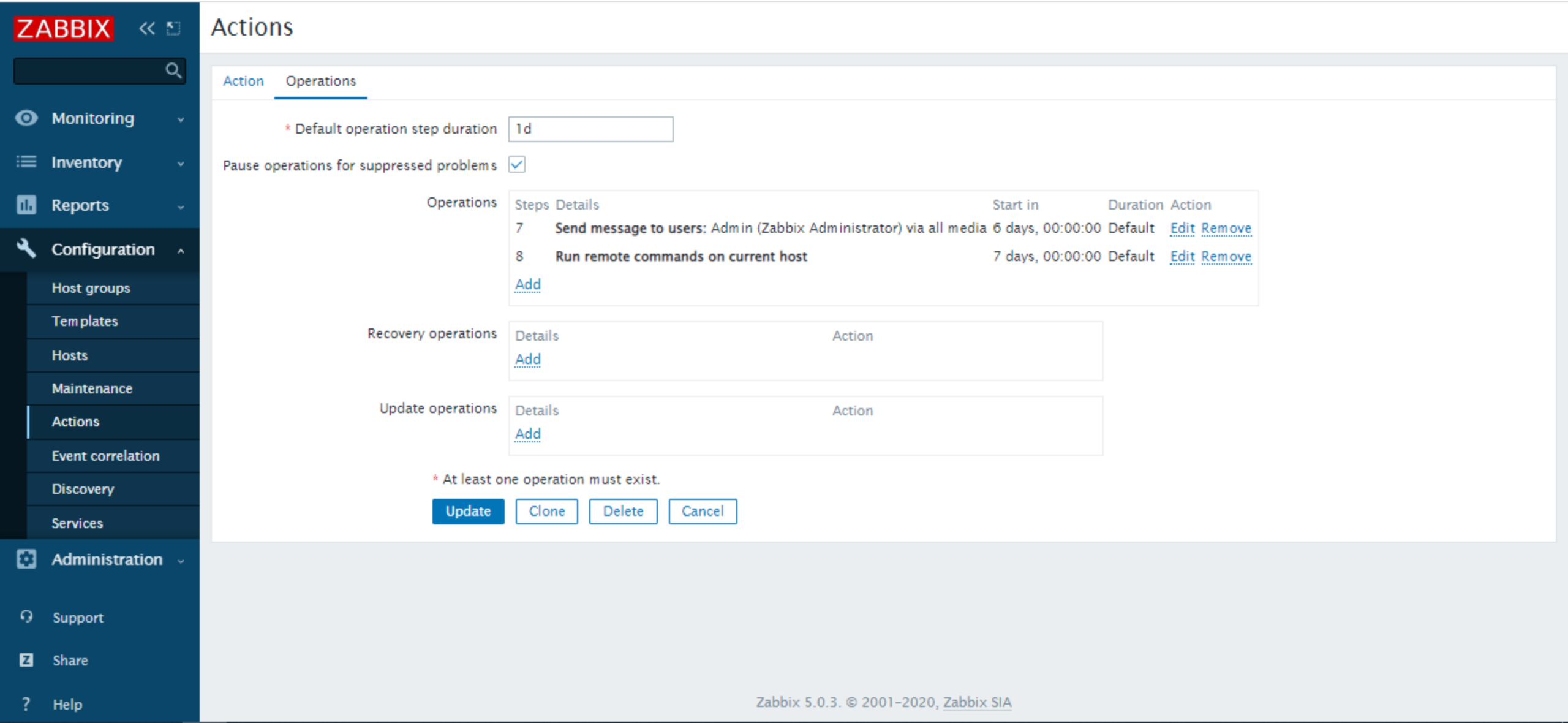


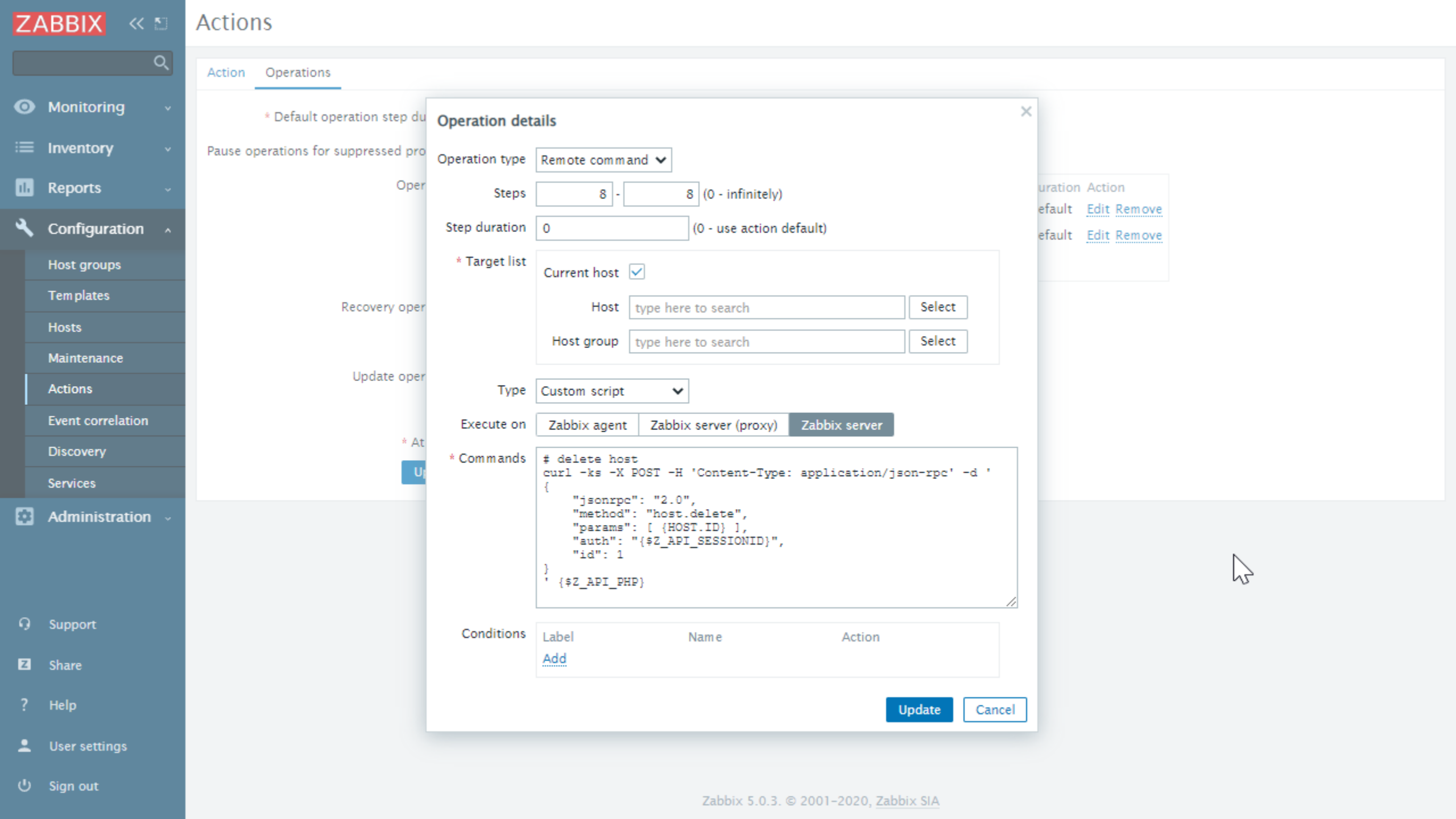
TRIGGER PROTOTYPE



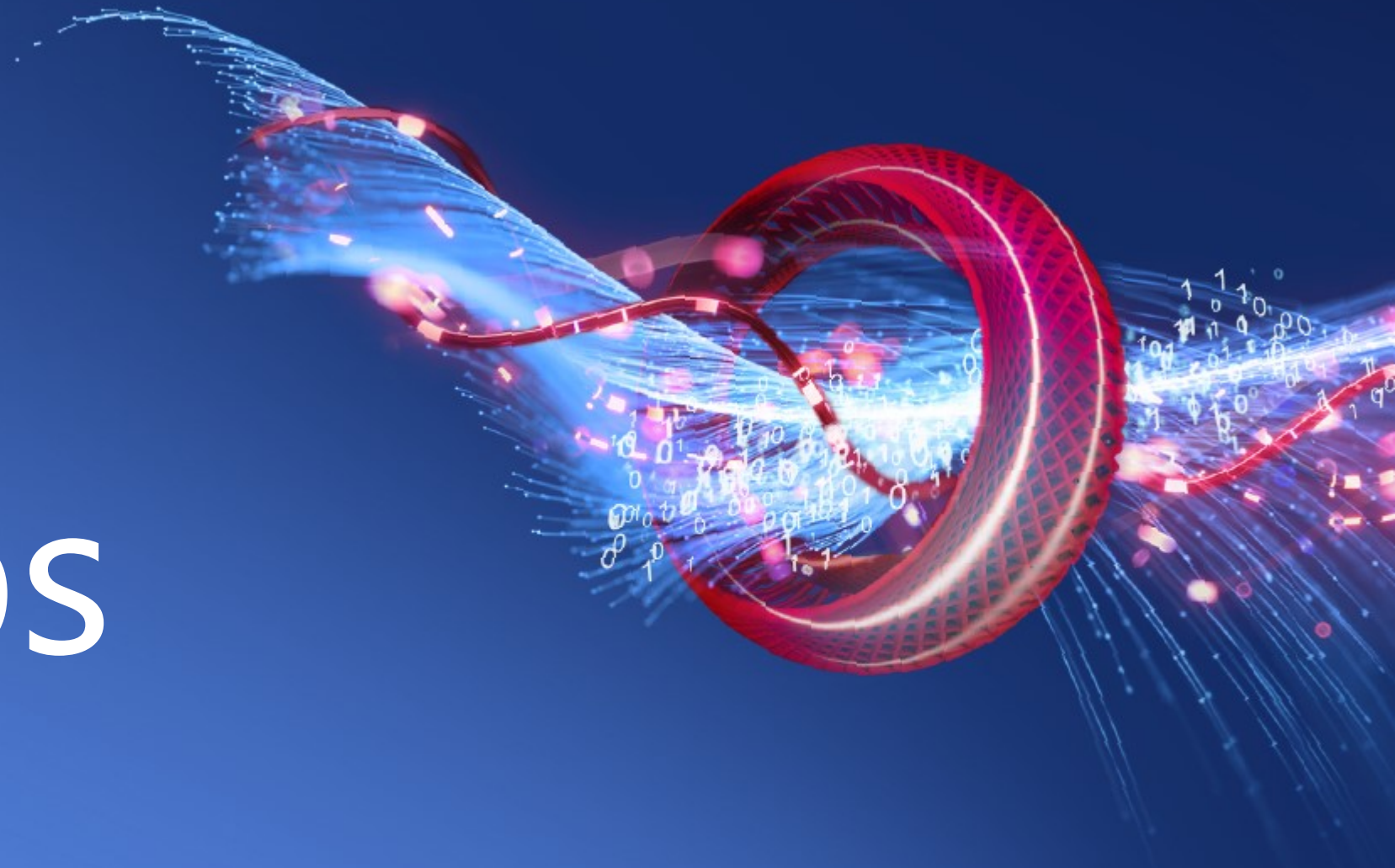
USE Case 5: Delete inactive hosts

DELAYED ESCALATION




ZABBIX
Actions
Monitoring
Inventory
Reports
Configuration
Host groups
Templates
Hosts
Maintenance
Actions
Event correlation
Discovery
Services
Administration
Support
Share
Help
User settings
Sign out
Action
Operations
Operation details
Operation type
Remote command
Steps
8
8
(0 - infinitely)
Step duration
0
(0 - use action default)
Target list
Current host
Host
type here to search
Select
Host group
type here to search
Select
Type
Custom script
Execute on
Zabbix agent
Zabbix server (proxy)
Zabbix server
Commands
delete host
curl -ks -X POST -H 'Content-Type: application/json-rpc' -d '
{
"jsonrpc": "2.0",
"method": "host.delete",
"params": [{HOST.ID}],
"auth": "{$Z_API_SESSIONID}",
"id": 1
}
' {$Z_API_PHP}
Conditions
Label
Name
Action
Add
Update
Cancel
Edit
Remove
Zabbix 5.0.3. © 2001–2020, Zabbix SIA

Tips



TIPS

- Use dedicated API user for API calls. For example, username «api»
- «curl» does not work via docker «alpine» container. Workaround is to use a «centos» container
- When «jq» utility is not available. Can try a standart grep:

```
grep -oP 'result\":\"\K\w+'
grep -oP 'itemid\":\K\d+’
grep -oP 'host\":\K\w+'
```

THANK YOU!