

The Zabbix API

and

Zabcon

“The Zabbix Console”

Andrew Nelson

- RHCE, Zabbix Certified Specialist
 - Active in the Zabbix community for approximately 7 years
 - Nelsonab in the Zabbix forums.
 - Creator of Zabcon, the Zabbix console.
- <http://trac.red-tux.net>
- Red Hat Consultant



- Introduced in Zabbix 1.8.0
- Based on JSON-RPC v 2.0 over HTTP
- Not RESTful
- Provides a somewhat raw interface to the internal data structures
- API calls are made to the Frontend, not the Zabbix server process
- API calls mimic Frontend activities

Sounds interesting what do I need to know?

- URL used to access the api:
<Server URL>/api_jsonrpc.php
Example:
http://myserver.example.com/zabbix/api_jsonrpc.php
- A valid Zabbix user is required for all API calls
 - Often referred to as the “API User”
 - Must be a member of a User group which has the API Access permission enabled
 - API Users are Zabbix users with the ability to perform API calls
- The latest documentation is online:
<http://www.zabbix.com/documentation/1.8/api>
- The forum is a great resource.

My API User is configured, now what?

- Use of a Zabbix API library can make things easier
 - Not required, the brave use Curl and Wget.
- API calls are divided into namespaces
 - Namespace.call
 - user.get, host.get etc
- Every API function requires a valid session ID.
 - Except user.login, which is used to generate the session ID.
- Basic API Call/Layout:

```
{  
  "jsonrpc": "2.0",  
  "method": "Function",  
  "params": { "Name": "Value" },  
  "auth": "SessionID",  
  "id": <Incrementing Number>  
}
```

- Order is not important, content is

Show me the data! (Login)

The next few slides will show the steps needed to show all of the hosts on a system.

The login:

```
{
  "auth":null,"method":"user.login",
  "id":0,"jsonrpc":"2.0",
  "params":{"password":"apitest","user":"apitest"}
}
```

Result:

```
{
  "jsonrpc":"2.0",
  "result":"ef1118f0c916f8c49d10913c31ba804c",
  "id":0
}
```

Show me the data! (host.get)

The call:

```
{
  "auth": "ef1118f0c916f8c49d10913c31ba804c",
  "method": "host.get", "id": 3,
  "params": {}, "jsonrpc": "2.0"
}
```

Result:

```
{
  "jsonrpc": "2.0",
  "result":
    [{"hostid": "10047"}, {"hostid": "10064"},
     {"hostid": "10067"}],
  "id": 3
}
```

Wait, is that all I get from host.get?

Let's try again with the parameter "extend output"

```
{  
  "auth": "ef1118f0c916f8c49d10913c31ba804c", "id": 4,  
  "method": "host.get", "params": {"output": "extend"},  
  "jsonrpc": "2.0"  
}
```


Holy data Batman, that's a lot!

Result:

```
{ "jsonrpc": "2.0",  
  "result":  
  [ { "maintenances": [ { "maintenanceid": "0" } ],  
    "hostid": "10047", "proxy_hostid": "0", "host": "test", "dns": "",  
    "useip": "1", "ip": "127.0.0.1", "port": "10050", "status": "0",  
    "disable_until": "0", "error": "", "available": "0",  
    "errors_from": "0", "lastaccess": "0", "inbytes": "0", "outbytes": "0",  
    "useipmi": "0", "ipmi_port": "623", "ipmi_authtype": "-1",  
    "ipmi_privilege": "2", "ipmi_username": "", "ipmi_password": "",  
    "ipmi_disable_until": "0", "ipmi_available": "0",  
    "snmp_disable_until": "0", "snmp_available": "0",  
    "maintenanceid": "0", "maintenance_status": "0",  
    "maintenance_type": "0", "maintenance_from": "0", "ipmi_ip": "",  
    "ipmi_errors_from": "0", "snmp_errors_from": "0", "ipmi_error": "",  
    "snmp_error": "" },  
    { Truncated Data for Host 2 }, { Truncated Data for Host3 } ],  
  "id": 4 }
```

Is there an easier way?

- Zabcon is intended to be the “Zabbix Console”
 - Started in late 2009
 - Latest version 0.0.332 (released today!)
- Website: <http://trac.red-tux.net>
- Written in Ruby
 - Requires at least 1.8.6
 - No major issues with 1.9.x, please report them when you find them
- Can accept commands from standard input
 - One person used Zabcon inside a script to import data into Zabbix, using this approach they were able to import about 1,000 hosts in a matter of minutes.
- Users can create their own custom Zabcon commands
- Easiest way to install is from Rubygems

```
# gem install zabcon
```

 - Installation and update of dependencies is managed by Rubygems.
 - Some dependencies may require a compiler and the Ruby development libraries

Let's see that get host example with Zabcon

```
$ zabcon.rb
Unable to find a default configuration file
i386-linux
Welcome to Zabcon. Build Number: 326
Use the command 'help' to get help on commands
-> login http://192.168.166.10 apitest apitest
http://192.168.166.10 connected
API Version: 1.3
+> get host
Host result set
+-----+-----+-----+-----+
| hostid | host      | dns  | ip      |
+-----+-----+-----+-----+
| 10047  | test     |      | 127.0.0.1 |
| 10064  | Lua 2    |      | 0.0.0.0   |
| 10067  | new test |      | 127.0.0.1 |
+-----+-----+-----+-----+
3 rows total
+>
```



That's nice do I really need to log in every time?

- Zabcon can load host and user information from a configuration file
- Default behavior is to load the config file upon startup
 - `./zabcon.conf` and `~/zabcon.conf` are searched in that order
 - Sample config file: `zabcon.conf.default` can be found in your system's Rubygems directory

```
$ gem which zabcon
```

```
/usr/lib/ruby/gems/1.8/gems/zabcon-0.0.332/./zabcon.rb
```

- Using the above information the sample config file would be found at :

```
/usr/lib/ruby/gems/1.8/gems/zabcon-0.0.332/zabcon.conf.default
```
- Example:

```
server=http://192.168.166.10/  
username=apitest  
password=apitest  
#debug=0  
#truncate_length=5000  
#custom_commands=sample_custom_commands
```

Let's try that again, with more information:

```
$ zabcon.rb
i386-linux
Found valid login credentials, attempting login
http://192.168.166.10 connected
API Version: 1.3
Welcome to Zabcon. Build Number: 326
Use the command 'help' to get help on commands
+> get host show=hostid,host,dns,ip,available,status
Host result set
+-----+-----+-----+-----+-----+-----+
| hostid | host      | dns | ip          | available | status |
+-----+-----+-----+-----+-----+-----+
| 10047  | test     |    | 127.0.0.1  | 0         | 0      |
| 10064  | Lua 2    |    | 0.0.0.0    | 0         | 0      |
| 10067  | new test |    | 127.0.0.1  | 0         | 0      |
+-----+-----+-----+-----+-----+-----+
3 rows total
+>
```



Ok that's cool but what if I want to disable a host from a script?

```
$ echo "get host show=hostid,host,dns,ip,status" | zabcon.rb
```

```
hostid,host,dns,ip,status  
10047,test,,127.0.0.1,0  
10064,Lua 2,,0.0.0.0,0  
10067,new test,,127.0.0.1,0
```

```
$ echo "update host hostid=10047 status=1" | zabcon.rb
```

```
$ echo "get host show=hostid,host,dns,ip,status" | zabcon.rb
```

```
hostid,host,dns,ip,status  
10047,test,,127.0.0.1,1  
10064,Lua 2,,0.0.0.0,0  
10067,new test,,127.0.0.1,0
```

Accessing the API directly

- Right now, Zabcon does not have a command for every possible API call.
- “raw api” allows direct access to the api.

```
+> raw api history.get itemids=[22163] time_from=1317367232
time_to=1317367262 output=extend
```

Raw_api result set

clock	itemid	value
1317367283	22163	1317367283
1317367343	22163	1317367343
...		
1317367943	22163	1317367943
1317368003	22163	1317368003

13 rows total



Custom commands

- End users can create their own Zabcon commands to suit their own unique needs
- The command “help commands” will be updated with the new commands
- Zabcon.conf file determines the location of the custom commands file
- Custom commands file is only parsed on Zabcon startup.

```
ZabconCommand.add_command "custom" do
  set_method do |params|
    server.connection.raw_api("user.get", {})
  end
  set_help_tag :none
  set_flag :print_output
end
```

```
+> custom
  result set
+-----+
| userid |
+-----+
| 1      |
| 2      |
+-----+
2 rows total
```


Custom commands

- End users can create their own Zabcon commands to suit their own unique needs
- The command “help commands” will be updated with the new commands
- Zabcon.conf file determines the location of the custom commands file
- Custom commands file is only parsed on Zabcon startup.

```
ZabconCommand.add_command "custom" do
  set_method do |params|
    server.connection.raw_api("user.get", {})
  end
  set_help_tag :none
  set_flag :print_output
end
```

```
+> custom
  result set
+-----+
| userid |
+-----+
| 1      |
| 2      |
+-----+
2 rows total
```

Time for me to stop talking

- API is JSON-RPC based
 - Allows for a nearly complete interface for modifying Zabbix
- API calls are made to the Web frontend
- Zabcon allows you to easilly call the Zabbix API form the command line or scripts
 - Help from more developers and testers is always appreciated (hint hint)

Website:

<http://trac.red-tux.net>

Email:

nelsonab@red-tux.net

anelson@redhat.com