

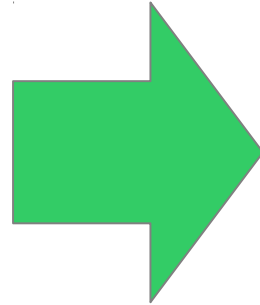


# Next generation API

**Pavels Jelisejevs**  
Web developer at Zabbix

# ZABBIX

Zabbix  
API



Mobile  
applications

Frontend  
extensions

Service  
integration

# ZABBIX

“Amazed at how not logical the #zabbix #API sometimes is.”



“Zabbix API my ass. Querying mysql database directly is way faster.”

# ZABBIX

## Next generation API

- 1. Improvements**

- 2. New features**

- 3. Future concepts**

## Better documentation

- ◆ A task with the highest priority
- ◆ Everything has to be documented
- ◆ Multiple examples will be provided to demonstrate the functionality

## Improved interface

- ◆ Some methods and parameters are confusing and unintuitive

host.massremove

“groupCount” or “only\_true” for trigger.get

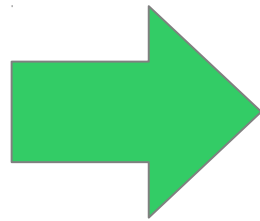
- ◆ They will be refactored so that their purpose would be completely clear

## Improved interface

- ◆ Some parameters will be grouped and restructured for easier use

**Output:** shorten /  
refer / extend /  
array

**countOutput**



**Output:** full / count /  
array

## Strict validation

### ◆ The API will throw errors if:

A mandatory parameter is missing

An unsupported parameter is passed

A parameter has incorrect data type or value

Something else is wrong with the input for example, duplicate values are present

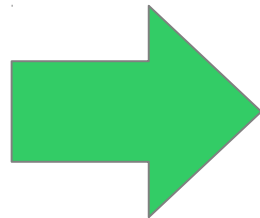


## Reliable error reporting

- ◆ The error messages will provide enough detail to locate the source of the problem

Incorrect arguments passed to function.

No permissions to referred object or it does not exist!



Missing the “key\_” parameter for item “Processor load”

No permissions to host group with ID 439 or it does not exist!

## Reliable error reporting

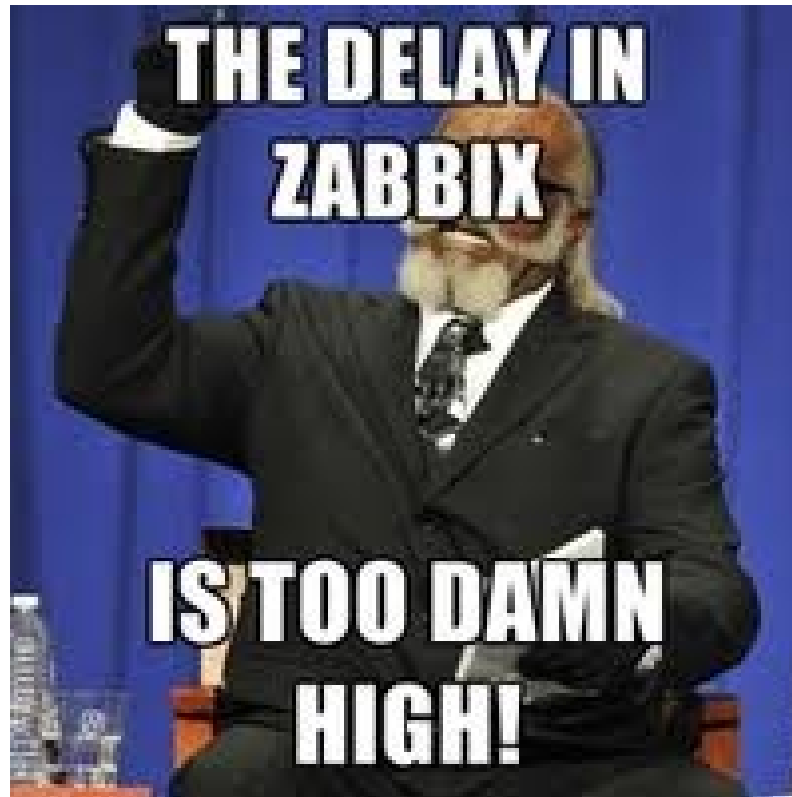
- ◆ Error handling will be easier with a set of error codes for common problems, such as:

Creating nonunique objects

Requesting unexisting objects or lacking permissions

# ZABBIX

## Boosted performance



## Boosted performance

- ◆ API should scale up to a **million hosts and 100 million items and triggers**
- ◆ The response times should not exceed **1 second** for most requests

# ZABBIX

## Next generation API

**1. Improvements**

**2. New features**

**3. Future concepts**

## Versioning

- ◆ A clear versioning policy will be implemented
- ◆ No backward incompatible changes in a major release
- ◆ Developers will be able to adjust their applications to specific API versions

## Maps and graphs API

- ◆ It will be possible to generate maps and graphs using the API
- ◆ They will be easily embedded in user applications to visualize data

## History and trends API

- ◆ All of the gathered data will be retrievable for further processing
- ◆ The client will be able to retrieve both exact and aggregated values



# ZABBIX

## Next generation API

- 1. Improvements**

- 2. New features**

- 3. Future concepts**

## More flexibility

- ◆ The API should remain easy to use, but allow more flexibility for advanced users
- ◆ Additional options for API parameters will allow to extend their behavior

## More flexibility

```
"search": "linux"
```

```
// search only in host technical and visible  
names
```

```
"search": {  
  "value": "linux",  
  "columns": "name,host"  
}
```

```
// perform an exact search
```

```
"search": {  
  "value": "linux",  
  "type": "exact"  
}
```

## Extended update and delete methods

- ◆ Additional parameters will allow to execute more complex requests, like:
  - “Delete all events older than one month”
  - “Reduce the update interval of all unsupported items”

## Batch API calls

- ◆ The JSON-RPC 2.0 standard support multiple method calls in a single request

```
// batch request
[
  {"jsonrpc": "2.0", "method": "host.get", ...},
  {"jsonrpc": "2.0", "method": "item.get", ...}
]
// response
[
  {"jsonrpc": "2.0", "result": { ... }},
  {"jsonrpc": "2.0", "result": { ... }}
]
```

**CONCEPT**

# ZABBIX

The next generation API  
will be:

Fully documented

Easy to learn and use

More flexible and reliable



Next generation  
API

**Thank you**