

# itemprototype.get

## Description

integer/array `itemprototype.get(object parameters)`

The method allows to retrieve item prototypes according to the given parameters.

## Parameters

(object) Parameters defining the desired output.

The method supports the following parameters.

Parameter	Type	Description
discoveryids	string/array	Return only item prototypes that belong to the given LLD rules.
graphids	string/array	Return only item prototypes that are used in the given graph prototypes.
hostids	string/array	Return only item prototypes that belong to the given hosts.
inherited	boolean	If set to <code>true</code> return only item prototypes inherited from a template.
itemids	string/array	Return only item prototypes with the given IDs.
monitored	boolean	If set to <code>true</code> return only enabled item prototypes that belong to monitored hosts.
templated	boolean	If set to <code>true</code> return only item prototypes that belong to templates.
templateids	string/array	Return only item prototypes that belong to the given templates.
triggerids	string/array	Return only item prototypes that are used in the given trigger prototypes.
selectApplications	query	Return an <a href="#">applications</a> property with applications that the item prototype belongs to.
selectApplicationPrototypes	query	Return <a href="#">applicationPrototypes</a> property with application prototypes linked to item prototype.
selectDiscoveryRule	query	Return a <a href="#">discoveryRule</a> property with the low-level discovery rule that the graph prototype belongs to.
selectGraphs	query	Return a <a href="#">graph_prototype</a> property with graph prototypes that the item prototype is used in. Supports count.
selectHosts	query	Return a <a href="#">hosts</a> property with an array of hosts that the item prototype belongs to.
selectTriggers	query	Return a <a href="#">triggers</a> property with trigger prototypes that the item prototype is used in. Supports count.

Parameter	Type	Description
selectPreprocessing	query	<p>Return a <a href="#">preprocessing</a> property with item preprocessing options.</p> <p>It has the following properties:</p> <p>type - (string) The preprocessing option type:</p> <ul style="list-style-type: none"> <li>1 - Custom multiplier;</li> <li>2 - Right trim;</li> <li>3 - Left trim;</li> <li>4 - Trim;</li> <li>5 - Regular expression matching;</li> <li>6 - Boolean to decimal;</li> <li>7 - Octal to decimal;</li> <li>8 - Hexadecimal to decimal;</li> <li>9 - Simple change;</li> <li>10 - Change per second;</li> <li>11 - XML XPath;</li> <li>12 - JSONPath;</li> <li>13 - In range;</li> <li>14 - Matches regular expression;</li> <li>15 - Does not match regular expression;</li> <li>16 - Check for error in JSON;</li> <li>17 - Check for error in XML;</li> <li>18 - Check for error using regular expression;</li> <li>19 - Discard unchanged;</li> <li>20 - Discard unchanged with heartbeat;</li> <li>21 - JavaScript;</li> <li>22 - Prometheus pattern;</li> <li>23 - Prometheus to JSON;</li> <li>24 - CSV to JSON;</li> <li>25 - Replace.</li> </ul> <p>params - (string) Additional parameters used by preprocessing option. Multiple parameters are separated by LF (\n)character.</p> <p>error_handler - (string) Action type used in case of preprocessing step failure:</p> <ul style="list-style-type: none"> <li>0 - Error message is set by Zabbix server;</li> <li>1 - Discard value;</li> <li>2 - Set custom value;</li> <li>3 - Set custom error message.</li> </ul> <p>error_handler_params - (string) Error handler parameters.</p>
filter	object	<p>Return only those results that exactly match the given filter.</p> <p>Accepts an array, where the keys are property names, and the values are either a single value or an array of values to match against.</p> <p>Supports additional filters:</p> <p>host - technical name of the host that the item prototype belongs to.</p>

Parameter	Type	Description
limitSelects	integer	Limits the number of records returned by subselects.  Applies to the following subselects: selectGraphs - results will be sorted by name; selectTriggers - results will be sorted by description.
sortfield	string/array	Sort the result by the given properties.  Possible values are: itemid, name, key_, delay, type and status.
countOutput	boolean	These parameters being common for all get methods are described in detail in the <a href="#">reference commentary</a> .
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

## Return values

(integer/array) Returns either:

- an array of objects;
- the count of retrieved objects, if the countOutput parameter has been used.

## Examples

### Retrieving item prototypes from an LLD rule

Retrieve all item prototypes from an LLD rule.

Request:

```
{
  "jsonrpc": "2.0",
  "method": "itemprototype.get",
  "params": {
    "output": "extend",
    "discoveryids": "27426"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

## Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "itemid": "23077",
      "type": "0",
      "snmp_oid": "",
      "hostid": "10079",
      "name": "Incoming network traffic on $1",
      "key_": "net.if.in[en0]",
      "delay": "1m",
      "history": "1w",
      "trends": "365d",
      "status": "0",
      "value_type": "3",
      "trapper_hosts": "",
      "units": "bps",
      "formula": "",
      "error": "",
      "logtimefmt": "",
      "templateid": "0",
      "valuemapid": "0",
      "params": "",
      "ipmi_sensor": "",
      "authtype": "0",
      "username": "",
      "password": "",
      "publickey": "",
      "privatekey": "",
      "flags": "0",
      "interfaceid": "0",
      "description": "",
      "inventory_link": "0",
      "lifetime": "30d",
      "state": "0",
      "evaltype": "0",
      "jmx_endpoint": "",
      "master_itemid": "0",
      "timeout": "3s",
      "url": "",
      "query_fields": [],
      "posts": "",
      "status_codes": "200",
      "follow_redirects": "1",
      "post_type": "0",
      "http_proxy": "",
      "headers": []
    }
  ]
}
```

```
    "retrieve_mode": "0",
    "request_method": "0",
    "output_format": "0",
    "ssl_cert_file": "",
    "ssl_key_file": "",
    "ssl_key_password": "",
    "verify_peer": "0",
    "verify_host": "0",
    "allow_traps": "0",
    "lastclock": "0",
    "lastns": "0",
    "lastvalue": "0",
    "prevvalue": "0",
    "discover": "0"
  },
  {
    "itemid": "10010",
    "type": "0",
    "snmp_oid": "",
    "hostid": "10001",
    "name": "Processor load (1 min average per core)",
    "key_": "system.cpu.load[percpu,avg1]",
    "delay": "1m",
    "history": "1w",
    "trends": "365d",
    "status": "0",
    "value_type": "0",
    "trapper_hosts": "",
    "units": "",
    "formula": "",
    "error": "",
    "logtimefmt": "",
    "templateid": "0",
    "valuemapid": "0",
    "params": "",
    "ipmi_sensor": "",
    "authtype": "0",
    "username": "",
    "password": "",
    "publickey": "",
    "privatekey": "",
    "flags": "0",
    "interfaceid": "0",
    "description": "The processor load is calculated as system CPU
load divided by number of CPU cores.",
    "inventory_link": "0",
    "lifetime": "0",
    "state": "0",
    "evaltype": "0",
    "jmx_endpoint": "",
    "master_itemid": "0",
```

```
    "timeout": "3s",
    "url": "",
    "query_fields": [],
    "posts": "",
    "status_codes": "200",
    "follow_redirects": "1",
    "post_type": "0",
    "http_proxy": "",
    "headers": [],
    "retrieve_mode": "0",
    "request_method": "0",
    "output_format": "0",
    "ssl_cert_file": "",
    "ssl_key_file": "",
    "ssl_key_password": "",
    "verify_peer": "0",
    "verify_host": "0",
    "allow_traps": "0",
    "lastclock": "0",
    "lastns": "0",
    "lastvalue": "0",
    "prevvalue": "0",
    "discover": "0"
  },
  "id": 1
}
```

## Finding dependent item

Find one Dependent item for item with ID "25545".

Request:

```
{
  "jsonrpc": "2.0",
  "method": "item.get",
  "params": {
    "output": "extend",
    "filter": {
      "type": "18",
      "master_itemid": "25545"
    },
    "limit": "1"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

```
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "itemid": "25547",
      "type": "18",
      "snmp_oid": "",
      "hostid": "10116",
      "name": "Seconds",
      "key_": "apache.status.uptime.seconds",
      "delay": "0",
      "history": "90d",
      "trends": "365d",
      "status": "0",
      "value_type": "3",
      "trapper_hosts": "",
      "units": "",
      "formula": "",
      "error": "",
      "logtimefmt": "",
      "templateid": "0",
      "valuemapid": "0",
      "params": "",
      "ipmi_sensor": "",
      "authtype": "0",
      "username": "",
      "password": "",
      "publickey": "",
      "privatekey": "",
      "flags": "0",
      "interfaceid": "0",
      "description": "",
      "inventory_link": "0",
      "lifetime": "30d",
      "state": "0",
      "evaltype": "0",
      "master_itemid": "25545",
      "jmx_endpoint": "",
      "master_itemid": "0",
      "timeout": "3s",
      "url": "",
      "query_fields": [],
      "posts": "",
      "status_codes": "200",
      "follow_redirects": "1",
      "post_type": "0",
    }
  ]
}
```

```
    "http_proxy": "",
    "headers": [],
    "retrieve_mode": "0",
    "request_method": "0",
    "output_format": "0",
    "ssl_cert_file": "",
    "ssl_key_file": "",
    "ssl_key_password": "",
    "verify_peer": "0",
    "verify_host": "0",
    "allow_traps": "0",
    "lastclock": "0",
    "lastns": "0",
    "lastvalue": "0",
    "prevvalue": "0",
    "discover": "0"
  },
  "id": 1
}
```

## Find HTTP agent item prototype

Find HTTP agent item prototype with request method HEAD for specific host id.

Request:

```
{
  "jsonrpc": "2.0",
  "method": "itemprototype.get",
  "params": {
    "hostids": "10254",
    "filter": {
      "type": "19",
      "request_method": "3"
    }
  },
  "id": 17,
  "auth": "d678e0b85688ce578ff061bd29a20d3b"
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
```



```
{
  "itemid": "28257",
  "type": "19",
  "snmp_oid": "",
  "hostid": "10254",
  "name": "discovered",
  "key_": "item[#{INAME}]",
  "delay": "#{IUPDATE}",
  "history": "90d",
  "trends": "30d",
  "status": "0",
  "value_type": "3",
  "trapper_hosts": "",
  "units": "",
  "formula": "",
  "error": "",
  "logtimefmt": "",
  "templateid": "28255",
  "valuemapid": "0",
  "params": "",
  "ipmi_sensor": "",
  "authtype": "0",
  "username": "",
  "password": "",
  "publickey": "",
  "privatekey": "",
  "flags": "2",
  "interfaceid": "2",
  "description": "",
  "inventory_link": "0",
  "lifetime": "30d",
  "state": "0",
  "evaltype": "0",
  "jmx_endpoint": "",
  "master_itemid": "0",
  "timeout": "3s",
  "url": "#{IURL}",
  "query_fields": [],
  "posts": "",
  "status_codes": "",
  "follow_redirects": "0",
  "post_type": "0",
  "http_proxy": "",
  "headers": [],
  "retrieve_mode": "0",
  "request_method": "3",
  "output_format": "0",
  "ssl_cert_file": "",
  "ssl_key_file": "",
  "ssl_key_password": "",
  "verify_peer": "0",
```

```
        "verify_host": "0",
        "allow_traps": "0",
        "discover": "0"
    },
    "id": 17
}
```

## See also

- [Application](#)
- [Host](#)
- [Graph prototype](#)
- [Trigger prototype](#)

## Source

`CItemPrototype::get()` in `ui/include/classes/api/services/CItemPrototype.php`.

From:

<https://www.zabbix.com/documentation/current/> - **Zabbix Documentation 5.0**

Permanent link:

<https://www.zabbix.com/documentation/current/manual/api/reference/itemprototype/get>

Last update: **2020/05/20 09:19**

