

discoveryrule.create

Description

object discoveryrule.create(object/array **LldRules**)

This method allows to create new LLD rules.

Parameters

(object/array) LLD rules to create.

Additionally to the [standard LLD rule properties](#), the method accepts the following parameters.

Parameter	Type	Description
filter	object	LLD rule filter object for the LLD rule.
preprocessing	array	LLD rule preprocessing options.

Return values

(object) Returns an object containing the IDs of the created LLD rules under the `itemids` property. The order of the returned IDs matches the order of the passed LLD rules.

Examples

Creating an LLD rule

Create a Zabbix agent LLD rule to discover mounted file systems. Discovered items will be updated every 30 seconds.

Request:

```
{
  "jsonrpc": "2.0",
  "method": "discoveryrule.create",
  "params": {
    "name": "Mounted filesystem discovery",
    "key_": "vfs.fs.discovery",
    "hostid": "10197",
    "type": "0",
    "interfaceid": "112",
    "delay": "30s"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

```
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "27665"
    ]
  },
  "id": 1
}
```

Using a filter

Create an LLD rule with a set of conditions to filter the results by. The conditions will be grouped together using the logical “and” operator.

Request:

```
{
  "jsonrpc": "2.0",
  "method": "discoveryrule.create",
  "params": {
    "name": "Filtered LLD rule",
    "key_": "lld",
    "hostid": "10116",
    "type": "0",
    "interfaceid": "13",
    "delay": "30s",
    "filter": {
      "evaltype": 1,
      "conditions": [
        {
          "macro": "#{MACRO1}",
          "value": "@regex1"
        },
        {
          "macro": "#{MACRO2}",
          "value": "@regex2"
        },
        {
          "macro": "#{MACRO3}",
          "value": "@regex3"
        }
      ]
    }
  }
}
```

```
    },
    "auth": "038e1d7b1735c6a5436ee9eae095879e",
    "id": 1
  }
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "27665"
    ]
  },
  "id": 1
}
```

Creating a LLD rule with macro paths

Request:

```
{
  "jsonrpc": "2.0",
  "method": "discoveryrule.create",
  "params": {
    "name": "LLD rule with LLD macro paths",
    "key_": "lld",
    "hostid": "10116",
    "type": "0",
    "interfaceid": "13",
    "delay": "30s",
    "lld_macro_paths": [
      {
        "lld_macro": "{#MACRO1}",
        "path": "$.path.1"
      },
      {
        "lld_macro": "{#MACRO2}",
        "path": "$.path.2"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "27665"
    ]
  },
  "id": 1
}
```

Using a custom expression filter

Create an LLD rule with a filter that will use a custom expression to evaluate the conditions. The LLD rule must only discover objects the “{#MACRO1}” macro value of which matches both regular expression “regex1” and “regex2”, and the value of “{#MACRO2}” matches either “regex3” or “regex4”. The formula IDs “A”, “B”, “C” and “D” have been chosen arbitrarily.

Request:

```
{
  "jsonrpc": "2.0",
  "method": "discoveryrule.create",
  "params": {
    "name": "Filtered LLD rule",
    "key_": "lld",
    "hostid": "10116",
    "type": "0",
    "interfaceid": "13",
    "delay": "30s",
    "filter": {
      "evaltype": 3,
      "formula": "(A and B) and (C or D)",
      "conditions": [
        {
          "macro": "{#MACRO1}",
          "value": "@regex1",
          "formulaid": "A"
        },
        {
          "macro": "{#MACRO1}",
          "value": "@regex2",
          "formulaid": "B"
        },
        {
          "macro": "{#MACRO2}",
          "value": "@regex3",
          "formulaid": "C"
        }
      ]
    }
  }
}
```

```
        {
            "macro": "{#MACRO2}",
            "value": "@regex4",
            "formulaid": "D"
        }
    ],
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "27665"
    ]
  },
  "id": 1
}
```

Using custom query fields and headers

Create LLD rule with custom query fields and headers.

Request:

```
{
  "jsonrpc": "2.0",
  "method": "discoveryrule.create",
  "params": {
    "hostid": "10257",
    "interfaceid": "5",
    "type": "19",
    "name": "API HTTP agent",
    "key_": "api_discovery_rule",
    "value_type": "3",
    "delay": "5s",
    "url": "http://127.0.0.1?discoverer.php",
    "query_fields": [
      {
        "mode": "json"
      },
      {
        "elements": "2"
      }
    ]
  }
}
```

```
    },
    ],
    "headers": {
      "X-Type": "api",
      "Authorization": "Bearer mF_A.B5f-2.1JcM"
    },
    "allow_traps": "1",
    "trapper_hosts": "127.0.0.1",
    "id": 35,
    "auth": "d678e0b85688ce578ff061bd29a20d3b",
  }
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "28336"
    ]
  },
  "id": 35
}
```

Creating a LLD rule with preprocessing

Request:

```
{
  "jsonrpc": "2.0",
  "method": "discoveryrule.create",
  "params": {
    "name": "Discovery rule with preprocessing",
    "key_": "lld.with.preprocessing",
    "hostid": "10001",
    "ruleid": "27665",
    "type": 0,
    "value_type": 3,
    "delay": "60s",
    "interfaceid": "1155",
    "preprocessing": [
      {
        "type": "20",
        "params": "20",
        "error_handler": "0",
      }
    ]
  }
}
```

```
        "error_handler_params": ""
    }
  ],
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "44211"
    ]
  },
  "id": 1
}
```

See also

- [LLD rule filter](#)
- [LLD macro paths](#)
- [LLD rule preprocessing](#)

Source

`CDiscoveryRule::create()` in *frontends/php/include/classes/api/services/CDiscoveryRule.php*.

From: <https://www.zabbix.com/documentation/4.4/> - **Zabbix Documentation 4.4**

Permanent link: <https://www.zabbix.com/documentation/4.4/manual/api/reference/discoveryrule/create?rev=1582890317>

Last update: **2020/02/28 11:45**

