



Retrieve events

ONTAP 9.15.1 REST API reference

NetApp
September 11, 2024

Table of Contents

- Retrieve events 1
- Support EMS events endpoint overview 1
- Retrieve observed events 3

Retrieve events

Support EMS events endpoint overview

Overview

Queries a live collection of observed events on the system. Note: The `filter.name` parameter is used to pass a filter to be applied to the event collection returned. To retrieve the list of available filters, refer to </support/ems/filters>

Example

Querying for the latest event received by EMS

```
# The API:
GET /api/support/ems/events

# The call:
curl -X GET "https://<mgmt-
ip>/api/support/ems/events?fields=message.name&max_records=1" -H "accept:
application/hal+json"

# The response:
200 OK

# JSON Body
{
  "records": [
    {
      "node": {
        "name": "node1",
        "uuid": "f087b8e3-99ac-11e8-b5a5-005056bb4ec7",
        "_links": {
          "self": {
            "href": "/api/cluster/nodes/f087b8e3-99ac-11e8-b5a5-
005056bb4ec7"
          }
        }
      },
      "index": 661,
      "message": {
        "name": "raid.aggr.log.CP.count"
      },
      "_links": {
        "self": {
          "href": "/api/support/ems/events/node1/661"
        }
      }
    }
  ],
  "num_records": 1,
  "_links": {
    "self": {
      "href": "/api/support/ems/events?fields=message.name&max_records=1"
    }
  }
}
```

Retrieve observed events

GET /support/ems/events

Introduced In: 9.6

Retrieves a collection of observed events.

Related ONTAP commands

- `event log show`

Parameters

| Name | Type | In | Required | Description |
|------------------|---------|-------|----------|--|
| time | string | query | False | Filter by time |
| message.name | string | query | False | Filter by message.name |
| message.severity | string | query | False | Filter by message.severity |
| parameters.name | string | query | False | Filter by parameters.name |
| parameters.value | string | query | False | Filter by parameters.value |
| index | integer | query | False | Filter by index |
| log_message | string | query | False | Filter by log_message |
| node.uuid | string | query | False | Filter by node.uuid |
| node.name | string | query | False | Filter by node.name |
| source | string | query | False | Filter by source |
| filter.name | string | query | False | Filter the collection returned using an event filter • Introduced in: 9.8 |

| Name | Type | In | Required | Description |
|----------------|---------------|-------|----------|---|
| fields | array[string] | query | False | Specify the fields to return. |
| max_records | integer | query | False | Limit the number of records returned. |
| return_records | boolean | query | False | The default is true for GET calls. When set to false, only the number of records is returned. <ul style="list-style-type: none"> • Default value: 1 |
| return_timeout | integer | query | False | The number of seconds to allow the call to execute before returning. When iterating over a collection, the default is 15 seconds. ONTAP returns earlier if either max records or the end of the collection is reached. <ul style="list-style-type: none"> • Default value: 1 • Max value: 120 • Min value: 0 |
| order_by | array[string] | query | False | Order results by specified fields and optional [asc |

Response

Status: 200, Ok

| Name | Type | Description |
|-------------|----------------------------------|-------------------|
| _links | _links | |
| num_records | integer | Number of records |
| records | array[records] | |

Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "num_records": 1,
  "records": [
    {
      "_links": {
        "action": [
          {
            "confirmation_message": {
              "arguments": [
                {
                  "code": "string",
                  "message": "string"
                }
              ],
              "code": "4",
              "message": "entry doesn't exist"
            },
            "description": {
              "arguments": [
                {
                  "code": "string",
                  "message": "string"
                }
              ],
              "code": "4",
              "message": "entry doesn't exist"
            }
          ],
          "href": "/api/resourcelink",
          "method": "PATCH",
          "name": "schedule",
          "parameters": [
            {
              "description": {
                "arguments": [
                  {
                    "code": "string",
```

```

        "message": "string"
    }
  ],
  "code": "4",
  "message": "entry doesn't exist"
},
"enum": [
  "value-1",
  "value-2"
],
"format": "date-time",
"help": {
  "arguments": [
    {
      "code": "string",
      "message": "string"
    }
  ],
  "code": "4",
  "message": "entry doesn't exist"
},
"in": "body",
"maxItems": 0,
"maxLength": 0,
"maximum": 0,
"minItems": 0,
"minLength": 0,
"minimum": 0,
"multipleOf": 0,
"name": "schedule-at",
"optional": [
  "end-time"
],
"pattern": "string",
"title": {
  "arguments": [
    {
      "code": "string",
      "message": "string"
    }
  ],
  "code": "4",
  "message": "entry doesn't exist"
},
"type": "string",
"validation_error_message": {

```



```

        "arguments": [
            {
                "code": "string",
                "message": "string"
            }
        ],
        "code": "4",
        "message": "entry doesn't exist"
    }
}
],
"title": {
    "arguments": [
        {
            "code": "string",
            "message": "string"
        }
    ],
    "code": "4",
    "message": "entry doesn't exist"
}
}
],
"self": {
    "href": "/api/resourcelink"
}
},
"creation_time": "string",
"history": [
    {
        "log_message": "string",
        "state": "resolving",
        "update_time": "string"
    }
],
"index": 1,
"last_update_time": "string",
"log_message": "string",
"message": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
},
"name": "callhome.spares.low",
"severity": "emergency"

```

```

    },
    "node": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "node1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "num_occurrences": 0,
    "parameters": [
      {
        "name": "numOps",
        "value": 123
      }
    ],
    "source": "string",
    "state": "resolving",
    "time": "string",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  }
]
}

```

Error

Status: Default

ONTAP Error Response Codes

| Error Code | Description |
|------------|---------------------------------|
| 983093 | The provided filter is unknown. |

Also see the table of common errors in the [Response body](#) overview section of this documentation.

| Name | Type | Description |
|-------|--------------------------------|-------------|
| error | returned_error | |

Example error

```
{
  "error": {
    "arguments": [
      {
        "code": "string",
        "message": "string"
      }
    ],
    "code": "4",
    "message": "entry doesn't exist",
    "target": "uuid"
  }
}
```

Definitions

See Definitions

href

| Name | Type | Description |
|------|--------|-------------|
| href | string | |

_links

| Name | Type | Description |
|------|----------------------|-------------|
| next | href | |
| self | href | |

arguments

| Name | Type | Description |
|---------|--------|------------------|
| code | string | Argument code |
| message | string | Message argument |

confirmation_message

Confirmation message that is presented in user facing applications, confirming the action to take.

| Name | Type | Description |
|-----------|------------------------------------|----------------------|
| arguments | array[arguments] | Message arguments |
| code | string | Unique message code. |
| message | string | User message. |

description

Description of the action is presented in user facing applications.

| Name | Type | Description |
|-----------|------------------------------------|----------------------|
| arguments | array[arguments] | Message arguments |
| code | string | Unique message code. |
| message | string | User message. |

description

Description of the parameter that is presented in user facing applications.

| Name | Type | Description |
|-----------|------------------|----------------------|
| arguments | array[arguments] | Message arguments |
| code | string | Unique message code. |
| message | string | User message. |

help

Help message of the parameter that is presented in user facing applications.

| Name | Type | Description |
|-----------|------------------|----------------------|
| arguments | array[arguments] | Message arguments |
| code | string | Unique message code. |
| message | string | User message. |

items

If the type of the parameter is an array, this specifies the type of items in the form of a JSON object where other properties applicable to that type can be included.

properties

If the type of the parameter is an object, this specifies what properties make up the object in the form of a JSON array where multiple parameters can be embedded within a single parameter. It is primarily used as a schema for an array type parameter.

title

Title of the parameter that is presented in user facing applications.

| Name | Type | Description |
|-----------|------------------|----------------------|
| arguments | array[arguments] | Message arguments |
| code | string | Unique message code. |
| message | string | User message. |

validation_error_message

Error message that is presented in user facing applications, in cases where parameter validation fails.

| Name | Type | Description |
|-----------|------------------|----------------------|
| arguments | array[arguments] | Message arguments |
| code | string | Unique message code. |
| message | string | User message. |

value

Specifies the current value(s) for the parameter encoded in the appropriate JSON type.

parameters

| Name | Type | Description |
|------------------|---------------|---|
| description | description | Description of the parameter that is presented in user facing applications. |
| enum | array[string] | Specifies the possible values of the parameter. |
| exclusiveMaximum | boolean | Specifies whether the "maximum" value is excluded in the parameter value range. |
| exclusiveMinimum | boolean | Specifies whether the "minimum" value is excluded in the parameter value range. |
| format | string | An optional modifier that serves as a hint at the content and format of the parameter. |
| help | help | Help message of the parameter that is presented in user facing applications. |
| in | string | Specifies where the parameter is placed when invoking the action. |
| items | items | If the type of the parameter is an array, this specifies the type of items in the form of a JSON object where other properties applicable to that type can be included. |

| Name | Type | Description |
|------------|----------------------------|---|
| maxItems | integer | Specifies the maximum length of an array type parameter. |
| maxLength | integer | Specifies the maximum length of a string type parameter. |
| maximum | integer | Specifies the maximum value of the parameter. |
| minItems | integer | Specifies the minimum length of an array type parameter. |
| minLength | integer | Specifies the minimum length of a string type parameter. |
| minimum | integer | Specifies the minimum value of the parameter. |
| multipleOf | number | Specifies that a number type parameter must be the multiple of this number. |
| name | string | Parameter name. |
| optional | array[string] | By default, all properties of an object type parameter are mandatory. This property specifies the list of optional properties. |
| pattern | string | Specifies a regular expression template for a string type parameter. |
| properties | properties | If the type of the parameter is an object, this specifies what properties make up the object in the form of a JSON array where multiple parameters can be embedded within a single parameter. It is primarily used as a schema for an array type parameter. |
| title | title | Title of the parameter that is presented in user facing applications. |

| Name | Type | Description |
|--------------------------|--|---|
| type | string | Parameter type. |
| uniqueItems | boolean | Specifies whether the "maximum" value is excluded in the parameter value range. |
| validation_error_message | validation_error_message | Error message that is presented in user facing applications, in cases where parameter validation fails. |
| value | value | Specifies the current value(s) for the parameter encoded in the appropriate JSON type. |

request_body_template

JSON object describing the structure of the request body if arguments must be provided in the body when invoking the action. A JSON string value that takes the form of {parameter-name} must be substituted by user input values encoded as the appropriate JSON type. The following table gives examples where the parameter type is a string, an integer, and an array:

| request_body_template value | parameter value | request body |
|-----------------------------|------------------------------|---------------------------------|
| {"name": "{user-name}"} | user-name="Joe" | {"name": "Joe"} |
| {"retry count": "{count}"} | count=10 | {"retry_count": 10} |
| {"domains": "{dns-names}"} | dns-names=["dns-1", "dns-2"] | {"domains": ["dns-1", "dns-2"]} |

Only JSON string values that start with a '{' and end with a '}' should be considered for parameter substitutions. A JSON string value such as "{user-name} is the syntax" should be treated as a literal string with no parameter substitution to be performed. Double curly braces '{{' and '}}' are used to escape parameter substitutions, therefore double curly braces must be converted to single curly braces. For example, "{{{user-name}}}" is converted to the JSON string value "{user-name}". Note that this rule only applies if a JSON string starts with '{{' and ends with '}}'. The following table provides examples of when parameter substitutions must not be performed:

| request_body_template value | request body |
|------------------------------------|------------------------------------|
| {"name": "{user-name} is bad"} | {"name": "{user-name} is bad"} |
| {"name": "{{{user-name}}}"} | {"name": "{user-name}"} |
| {"name": "{{{user-name}}} is bad"} | {"name": "{{{user-name}}} is bad"} |
| {"name": ">{{{user-name}}}"} | {"name": ">{{{user-name}}}"} |

title

Title of the action that is presented in user facing applications.

| Name | Type | Description |
|-----------|------------------|----------------------|
| arguments | array[arguments] | Message arguments |
| code | string | Unique message code. |
| message | string | User message. |

action

| Name | Type | Description |
|-----------------------|-----------------------|--|
| confirmation_message | confirmation_message | Confirmation message that is presented in user facing applications, confirming the action to take. |
| description | description | Description of the action is presented in user facing applications. |
| href | string | URI on which to perform the action, using the HTTP method specified in the method property. |
| method | string | HTTP verb, such as PATCH, POST, used to perform the action. |
| name | string | Name of the action. |
| parameters | array[parameters] | Parameter list for the action. |
| request_body_template | request_body_template | JSON object describing the structure of the request body if arguments must be provided in the body when invoking the action. A JSON string value that takes the form of {parameter-name} must be substituted by user input values encoded as the appropriate JSON type. The following table gives examples where the parameter type is a string, an integer, and an array: |

| request_body_template value | parameter value | request body

| {"name": "{user-name}"} | user-name="Joe" | {"name": "Joe"}

```
| {"retry count": "{count}"} | count=10 | {"retry_count": 10}
```

```
| {"domains": "{dns-names}"} | dns-names=["dns-1", "dns-2"] | {"domains": ["dns-1", "dns-2"]}
```

Only JSON string values that start with a '{' and end with a '}' should be considered for parameter substitutions. A JSON string value such as "{user-name} is the syntax" should be treated as a literal string with no parameter substitution to be performed. Double curly braces '{{' and '}}' are used to escape parameter substitutions, therefore double curly braces must be converted to single curly braces. For example, "{{{user-name}}}" is converted to the JSON string value "{user-name}". Note that this rule only applies if a JSON string starts with '{{' and ends with '}}'. The following table provides examples of when parameter substitutions must not be performed:

| request_body_template value | request body

```
| {"name": "{user-name} is bad"} | {"name": "{user-name} is bad"}
```

```
| {"name": "{{user-name}}"} | {"name": "{user-name}"}
```

```
| {"name": "{{{user-name}} is bad"} | {"name": "{{user-name}} is bad"}
```

```
| {"name": "{{{{{user-name}}}}"} | {"name": "{{user-name}}"}}
```

| |
|--|
| title |
| title |
| Title of the action that is presented in user facing applications. |

self

| Name | Type | Description |
|------|--------|-------------|
| href | string | |

_links

| Name | Type | Description |
|--------|---------------------------------|--|
| action | array[action] | Actions that can be performed to remediate the issue that the event instance is reporting. To perform an action, send a request to the href URL using the HTTP method with an appropriate set of parameter values. |
| self | self | |

action

Some automatic remediation actions included in the `action` link relations are links to the event instance itself, while others are to other public REST APIs. In cases where the `action` link points to the event instance itself, this property is used in a PATCH request body to trigger a remediation action. The value

included as the value of the property depends on each remediation action.

history

| Name | Type | Description |
|-------------|--------|--|
| log_message | string | A formatted text string about the update. |
| state | string | State of the event instance when the update is raised. |
| update_time | string | Timestamp of the update. |

_links

| Name | Type | Description |
|------|----------------------|-------------|
| self | href | |

message

| Name | Type | Description |
|----------|------------------------|---|
| _links | _links | |
| name | string | Message name of the event. Returned by default. |
| severity | string | Severity of the event. Returned by default. |

node

| Name | Type | Description |
|--------|------------------------|-------------|
| _links | _links | |
| name | string | |
| uuid | string | |

parameters

| Name | Type | Description |
|-------|--------|--------------------|
| name | string | Name of parameter |
| value | string | Value of parameter |

records

| Name | Type | Description |
|-------------------------------|-------------------------------|---|
| <code>_links</code> | <code>_links</code> | |
| <code>action</code> | <code>action</code> | Some automatic remediation actions included in the <code>action</code> link relations are links to the event instance itself, while others are to other public REST APIs. In cases where the <code>action</code> link points to the event instance itself, this property is used in a PATCH request body to trigger a remediation action. The value included as the value of the property depends on each remediation action. |
| <code>actionable</code> | boolean | Indicates whether the event instance currently has actions that can be performed. |
| <code>creation_time</code> | string | Timestamp of the event creation. |
| <code>history</code> | array[<code>history</code>] | History of updates to the event instance. |
| <code>index</code> | integer | Index of the event. Returned by default. |
| <code>last_update_time</code> | string | Timestamp of the last update to the event instance. |
| <code>log_message</code> | string | A formatted text string populated with parameter details. Returned by default. When changing the state of an event instance, include the 'state' property and this property in a PATCH request. The value of this property should describe why the state is changed. |
| <code>message</code> | <code>message</code> | |
| <code>node</code> | <code>node</code> | |
| <code>num_occurrences</code> | integer | Number of times the event has repeatedly reported the issue. |

| Name | Type | Description |
|------------|-------------------------------------|--|
| parameters | array[parameters] | A list of parameters provided with the EMS event. |
| source | string | Source |
| state | string | Indicates the event state. A stateful event tracks the state changes of a system condition and reflects the current system condition. This property and the 'action' property are mutually exclusive in a PATCH request. The state of an event instance can be changed to 'resolved' or 'closed' by sending a PATCH request with this property along with the 'log_message' property describing the reason. An event instance changes its state to 'resolving' only as a result of the user triggering an automatic remediation action, which is done by invoking an action specified in the 'action' link relation. |
| stateful | boolean | Indicates whether the event is stateful. A stateful event tracks the state changes of a system condition and reflects the current system condition while a stateless event simply reports a certain system condition that has occurred sometime in the past. |
| time | string | Timestamp of the event. Returned by default. |
| uuid | string | The UUID that uniquely identifies the event instance. |

error_arguments

| Name | Type | Description |
|---------|--------|------------------|
| code | string | Argument code |
| message | string | Message argument |

returned_error

| Name | Type | Description |
|-----------|--|---|
| arguments | array[error_arguments] | Message arguments |
| code | string | Error code |
| message | string | Error message |
| target | string | The target parameter that caused the error. |

Copyright information

Copyright © 2024 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.