Splunk® Enterprise REST API Reference Manual 7.3.2

Using the REST API reference

Generated: 10/03/2019 9:43 am
Using the REST API reference

Use the REST API Reference to learn about available endpoints and operations for accessing, creating, updating, or deleting resources. See the REST API User Manual to learn about the Splunk REST API basic concepts.

See the Endpoints reference list for an alphabetical list of endpoints.

Splunk Cloud REST API usage

There are some REST API access and usage differences between Splunk Cloud and Splunk Enterprise. If you are using Splunk Cloud, review details in Using the REST API with Splunk Cloud.

Splunk REST API admin endpoints

Splunk does not support or document REST API endpoints that contain /admin/ in their URIs. Use the corresponding publicly documented endpoint instead.

Resource groups

Resources are grouped into the following categories.

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access control</td>
<td>Authorize and authenticate users.</td>
</tr>
<tr>
<td>Applications</td>
<td>Install applications and application templates.</td>
</tr>
<tr>
<td>Clusters</td>
<td>Configure and manage indexer clusters and search head clusters.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Manage configuration files and settings.</td>
</tr>
<tr>
<td>Deployment</td>
<td>Manage deployment servers and clients.</td>
</tr>
<tr>
<td>Inputs</td>
<td>Manage data input.</td>
</tr>
<tr>
<td>Introspection</td>
<td>Access system properties.</td>
</tr>
<tr>
<td>Knowledge</td>
<td>Define indexed and searched data configurations.</td>
</tr>
<tr>
<td>KV store</td>
<td>Manage app key-value store.</td>
</tr>
<tr>
<td>Licensing</td>
<td>Manage licensing configurations.</td>
</tr>
<tr>
<td>Outputs</td>
<td>Manage forwarder data configuration.</td>
</tr>
<tr>
<td>Search</td>
<td></td>
</tr>
</tbody>
</table>
Manage searches and search-generated alerts and view objects.

**Category Description**

**System**
Manage server configuration.

**Workload management**
Manage system resources for search workloads.

See the Endpoints reference list for an alphabetical list of endpoints.

**Available operations**

Depending on the endpoint, GET, POST, and/or DELETE operations are available for accessing, creating, updating, or deleting resources. Some operations have specific capability requirements, as noted.

**Using endpoint reference entries**

Reference information for each endpoint in the REST API includes the following items.

- **URL**
- **Usage details**
- **Expandable elements showing available operations (GET, POST, and/or DELETE) for the endpoint.**

Expand a GET, POST, or DELETE element to show the following usage information about the operation.

- **Request parameter information and requirements.**
- **Returned values included in the response.**
- **Example request and response.**

**Request and response details**

**Pagination and filtering parameters**

In addition to the parameters specific to each endpoint and operation, the following request parameters are valid for some GET methods.

<table>
<thead>
<tr>
<th>Name</th>
<th>Datatype</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
</table>

2
### count
- **Type:** Number
- **Value:** 30
- **Description:** Maximum number of entries to return. Set value to 0 to get all available entries.

Filters the response to include only the named values. Specify multiple times to return multiple values.

**Examples:**
- \( f=\text{qualifiedSearch} \) returns only the value for \text{qualifiedSearch}
- \( f=s* \) returns all the values that have names beginning with \text{s}
- \( f=\text{qualifiedSearch}&f=\text{is_visible} \) returns the values for \text{qualifiedSearch} as well as \text{is_visible}

### offset
- **Type:** Number
- **Value:** 0
- **Description:** Index of first item to return.

Response filter, where the response field values are matched against this search expression.

**Example:**
- \( \text{search}=\text{foo} \) matches on any field with the string \text{foo} in the name.
- \( \text{search}=	ext{field_name}%3D\text{field_value} \) restricts the match to a single field. (Requires URIs-encoding.)

### search
- **Type:** String

**Examples:**
- \( \text{search}=\text{foo} \) matches on any field with the string \text{foo} in the name.
- \( \text{search}=%\text{field_name}%3D\text{field_value} \) restricts the match to a single field. (Requires URI-encoding.)

### sort_dir
- **Type:** Enum
- **Value:** asc
- **Description:** Response sort order:

**Example:**
- \( \text{sort_dir}=\text{asc} \) asc = ascending
- \( \text{sort_dir}=\text{desc} \) desc = descending

### sort_key
- **Type:** String
- **Value:** name
- **Description:** Field name to use for sorting.

### sort_mode
- **Type:** Enum
- **Value:** auto
- **Description:** Collated ordering:

**Example:**
- \( \text{sort_mode}=\text{auto} \) auto = If all field values are numeric, collate numerically. Otherwise, collate alphabetically.
- \( \text{sort_mode}=\text{alpha} \) alpha = Collate alphabetically, not case-sensitive.
- \( \text{sort_mode}=\text{alpha_case} \) alpha_case = Collate alphabetically, case-sensitive.
### Response Type:
- **true** = Summarized response, omitting some index details, providing a faster response.
- **false** = Full response.

### Returned Values
The response to GET and other requests typically includes key-value pairs representing details about the resource that you are accessing. Returned values specific to the resource and/or operation are listed along with their descriptions.

### HTTP Status Codes
Responses can include HTTP status codes. Standard HTTP status codes are not included in endpoint documentation, but status codes with specific meaning for an endpoint and/or operation are noted.

### Error Messages
Requests with an error, such as a missing required parameter, can prompt an error response like the following example.

```xml
<response>
  <messages>
    <msg type="ERROR">
      In handler 'datamodelgenerate': The following required arguments are missing: sid.
    </msg>
  </messages>
</response>
```

### EAI Response Data
EAI response data, the `<eai:acl>` and `<eai:attributes>` elements, typically apply to all endpoints and are configuration-dependent, so redundant explanation is omitted. These elements are also elided from the response examples to make the documentation easier to read.

#### Access Control List (ACL)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>app</strong></td>
<td>The name of an app, one of: system.</td>
</tr>
<tr>
<td><strong>can_list</strong></td>
<td>For internal use only for the Splunk Web manager UI.</td>
</tr>
</tbody>
</table>

**can_share_*** indicates whether or not the current user can change the sharing state. The sharing state can be one of:
- **can_share_app** = App-level sharing
- **can_share_global** = Global sharing
- **can_share_user** = User-level sharing

**can_write** indicates whether or not the current user can edit this item.

**owner**
A value of nobody indicates that all users have access to the resource, but that write access to the resource might be restricted.

**modifiable**
Set to false for items not controlled by ACLs, such as items under `/server/logger`.

**perms.read** Properties that indicate read permissions of the resource.

**perms.write** Properties that indicate write permissions of the resource.

**removable** Indicates if an admin or user with sufficient permissions can remove the entity.

**sharing**
- **app** = Shared through an app.
- **global** = Shared to all apps.
- **user** = Private to a user.

**Note:** You can append `/acl` to an endpoint to access its ACL properties. For more information, see Access Control List in the REST API User Manual.

#### EAI attributes [<eai:attributes>]

The `<eai:attributes>` element shows the mandatory and optional fields.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>optionalFields</strong></td>
<td>Field is optional.</td>
</tr>
<tr>
<td><strong>requiredFields</strong></td>
<td>Field is required.</td>
</tr>
<tr>
<td><strong>wildcardFields</strong></td>
<td>Field can use wildcard.</td>
</tr>
<tr>
<td>Attribute</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>

**References**

See the following resources for more information on working with the Splunk REST API.

- *REST API User Manual*
- *REST API Tutorials*