Resource: feeds

List of available feeds

http://{device}/api/mgmt.newsfeeds/1.0/feeds

JSON

```
{
  "items": [ feed ],
  <prop>: any
}
```

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>feeds</td>
<td>&lt;object&gt;</td>
<td>List of available feeds</td>
<td></td>
</tr>
<tr>
<td>feeds.items</td>
<td>&lt;array of &lt;feed&gt;&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>feeds.items[items]</td>
<td>&lt;feed&gt;</td>
<td>Provides information about the feed. Also gives the categories and ID's for what news can be posted.</td>
<td></td>
</tr>
<tr>
<td>feeds.&lt;prop&gt;</td>
<td>&lt;any&gt;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Links

feeds: get

GET http://{device}/api/mgmt.newsfeeds/1.0/feeds

Response Body

Returns a **feeds** data object.

Resource: feed

Provides information about the feed. Also gives the categories and ID's for what news can be posted.

http://{device}/api/mgmt.newsfeeds/1.0/feeds/items/{name}

JSON

```
{
  "name": string,
  "categories": [
    { 
      "name": string,
      "items": [ summary ],
      <prop>: any
    }
  ]
}
```

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>feed</td>
<td>&lt;object&gt;</td>
<td>Provides information about the feed. Also gives the categories and ID's for what news can be posted.</td>
<td></td>
</tr>
<tr>
<td>feed.name</td>
<td>&lt;string&gt;</td>
<td>Name of the feed</td>
<td>Optional;</td>
</tr>
<tr>
<td>feed.categories</td>
<td>&lt;array of &lt;object&gt;&gt;</td>
<td></td>
<td>Optional;</td>
</tr>
<tr>
<td>feed.categories[items]</td>
<td>&lt;object&gt;</td>
<td></td>
<td>Optional;</td>
</tr>
<tr>
<td>feed.categories[items].name</td>
<td>&lt;string&gt;</td>
<td></td>
<td>Optional;</td>
</tr>
<tr>
<td>feed.categories[items].items</td>
<td>&lt;array of &lt;summary&gt;&gt;</td>
<td></td>
<td>Optional;</td>
</tr>
<tr>
<td>feed.categories[items].items[items]</td>
<td>&lt;summary&gt;</td>
<td>Describes the feed</td>
<td></td>
</tr>
<tr>
<td>feed.categories[items].&lt;prop&gt;</td>
<td>&lt;any&gt;</td>
<td></td>
<td>Optional;</td>
</tr>
</tbody>
</table>

Links
**feed: get**

```
GET http://{device}/api/mgmt.newsfeeds/1.0/feeds/items/{name}
```

Response Body

Returns a **feed** data object.

---

**Resource: summary**

Describes the feed

```
http://{device}/api/mgmt.newsfeeds/1.0/feeds/items/{name}/items/{id}
```

**JSON**

```json
{
  "id": string,
  "summary": string,
  <prop>: any
}
```

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>summary</td>
<td>&lt;object&gt;</td>
<td>Describes the feed</td>
<td></td>
</tr>
<tr>
<td>summary.id</td>
<td>&lt;string&gt;</td>
<td>Unique summary ID</td>
<td>Optional;</td>
</tr>
<tr>
<td>summary.summary</td>
<td>&lt;string&gt;</td>
<td>Brief user visible summary of the description</td>
<td>Optional;</td>
</tr>
<tr>
<td>summary.&lt;prop&gt;</td>
<td>&lt;any&gt;</td>
<td></td>
<td>Optional;</td>
</tr>
</tbody>
</table>

**Links**

**summary: get**

```
GET http://{device}/api/mgmt.newsfeeds/1.0/feeds/items/{name}/items/{id}
```

Response Body

Returns a **summary** data object.

---

**Resource: news**

A collection of all news on the system. It supports various query parameters for getting news. New news can be posted here.

```
http://{device}/api/mgmt.newsfeeds/1.0/news{?remote_user,severity,audit_id,feed_category,feed_name,source,feed_id,limit,user,offset,start_time,end_time}
```

**JSON**

```json
{
  "items": [ news_item ],
  <prop>: any
}
```

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>news</td>
<td>&lt;object&gt;</td>
<td>A collection of all news on the system. It supports various query parameters for getting news. New news can be posted here.</td>
<td></td>
</tr>
<tr>
<td>news.items</td>
<td>&lt;array of news_item&gt;</td>
<td></td>
<td>Optional;</td>
</tr>
<tr>
<td>news.items[].news_item</td>
<td>&lt;news_item&gt;</td>
<td>The schema for news items</td>
<td></td>
</tr>
<tr>
<td>news.&lt;prop&gt;</td>
<td>&lt;any&gt;</td>
<td></td>
<td>Optional;</td>
</tr>
</tbody>
</table>

**Links**
news: get
GET http://{device}/api/mgmt.newsfeeds/1.0/news{?
remote_user,severity,audit_id,feed_category,feed_name,source,feed_id,limit,user,offset,start_time,end_time}

Response Body
Returns a news data object.

news: create
POST http://{device}/api/mgmt.newsfeeds/1.0/news{?
remote_user,severity,audit_id,feed_category,feed_name,source,feed_id,limit,user,offset,start_time,end_time}

Request Body
Provide a news_item data object.

Response Body
Returns a news_item data object.

news: set
PUT http://{device}/api/mgmt.newsfeeds/1.0/news{?
remote_user,severity,audit_id,feed_category,feed_name,source,feed_id,limit,user,offset,start_time,end_time}

Request Body
Provide a news data object.

Response Body
Returns a news data object.

Resource: news_item
The schema for news items

http://{device}/api/mgmt.newsfeeds/1.0/news/items/{id}

JSON

```json
{
  "id": integer,
  "timestamp": timestamp-hp,
  "feed_name": string,
  "feed_category": string,
  "feed_id": string,
  "user": string,
  "remote_user": string,
  "audit_id": string,
  "source": string,
  "severity": string,
  "details": string,
  "resources": news_resources
}
```

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>news_item</td>
<td>object</td>
<td>The schema for news items</td>
<td>Required properties: [feed_id, feed_name, feed_category, timestamp, severity, user, remote_user, audit_id, source, details];</td>
</tr>
<tr>
<td>news_item.id</td>
<td>integer</td>
<td>ID of the news item</td>
<td></td>
</tr>
<tr>
<td>news_item.timestamp</td>
<td>timestamp-hp</td>
<td>Miliseconds since epoch of news feed item</td>
<td>Read-only; Optional;</td>
</tr>
<tr>
<td>news_item.feed_name</td>
<td>string</td>
<td>Name of the feed</td>
<td></td>
</tr>
<tr>
<td>news_item.feed_category</td>
<td>string</td>
<td>Category of the feed</td>
<td></td>
</tr>
<tr>
<td>news_item.feed_id</td>
<td>string</td>
<td>Registered feed identification tag</td>
<td></td>
</tr>
<tr>
<td>news_item.user</td>
<td>string</td>
<td>Username for user driven feeds. Defaults to Blank for nonuser driven feeds.</td>
<td></td>
</tr>
</tbody>
</table>
news_item.remote_user <string> Remote username for user driven feeds. This can be the SCC user or the user authenticated through TACACs. Defaults to Blank for nonuser driven feeds.

news_item.audit_id <string> An audit identification ID to tag related resources together. Blank string if none.

news_item.source <string> Service sending out the news feed item.

news_item.severity <string> Severity of the news feed item. Values: low, normal, high, critical;

news_item.details <string> Detailed description of news feed item.

news_item.resources <news_resources> Additional detailed information about the news.

Links

news_item: get

GET http://{device}/api/mgmt.newsfeeds/1.0/news/items/{id}

Response Body
Returns a news_item data object.

Relations

news_item: feed

Related resource
feed

Variables

<table>
<thead>
<tr>
<th>Related var</th>
<th>Data value for replacement</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>0/feed_name</td>
</tr>
</tbody>
</table>

news_item: summary

Related resource
summary

Variables

<table>
<thead>
<tr>
<th>Related var</th>
<th>Data value for replacement</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>0/feed_name</td>
</tr>
<tr>
<td>id</td>
<td>0/feed_id</td>
</tr>
</tbody>
</table>

news_item: instances

Related resource
news

Type: news_resource

Contains JSON data supporting the news. It should be self describing, containing information to interpret the JSON data.

```
{
    "resource_id": string,
    "json_object": {
        "prop": any
    },
    "object_tag": string
}
```
### Property Name | Type | Description | Notes
--- | --- | --- | ---
news_resource | `<object>` | Contains JSON data supporting the news. It should be self-describing, containing information to interpret the JSON data. | Required properties: [resource_id, object_tag, json_object];

news_resource.resource_id | `<string>` | ID of the schema containing resource | 

news_resource.json_object | `<object>` | Object containing details of the news feed item: contains new object for configuration type, contains event object if based on lumberjack events framework. | 

news_resource.json_object.<prop> | `<any>` | Optional; | 

news_resource.object_tag | `<string>` | Tag for the JSON resource | 

---

**Type: news_resources**

Additional detailed information about the news

**JSON**

```json
{
  "items": [ news_resource ],
  <prop>: any
}
```

---

### Property Name | Type | Description | Notes
--- | --- | --- | ---
news_resources | `<object>` | Additional detailed information about the news | 

news_resources.items | `<array of <news_resource>>` | | Optional;

news_resources.items[items] | `<news_resource>` | Contains JSON data supporting the news. It should be self-describing, containing information to interpret the JSON data. | 

news_resources.<prop> | `<any>` | Optional; |