Oracle® Fusion Middleware

WebCenter Sites REST API Resource Reference

11*g* Release 1 (11.1.1.8.0)

E35835-02

October 2013

Oracle Fusion Middleware WebCenter Sites REST API Resource Reference, 11*g* Release 1 (11.1.1.8.0)

E35835-02

Contents

Preface

The *Oracle WebCenter Sites REST API Resource Reference* describes the REST API resources that are provided with Oracle WebCenter Sites. The REST API is used to access Oracle WebCenter Sites data.

Audience

This guide is written primarily for developers. It is assumed that developers have a clear knowledge of their company's business needs, and a basic understanding of their roles in the development of the online site and its back end. This guide is also useful to administrators, who collaborate with developers by setting up content management sites, site users, workflow processes, publishing methods, and Oracle WebCenter Sites client options.

Developers must know Java, JavaServer Pages (JSP), XML, and HTML. Administrators are not required to have programming experience, although a technical background is assumed.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle WebCenter Sites Release 11*g*R1 documentation set:

- *Oracle WebCenter Sites Certification Matrix*
- *Oracle Fusion Middleware WebCenter Sites Release Notes*
- *Oracle Fusion Middleware WebCenter Sites Installation Guide*
- *Oracle Fusion Middleware WebCenter Sites User's Guide*

- *Oracle Fusion Middleware WebCenter Sites Administrator's Guide*
- *Oracle WebCenter Sites Tag Reference*
- *Oracle WebCenter Sites Java API Reference*

These additional documents may also be useful:

- *Oracle Fusion Middleware WebCenter Sites Property Files Reference*
- *Oracle WebCenter Sites REST API Bean Reference*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

REST API

Oracle WebCenter Sites provides a REST API for accessing WebCenter Sites data. REST services are supported on both WebCenter Sites and Satellite Server to leverage the Satellite Server cache.

This chapter contains the following sections:

- Section 1.1, "Default Location of Resources"
- Section 1.2, "REST Resources"

1.1 Default Location of Resources

The default location of REST resources is under the WebCenter Sites `BaseURI`:

```
<protocol>://<hostname:port>/<servletPath>/REST/
```

For example, on a local installation the `BaseURI` could be the following:

```
http://localhost:<port>/cs/REST/
```

The `<BaseURI>/application.wadl` file lists the resources described in this reference.

1.2 REST Resources

This section lists supported REST resources and methods, including requirements and error codes that can be returned if a resource does not exist or unforeseen errors occur.

Types of Resource Views

Each resource has two types of views: list and detail. The list view lists resources of a given type (`users` for example) and for each resource, an `href` pointing to the detailed view. For example, the resource `<BaseURI>/users` returns a list view naming all users in the system, among them `fwadmin`.

```
<users>
<href>http://<hostname>:<port>/cs/REST/users/fwadmin</href>
<name>fwadmin</name>
</users>
```

The `fwadmin` entry contains an `<href>` that specifies the URL of the detail view where more information about `fwadmin` can be found.

REST Security

The following is a list of REST resource security requirements:

- Except for `/timezone` and `/userlocales`, all other REST resources are protected. For information about REST resource security, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide* and the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide*.
- By default, if an authorization failure occurs, then the login page for Central Authentication Service (CAS) is displayed. If you want to receive a `500` error instead, add `auth-redirect=false` to the URL when making the request.
- To authenticate all REST `POST/PUT/DELETE` requests as valid, each request requires a header with `X-CSRF-Token` as the key and a value of either a CAS ticket (multi or single) or a `sessionid`.

The rest of this section contains information about the following resources:

- Section 1.2.1, "Asset Type"
- Section 1.2.2, "Asset"
- Section 1.2.3, "Index"
- Section 1.2.4, "Site"
- Section 1.2.5, "Site Plan Navigation"
- Section 1.2.6, "Role"
- Section 1.2.7, "Application"
- Section 1.2.8, "User"
- Section 1.2.9, "User Locale"
- Section 1.2.10, "User Definition"
- Section 1.2.11, "ACLs"
- Section 1.2.12, "Groups"
- Section 1.2.13, "Geographical Location"
- Section 1.2.14, "Current Device"
- Section 1.2.15, "Recommendation"
- Section 1.2.16, "Segments"
- Section 1.2.17, "Visitor History"

1.2.1 Asset Type

All Asset Types	**/types**
Description	This resource reads the list of asset types in the system.
REST Methods	`GET`
Possible Error Codes	`500` if unforeseen error occurred

Selected Asset Type	**/types/{assettype}**
Description	This resource displays detailed information about the specified asset type.
REST Methods	`GET`, `PUT`, `DELETE`

Selected Asset Type	/types/{assettype}
Possible Error Codes	`404` if asset type does not exist `500` if unforeseen error occurred

All Subtypes	/types/{assettype}/subtypes
Description	This resource reads all subtypes of the specified asset type.
REST Methods	`GET`
Possible Error Codes	`404` if asset type does not exist `500` if unforeseen error occurred

Selected Subtype	/types/{assettype}/subtypes/{subtype}
Description	This resource reads the specified subtype of the specified asset type.
REST Methods	`GET`
Possible Error Codes	`404` if asset type or subtype does not exist `500` if unforeseen error occurred

Site-Enabled Asset Types	/sites/{sitename}/types
Description	This resource lists all asset types on the specified site.
REST Methods	`GET`
Possible Error Codes	`404` if site does not exist `500` if unforeseen error occurred

Site-Specific Asset Type Search	/sites/{sitename}/types/{assettype}/search
Description	This resource supports searches on the specified asset type on the specified site. **Note:** Access to any search is provided by granting `LIST` privileges on an `Asset` resource. In this case, search access can be configured per site or for all sites, but not per asset type. All users with access to the search service will be able to search assets of all asset types (in a particular site, if one is specified).
REST Methods	`GET`
Possible Error Codes	`404` if site does not exist

Global Asset Type Search	**/types/{assettype}/search**
Description	This resource supports searches on the specified asset type. For this resource to work, the asset type index must be configured in WebCenter Sites (otherwise, an exception is thrown and the 404 error code is displayed to the user). **Note:** Access to any search is provided by granting `LIST` privileges on an `Asset` resource. In this case, search access can be configured per site or for all sites, but not per asset type. All users with access to the search service will be able to search assets of all asset types (in a particular site, if one is specified).
Query Parameters	■ **`q`** Specify a query String, the search engine-specific query that was specified during the index configuration. (The default search engine is Lucene.) ■ **`field:[fieldname]:[Operation]`** Specify a query condition for a field of `[fieldname]`. `[Operation]` is optional and can be one of `equals`, `range`, `startswith`, `phrase`, `wildcard`, `similar`, or `contains`; the default is contains. For the range operation, the value's format is `{lower bound}:{upper bound}`, e.g., `1:10`. Only the first value specified will be used in the search. ■ **`startindex`** Specify the starting index from where the results will be returned. ■ **`count`** Specify the total number of items to be returned. ■ **`sortfield:[fieldname]:[asc/des]`** Specify the field on which the results would be sorted. **Note:** For Lucene, sorting is possible only fields that were indexed as "not tokenized". **Example:** `http://<hostname>:<port>/cs/REST/types/Content_C/search?field:name:contains=FSII&startindex=5&count=3`
REST Methods	`GET`
Possible Error Codes	`404` if the indexing configuration does not exist `500` if unforeseen errors occurred

1.2.2 Asset

Selected Asset, Selected Site	**/sites/{sitename}/types/{assettype}/assets/{id}**
Description	This resource provides developers with `Create`, `Edit`, `Update`, `Delete(CRUD)` functions for operating on the specified asset. `{id}` is in the long asset id format.
REST Methods	`HEAD`, `GET`, `POST`, `PUT`, `DELETE`

Selected Asset, Selected Site	**/sites/{sitename}/types/{assettype}/assets/{id}**
Possible Error Codes	404 if site does not exist 404 if asset type does not exist 404 if asset does not exist 500 if unforeseen error occurred

All Asset Associations, Selected Site	**/sites/{sitename}/types/{assettype}/assets/{id}/associations**
Description	This resource lists all asset associations on the specified site.
REST Methods	GET
Possible Error Codes	404 if site does not exist 404 if asset type does not exist 404 if asset does not exist 500 if unforeseen error occurred

Selected Asset Association and Site	**/sites/{sitename}/types/{assettype}/assets/{id}/associations/{assocName}**
Description	This resource lists the specified asset association on the specified site.
REST Methods	GET
Possible Error Codes	404 if site does not exist 404 if asset type does not exist 404 if asset does not exist 500 if unforeseen error occurred

Site-Specific Assets Search	**/sites/{sitename}/search**
Description	This resource supports searches on all assets on the specified site. **Note**: Access to any search is provided by granting LIST privileges on an Asset resource. In this case, search access can be configured per site or for all sites, but not per asset type. All users with access to the search service will be able to search assets of all asset types (in a particular site, if one is specified).
REST Methods	GET
Possible Error Codes	404 if site does not exist

Global Assets Search	**/search**
Description	This resource supports global searches on assets. **Note:** REST does not restrict the types of assets that are returned. As a result, assets of a certain type can be returned to a user who may otherwise not have permission to access the asset type. **Note**: Access to any search is provided by granting `LIST` privileges on an `Asset` resource. In this case, search access can be configured per site or for all sites, but not per asset type. All users with access to the search service will be able to search assets of all asset types (in a particular site, if one is specified).
Query Parameters	■ **`q`** Specify a query String, the search engine-specific query that was specified during the index configuration. (The default search engine is Lucene.) ■ **`field:[fieldname]:[Operation]`** Specify a query condition for field of `[fieldname]` with the specified operation. `[Operation]` is optional and can be one of `equals`, `range`, `startswith`, `phrase`, `wildcard`, similar, or `contains`; the default is `contains`. For the `range` operation, the value's format is `{lower bound}:{upper bound}`, e.g., `1:10` ■ **`startindex`** Specify the starting index from where the results will be returned. ■ **`count`** Specify the total number of items to be returned. ■ **`sortfield:[fieldname]:[asc/des]`** Specify the field on which the results would be sorted. **Note:** For Lucene, sorting is possible only for fields that were indexed as "not tokenized". **Example:** `http://<hostname>:<port>/cs/REST/types/Content_C/search?field:name:contains=FSII&startindex=5&count=3`
REST Methods	`GET`
Possible Error Codes	`500` if unforeseen error occurred

1.2.3 Index

Indexing Configurations	**/indexes**
Description	This resource lists configurations for global and asset type-based indexing. Global indexing produces a single index of asset types. Asset type indexing produces one index per asset type.
REST Methods	`GET`, `PUT`, `POST`, `DELETE`
Possible Error Codes	`404` if indexing configuration does not exist `500` if unforeseen error occurred

Indexing Configuration	/indexes/{source}
Description	This resource supports the configuration of global indexing and provides CRUD functions for operating on global and per asset type indexing configurations. Global indexing produces a single index of asset types. Asset type indexing produces one index per asset type. **Note:** In this resource's URL, source is either Global (for the global index) or the name of the asset type (for each asset type-based index). For asset type-based indexing, the indexing process will be also started, stopped, paused, or resumed automatically depending on your choice of HTTP method (PUT, POST, DELETE) and index status (ENABLED, PAUSED). Global indexing is enabled for all asset types by default, although the configuration may not exist.
REST Methods	GET, POST, PUT, DELETE
Possible Error Codes	404 if indexing configuration does not exist 500 if unforeseen error occurred

1.2.4 Site

All Sites	/sites
Description	This resource lists all sites in the system.
REST Methods	GET
Possible Error Codes	500 if unforeseen error occurred

Selected Site	/sites/{sitename}
Description	This resource provides developers with CRUD functions for operating on the specified site. To perform CRUD operations on a site, the user must have either the GeneralAdmin or SiteAdmin role on that site in addition to the privileges granted via groups. To manage AdminSite, a user must have the GeneralAdmin role.
REST Methods	HEAD, GET, POST, PUT, DELETE
Possible Error Codes	404 if site does not exist 500 if unforeseen error occurred

1.2.5 Site Plan Navigation

Publication Node	**/sites/{sitename}/navigation**
Description	This resource reads the Site Plan tree for the site specified by `{sitename}`. Accessing this resource requires having `READ` permission on the corresponding site resource. Query parameters defined for the service are as follows: ■ `depth`: Controls the depth to which the Site Plan tree can be read. Default value: `1` Accepted values: Integers or the string `all` (case-insensitive) to read up to the maximum depth. ■ `code`: Controls the type of site plan nodes (placed or unplaced) to be read. By default, both placed and unplaced pages are read back. The value for this parameter is case-insensitive. For more information about the Site Plan tree, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.
REST Methods	`GET`
Possible Error Codes	`400` Exception in parsing depth. `400` Depth is less than or equal to `0`. `403` User does not have permission. `404` Site does not exist. `500` No site plan publication node for the site. `500` Any other unforeseen error.

Page Node	**/sites/{sitename}/navigation/{pageid}**
Description	This resource reads the Site Plan sub-tree for the site specified by `{sitename}` and page specified by `{pageid}` as the root. Accessing this resource requires having `READ` permission to at least the page asset resource that is specified by `pageid`. Query parameters defined for the service are as follows: `depth`: Controls the depth to which the Site Plan sub-tree can be read. Default value: `1` Accepted values: Integers or the string `all` (case-insensitive) to read up to the maximum depth. For more information about the Site Plan tree, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.
REST Methods	`GET`

Page Node	/sites/{sitename}/navigation/{pageid}
Possible Error Codes	400 Page asset type is not enabled on site. 400 Exception in parsing depth. 400 Depth is less than or equal to 0. 403 User does not have permission. 404 Either the site and/or page asset does not exist. 500 Node path is null for page. 500 No node elements exist in node path for page. 500 Root node is not a publication node for page. 500 Inconsistent child node. 500 Any other unforeseen error.

1.2.6 Role

All Roles	/roles
Description	This resource reads the list of roles that are stored in the system.
REST Methods	GET
Possible Error Codes	500 if unforeseen error occurred

Selected Role	/roles/{rolename}
Description	This resource provides CRUD functions for operating on the specified role.
REST Methods	GET, POST, PUT, DELETE
Possible Error Codes	404 if role does not exist. 500 if unforeseen error occurred

1.2.7 Application

All Applications	/applications
Description	This resource lists registered applications.
REST Methods	GET
Possible Error Codes	500 if unforeseen error occurred

Selected Application	/applications/{applicationid}
Description	This resource registers the specified application by performing CRUD operations on the application's FW_View and FW_Application assets. More information about application registration is available in the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.
REST Methods	GET, PUT, POST, DELETE
Possible Error Codes	404 if application does not exist. 500 if unforeseen error occurred

1.2.8 User

All Users	**/users**
Description	This resource lists all users that are defined in the system.
REST Methods	`GET`
Possible Error Codes	`500` if unforeseen error occurred

Selected User	**/users/{username}**

Description	This resource provides `CRUD` functions for operating on the specified user. If the user profile contains an image, a 90 x 90 pixel thumbnail image is returned by the service. If the thumbnail image is required in a different size, the request can be made with the following parameters: `imageHeight` and `imageWidth` (in pixels); the thumbnail image is then returned at the specified size. The actual uploaded image is always returned in its original size when user information is requested. To restrict the user service to return only the thumbnail image and not the actual image, set the `thumbOnly` parameter to `true`.
	The "otherAttributes" field provides placeholders for custom user attributes, which you define when creating a WebCenter Sites user either in LDAP or in the WebCenter Sites database. (Note that for LDAP, each user will be a descendant of the value that is specified for the `peopleParent` property in `dir.ini`. The `peopleParent` property value is set during integration with LDAP.)
	Regardless of where custom user attributes are created, to take effect in the WebCenter Sites application they need to be specified in the WebCenter Sites `dir.ini` file, in the `requiredPeopleAttrs` property as a key-value pair in the following format:
	`requiredPeopleAttrs=`
	`  attribute1=description1&attribute2=description2`
	The key is the attribute name as it appears in the WebCenter Sites database (or in LDAP), and the value is the attribute's description as it appears in the WebCenter Sites interface. The property will be looked up by both the WebCenter Sites Admin interface and REST API in defining user attributes.

Selected User	**/users/{username}**

Description (*continued*)	For example, if you want to specify two attributes called "Phone number" and "Fax" for all users, do the following:
	1. Define two attributes called "Phone number" and "Fax". If you are using WebCenter Sites, define them as user attributes (they are stored in the `SystemUserAttr` table).
	2. In the WebCenter Sites `dir.ini` file, provide the attribute references in the `requiredPeopleAttrs` property as follows: `requiredPeopleAttrs=phone=Phone number&fax=Fax`
	3. The above configuration allows `Phone number` and `Fax` to be displayed as attributes in the user account form in the WebCenter Sites Admin interface and the WEM Admin interface.
	4. If you create a user with `Phone number:12345678` and `Fax:23456789`, invoking a REST request on the user URI `/users/{username}` provides the following information for the "otherAttributes" field:
	`<otherAttributes>`
	`<name>phone</name>`
	`<value>12345678</value>`
	`</otherAttributes>`
	`<otherAttributes>`
	`<name>fax</name>`
	`<value>23456789</value>`
	`</otherAttributes>`

REST Methods	`HEAD`, `GET`, `POST`, `PUT`, `DELETE`
Possible Error Codes	`404` if user does not exist `500` if unforeseen error occurred

All Site Users	**/sites/{sitename}/users**
Description	This resource lists the roles of all users on the specified site.
REST Methods	`GET`
Possible Error Codes	`404` if site does not exist `500` if unforeseen error occurred

Selected Site User	**/sites/{sitename}/users/{username}**
Description	This resource lists the roles of the specified user on the specified site and supports the REST methods listed below.
REST Methods	`GET`
Possible Error Codes	`404` if site does not exist `404` if user does not exist `500` if unforeseen error occurred

1.2.9 User Locale

The `/userlocales` resource is not protected

User Locales	/userlocales
Description	This resource lists all locales that are defined in the system.
REST Methods	`GET`
Possible Error Codes	`500` if unforeseen error occurred

1.2.10 User Definition

User Def	/userdef
Description	This resource reads attributes in the user profile. When the database is used for authentication, only the WebCenter Sites predefined user attributes are returned. If LDAP is used, the service returns the attributes that are defined in the `requiredPeopleAttrs` property of the `dir.ini` file, along with the predefined WebCenter Sites attributes.
REST Methods	`GET`
Possible Error Codes	`500` if unforeseen error occurred

1.2.11 ACLs

ACLs	/acls
Description	This resource lists all ACLs that are defined in WebCenter Sites.
REST Methods	`GET`
Possible Error Codes	`500` if unforeseen error occurred

1.2.12 Groups

All Groups	/groups
Description	This resource lists all REST security groups in the system.
REST Methods	`GET`
Possible Error Codes	`500` if unforeseen error occurred

Selected Group	/groups/{groupname}
Description	This resource provides the security privilege configuration of the specified group.
REST Methods	`GET`
Possible Error Codes	`404` if requested group is not found

1.2.13 Geographical Location

The `/timezone` resource is not protected.

Time Zone	/timezone
Description	This resource returns the server's time zone.

Time Zone	/timezone
REST Methods	GET
Possible Error Codes	500 if unforeseen error occurred

1.2.14 Current Device

Current Device Information	/currentdevice
Description	This service retrieves the device information from the User-Agent header present in an HTTP request, and returns the matching device and device group information in the form of a DeviceBean. If the User-Agent is present in the current device repository, and it has device capabilities that are mentioned in the repository, the returned DeviceBean will contain a DeviceCapabilites bean. **For example:**

```
Input to the service is User-Agent header.
Behavior of service for different cases -
Case 1) User-Agent is present in device repository and
          a) User-Agent matches to a device group -
            Result:   A DeviceBean with following info
               is returned -
                   DeviceBean
                            {
                             Name of device;
                             User Agent of device;
                             DeviceGroupBean; //A bean
                                representing a matching
                                device group, with
                                details such as name,
                                suffix, priority.
                             DeviceCapabilitiesBean; //A
                                bean representing all
                                device capabilities as
                                present in current device
                                repository.
                            }

      b) User-Agent does not match to any device group -
          Result: A DeviceBean with name of device as
            present in repository, its capabilities as
            DeviceCapabilitiesBean and matching device
            group as default one i.e. Desktop device
            group.

Case 2) User-Agent is NOT present in device repository(In
   which case DeviceCapabilitiesBean will not be
   returned, as capabilities are read from device
   repository) and
```

Current Device Information	/currentdevice

Description (*continued*)	`a) User-agent matches to a device group based on`
	`user-agent regular expression as a criteria`
	`defined in some Device Group -`
	`Result:   A DeviceBean with name as 'Unknown`
	`Device',its user-agent and a DeviceGroupBean`
	`with info about matched DeviceGroup. NO`
	`DeviceCapabilitiesBean.`
	
	`b) User-Agent does not match to any device group.`
	`The default device group DESKTOP will be`
	`returned as matching device group.`
	`Result: A DeviceBean with name as 'Unknown`
	`Device',it's user-agent and DESKTOP`
	`DeviceGroupBean."`

REST Methods	`GET`
Possible Error Codes	`500` if unforeseen error occurred.

1.2.15 Recommendation

Selected Recommendation	/sites/{sitename}/engage/recommendation/{recommendation}
Description	This resource provides the Recommendation asset in the context of the specified site. You can use the `fields` parameter to specify the attributes to be retrieved from and populated in the Recommendation asset. You can also use the `segments` parameter to specify a comma separated list of segments to be included in the result. **Parameters:** ■ `visitorid` (required): The ID or user name of the user. ■ `fields`: A comma separated list of attributes to be retrieved from the Recommendation assets. The attributes `name`, `id`, and `description` are selected by default. **Example:** `fields=name,id,createdby,updatedby,description` ■ `segments`: A comma separated list of segments to be included in the result. **Example URL:** `http://localhost:8080/cs/REST/sites/FirstSiteII/engage/recommendation/SportsRecommendation?fields=name,id,description,createdby,updatedby&visitorid=hellouser`
REST Methods	`GET`
Possible Error Codes	`404` if site or recommendation does not exist. `500` if unforeseen error occurs.

1.2.16 Segments

Selected Segments	**/sites/{sitename}/engage/segments**

Description	This service provides the segment details for a named set of segments or the segment details for a visitor.
	Parameters:
	■ `visitorid` (required): The visitor ID or user name of the user.
	■ `segments`: A comma separated list of segments.
	Example: `?segments=seg1,seg2,seg3`
	■ `fields`: A comma separated list of fields of the Segment asset to be included in the result. The fields `name`, `description`, and `id` are included by default.
	Example: `fields=name,id,createdby,updatedby,description`
	Example URLs:
	■ To retrieve the segment details for a list of segments:
	`http://localhost:8080/cs/REST/sites/FirstSiteII/engage/segments?segments=Seg1,Seg2&fields=name,updatedby&visitorid=username`
	■ To retrieve the segment details for a visitor:
	`http://localhost:8080/cs/REST/sites/FirstSiteII/engage/segments?visitorid=usern`

REST Methods	`GET`
Possible Error Codes	`404` if site does not exist. `500` if unforeseen error occurs.

1.2.17 Visitor History

Visitor History	**/visitor/visitorhistory**

Description	This resource provides interfaces for retrieving and recording the history of visitors. To record visitor history, the history attribute must be provided in the following format:
	`?visitorid=testuser&historyDef=LoginRecord&`
	`    param1=attributeName1&value1=valueOfAttributeName1`
	Parameters:
	■ `visitorid` (required): The ID or user name of the user.
	■ `historyDef` (required): The history definition being retrieved or recorded.
	■ `fields`: A comma separated list of fields of the Visitor History to be included in the result. The fields `name`, `description`, and `id` are included by default.
	Example: `fields=name,id,createdby,updatedby,description`

REST Methods	`GET`, `POST`
Possible Error Codes	`500` if unforeseen error occurs.