

Grouper Voot Connector

Wiki Home	Download Grouper	Grouper Guides	Community Contributions	Developer Resources	Deployment Guide
---------------------------	----------------------------------	--------------------------------	---	-------------------------------------	----------------------------------

The Grouper VOOT Connector is a jar plugin for the Grouper WS which implements the VOOT protocol specification. Note: in 2.3 and 2.4 it is in a patch in the WS instead of a separate jar. You can have both the patch and jar if it so happens that way.

VOOT is a protocol for exchanging group information externally to applications, defined within the GN3 JRA3 T2 ([Identity Federations Task](#)).

The plugin developed for Grouper implements the version 0.9 of the VOOT protocol, you can find detailed specification at the [Terena wiki page](#). More recent versions of the protocol are still under definition phase, for an update on advancements you can refer to the [VOOT specification github project](#).

Note: only the REST operations are implemented, not the OAuth 2.0. VOOT also described an authentication process based on OAuth. This implementation inside Grouper delegates authentication to anything already implemented for Grouper WS and is pluggable in the same way that Grouper WS is. You can easily implement servlet container or web server basic auth without any effort. If someone has an OAuth 2.0 VOOT consumer and wants to work with the Grouper team to make OAuth 2.0 work with this plugin, let the Grouper team know.

This is currently running on the [Grouper Demo Server](#).

Calls

The voot protocol defines two different sets of calls that can be made to a Grouper server: retrieve group membership and retrieve members of a group.

Retrieve Group Membership:

This call retrieves a list of groups the user is a member of:

```
/groups/@me
```

or

```
/groups/{{userId}}
```

where `userId` is replaced with an identifier of the user at the provider.
The response can include the following keys:

- (REQUIRED) `id`: The, to the provider, local unique identifier of the group;
- (REQUIRED) `voot_membership_role`: The role the user has in this group;
- (OPTIONAL) `title`: The short human readable name of the group;
- (OPTIONAL) `description`: A description of the group.

The `id` field SHOULD be opaque to the client. The field `voot_membership_role` can be any of these values: `admin`, `manager` or `member`.

Retrieve Members of a Group:

This call retrieves a list of all members of a group the user is a member of:

```
/people/@me/{groupId}
```

or

```
/people/{userId}/{groupId}
```

Where `userId` is replaced with an identifier of the user at the provider and `groupId` `http://www.internet2.edu/media/medialibrary/2013/09/04/internet2-mace-dir-eduperson-201203.html` is replaced with a group identifier which was obtained through for instance the call used to retrieve group membership for a particular user.

The response can include the following keys:

- (REQUIRED) `id`: The, to the provider, local unique identifier of the user;
- (OPTIONAL) `displayName`: The name by which the user prefers to be addressed;
- (OPTIONAL) `voot_membership_role`: The role the user has in this group;
- (OPTIONAL) `emails`: The email address(es) of the user;

The `id` field SHOULD be opaque to the client. The field `voot_membership_role` can be any of these values: `admin`, `manager` or `member`. The user MUST be a member, but not necessary have the role `member` of the group being queried. The `emails` field SHOULD contain a list of email addresses which provides the type to be any of `work`, `home` or `other`.

Mappings

- The voot Group ID is the grouper group system name
- The voot Group Name is the grouper group display name
- The voot Group description is the grouper group description
- The voot person email maps to the subject attribute name in the sources.xml file identified by `emailAttributeName`
- The voot person id is the subject id
- The voot person displayName is the subject name
- The voot member admin role maps to the group admin privilege (note, if they are a grouper admin, they are also a member of the group)
- The voot member manager role maps to the group update privilege (note, if they are a grouper update, they are also a member of the group)
- The voot member member role maps to the group members list

Request Parameters

The API calls have three OPTIONAL parameters that manipulate the result obtained from the provider:

- `sortBy`
- `startIndex`
- `count`

The `sortBy` parameter determines the key in the result that is used for sorting the groups or group members. It can be any of the keys of the objects returned by VOOT:

- `id`
- `displayname`
- `title`
- `description`
- `voot_membership_role`

If the key is not available in the set being sorted, the results are not sorted or sorted at the provider's discretion. It is up to the provider whether or not to sort and by what key in what order if these parameters are not present.

If the results are to be sorted, the value SHOULD be compared as strings and SHOULD be sorted case-insensitive in ascending order.

The `startIndex` parameter determines the offset as an integer ≥ 0 at which the start for giving back results.

The `count` parameter, an integer ≥ 0 indicates the number of results to be given back.

The `startIndex` and `count` parameters can be used to implement paging by returning only a subset of the results.

These parameters are OPTIONAL, if they are not provided or invalid the provider MUST consider `startIndex` equal to 0 and `count` equal to the total number of items available in the entire set for the particular query.

The sorting, if requested, MUST be performed on the provider before applying limiting the results based on the `startIndex` and `count` parameters.

For the API call, requesting user information, the `sortBy` parameter has no effect. Using `startIndex` and `count` is possible, however they are of little use as there always will be only one answer, assuming the user exists.

Grouper extensions

- If you pass in the request parameter `indentResponse=true`, then the resulting JSON will be indented to be more easily readable

Examples

Ask the Grouper team for a user/pass to the grouper demo WS (currently basic auth via apache)

Retrieve all the groups I am member of:

https://grouperdemo.internet2.edu/grouper-ws_v2_2/voot/groups/@me?indentResponse=true

```

{
  "entry": [
    {
      "description": "",
      "id": "atest:accentó:test",
      "name": "atest:accentó:test",
      "voot_membership_role": "admin"
    },
    {
      "description": "allowed to invite people to this application",
      "id": "etc:externalSubjectInviters",
      "name": "Grouper Administration:externalSubjectInviters",
      "voot_membership_role": "member"
    },
    {
      "description": "user interface users",
      "id": "etc:uiGroup",
      "name": "Grouper Administration:uiGroup",
      "voot_membership_role": "member"
    },
    {
      "description": "users allowed to log in to the UI",
      "id": "etc:webServiceClientUsers",
      "name": "Grouper Administration:webServiceClientUsers",
      "voot_membership_role": "member"
    },
    {
      "description": "",
      "id": "users:garr:Andrea:aGroup",
      "name": "users:garr:Andrea:aGroup",
      "voot_membership_role": "admin"
    },
    {
      "description": "",
      "id": "users:garr:Andrea:aGroup2",
      "name": "users:garr:Andrea:aGroup2",
      "voot_membership_role": "admin"
    },
    {
      "description": "",
      "id": "users:garr:Andrea:aGroup3",
      "name": "users:garr:Andrea:aGroup3",
      "voot_membership_role": "admin"
    },
    {
      "description": "",
      "id": "users:garr:Andrea:aGroup4",
      "name": "users:garr:Andrea:aGroup4",
      "voot_membership_role": "admin"
    }
  ]
  ,
  "itemsPerPage": 8,
  "startIndex": 0,
  "totalResults": 8
}

```

Retrieve groups I'm member of, starting from third and retrieve maximum 4 of them (pagination):

https://grouperdemo.internet2.edu/grouper-ws_v2_2/voot/groups/@me?indentResponse=true&startIndex=3&count=4

```
{
  "entry": [
    {
      "description": "users allowed to log in to the UI",
      "id": "etc:webServiceClientUsers",
      "name": "Grouper Administration:webServiceClientUsers",
      "voot_membership_role": "member"
    },
    {
      "description": "",
      "id": "users:garr:Andrea:aGroup",
      "name": "users:garr:Andrea:aGroup",
      "voot_membership_role": "admin"
    },
    {
      "description": "",
      "id": "users:garr:Andrea:aGroup2",
      "name": "users:garr:Andrea:aGroup2",
      "voot_membership_role": "admin"
    },
    {
      "description": "",
      "id": "users:garr:Andrea:aGroup3",
      "name": "users:garr:Andrea:aGroup3",
      "voot_membership_role": "admin"
    }
  ]
  ,
  "itemsPerPage": 4,
  "startIndex": 3,
  "totalResults": 8
}
```

Retrieve all groups I'm able to list:

https://grouperdemo.internet2.edu/grouper-ws_v2_2/voot/groups?indentResponse=true

```

{
  "entry": [
    {
      "description": "",
      "id": "users:nebraska:lfrerichs1:Bunnies:NuclearBunnys",
      "name": "users:nebraska:lfrerichs1:Bunnies:NuclearBunnys"
    },
    {
      "description": "Group containing list of addincludethingbwh after adding the includes and subtracting the
excludes",
      "id": "users:penn:bwh:manilla:addincludethingbwh",
      "name": "users:penn:bwh:manilla:addincludethingbwh"
    },
    {
      "description": "sys admin users",
      "id": "etc:sysadmingroup",
      "name": "Grouper Administration:sysadmingroup"
    },
    {
      "description": "",
      "id": "users:penn:mageerc:mageercAdmins",
      "name": "users:penn:mageerc:mageercAdmins"
    },
    #####ETC#####
  ]
  ,
  "itemsPerPage": 294,
  "startIndex": 0,
  "totalResults": 294
}

```

Search among all the groups I'm able to see, those containing "garr" in their name or id:

https://grouperdemo.internet2.edu/grouper-ws_v2_2/voot/groups?indentResponse=true&search=garr

```

{
  "entry": [
    {
      "description": "",
      "id": "users:garr:Andrea:aGroup",
      "name": "users:garr:Andrea:aGroup"
    },
    {
      "description": "",
      "id": "users:garr:Andrea:aGroup3",
      "name": "users:garr:Andrea:aGroup3"
    },
    {
      "description": "",
      "id": "users:garr:Andrea:aGroup2",
      "name": "users:garr:Andrea:aGroup2"
    },
    {
      "description": "",
      "id": "users:garr:Andrea:aGroup4",
      "name": "users:garr:Andrea:aGroup4"
    }
  ]
},
"itemsPerPage": 4,
"startIndex": 0,
"totalResults": 4
}

```

Show all users member of "users:garr:Andrea:aGroup2" group:

https://grouperdemo.internet2.edu/grouper-ws_v2_2/voot/people/@me/users:garr:Andrea:aGroup2?indentResponse=true

```

{
  "entry": [
    {
      "displayName": "Andrea",
      "id": "45a4fb096ba541c18620700e337508cf",
      "voot_membership_role": "admin"
    },
    {
      "displayName": "Chris Hyzer",
      "id": "0b5949edd3bf4b65a0ab7e9ce97a4cf9",
      "voot_membership_role": "member"
    }
  ]
},
"itemsPerPage": 2,
"startIndex": 0,
"totalResults": 2
}

```

Show users member of "users:garr:Andrea:aGroup2", order them by displayName, start from the fifth and retrieve maximum 2 of them (pagination):

https://grouperdemo.internet2.edu/grouper-ws_v2_2/voot/people/@me/users:garr:Andrea:aGroup4?sortBy=displayName&startIndex=5&count=2&indentResponse=true

```
{
  "entry": [
    {
      "displayName": "Fiona Brooks",
      "id": "fibr",
      "voot_membership_role": "member"
    },
    {
      "displayName": "Fiona Bush",
      "id": "fibu",
      "voot_membership_role": "member"
    }
  ]
},
{
  "itemsPerPage": 2,
  "startIndex": 5,
  "totalResults": 20
}
```

Installation

Here is the [README.txt](#)

Install the patch for 2.3 and 2.4.

To run Grouper Voot

- Setup and run the grouper WS
- Make sure your sources.xml has an email attribute name in applicable subject sources:

```
<!-- If using emails and need email addresses in sources, set which attribute has the email address in
this source -->
<init-param>
  <param-name>emailAttributeName</param-name>
  <param-value>email</param-value>
</init-param>
```

- Setup the web.xml

```
<filter-mapping>
  <filter-name>Grouper service filter</filter-name>
  <url-pattern>/voot/*</url-pattern>
</filter-mapping>

<servlet>
  <servlet-name>VootServlet</servlet-name>
  <display-name>Voot Servlet</display-name>
  <servlet-class>edu.internet2.middleware.grouperVoot.VootServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
  <servlet-name>VootServlet</servlet-name>
  <url-pattern>/voot/*</url-pattern>
</servlet-mapping>
```


- If you are using basic auth in the web.xml, make sure the voot servlet is protected:

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Voot services</web-resource-name>
    <url-pattern>/voot/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>grouper_user</role-name>
  </auth-constraint>
</security-constraint>
```

- Voot documentation:

<https://github.com/andreassolberg/voot/wiki/Protocol>

- Turn on the grouper web services, and try the following URL's:

```
https://grouper.whatever.com/grouperWs/voot/groups
https://grouper.whatever.com/grouperWs/voot/groups/@me
https://grouper.whatever.com/grouperWs/voot/groups/aSubjectId [note: put in valid subject ID]
https://grouper.whatever.com/grouperWs/voot/people/@me
https://grouper.whatever.com/grouperWs/voot/people/@me/aStem:aGroup2 [note: put in valid group name]
https://grouper.whatever.com/grouperWs/voot/people/aSubjectId/aStem:aGroup2 [note: put in valid subject ID and
group name]
```

You can pass in a param to indent the response:

```
https://grouper.whatever.com/grouperWs/voot/groups?indentResponse=true
https://grouper.whatever.com/grouperWs/voot/groups/@me?indentResponse=true
https://grouper.whatever.com/grouperWs/voot/groups/aSubjectId?indentResponse=true
https://grouper.whatever.com/grouperWs/voot/people/@me?indentResponse=true
https://grouper.whatever.com/grouperWs/voot/people/@me/aStem:aGroup2?indentResponse=true
https://grouper.whatever.com/grouperWs/voot/people/aSubjectId/aStem:aGroup2?indentResponse=true
```

You can also pass additional params to the different calls, being:

- sortBy: to get the results filtered by one field in the output JSON
- startIndex: start index (in case of paginated call it indicates the first result to be returned)
- count: number of results to show (in case of paginated call it indicates the page size)

This is running on the Grouper demo server e.g. here:

```
https://grouperdemo.internet2.edu/grouper-ws_v2_0_0/voot/groups/@me?indentResponse=true
https://grouperdemo.internet2.edu/grouper-ws_v2_0_0/voot/groups?indentResponse=true
https://grouperdemo.internet2.edu/grouper-ws_v2_0_0/voot/people/@me?indentResponse=true
https://grouperdemo.internet2.edu/grouper-ws_v2_0_0/voot/people/@me/aStem:aGroup2?indentResponse=true
```