

Groups Webservice Create Group

Create a group

Resource

- **PUT** **(root)/group/{group_identifier}**

The **group_identifier** can be a group name (cn) or a regid. The request must not contain an If-Match header.

Upload data

1. Upload content must contain one class "group".
2. Formatting html is ignored.
3. The content can be similar to that of a GET request.
4. Only one name (cn) can be specified.
5. You do not need to include a regid. If that class is omitted one will be generated.
6. You do not need to include a gid. If that class is supplied it will be ignored.
7. You must specify at least one Administrator.
8. An ACL identifier of "**dc=all**" (type="**none**") means everyone.
9. An ACL identifier of "**dc=none**" (type="**none**") means no one.
10. ACL elements may be omitted.
11. Version 2
 - a. You may specify an authfactor, but the value will be ignored.
 - b. You may specify "publishemail" but the value will be ignored.
 - c. You may omit the classification.

Conventions

{text} : replace {text} with specific content

(root): Webservice URL base

(type) : uwnetid | group | dns | eppn

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
  <meta http-equiv="Content-Type" content="application/xhtml+xml; charset=utf-8"/>
</head>
<body>
  <div class="group">
    Regid: <span class="regid">Group's regid</span>
    <br/>
    Desc: <span class="description">Group's title</span>
    <ul class="names">
      <li class="name">Group_name</li>
    </ul>

    <!-- version v2 and greater -->
    Authn factor: <span class="authnfactor">1</span>
    <br/>
    Classification: <span class="classification">u|p|r|c</span>
    <br/>
    Membership dependency group: <span class="dependson">dependency group cn</span>
    <br/>
    <!-- end v2 -->

    <!-- The enabled flag is "uwexchange" or "disabled" -->
    Email Enabled: <span class="emailenabled">uwexchange</span>

    <!-- publishemail is set to group_name@uw.edu when the group is created. Updates to the value will be
    ignored -->
    Published email address: <span class="publishemail">email_address</span>

    Allowed senders:
    <ul class="authorigs">
      <li class="authorig">sender id</li>
      <!-- More (li) if there are multiple allowed senders -->
    </ul>
    <!-- reporttoorig is a deprecated attribute. New groups will not return the attribute. Updates to the
    value will be ignored -->
```

Report to originator: `0 or 1`

Group contact: `contact id`

Admins:

```
<ul class="admins">
<li class="admin" type="(type)">identifier</li>
<!-- More (li) if the group has more admins -->
</ul>
<br/>
```

Allowed updaters:

```
<ul class="updaters">
<li class="updater" type="(type)">identifier</li>
<!-- More (li) if the group has multiple updaters -->
</ul>
<br/>
```

Allowed creators:

```
<ul class="creators">
<li class="creator" type="(type)">identifier</li>
<!-- More (li) if the group has multiple creators -->
</ul>
<br/>
```

Allowed readers

```
<ul class="readers">
<li class="reader" type="(type)">identifier</li>
<!-- More (li) if the group has multiple readers -->
</ul>
<br/>
```

Allowed viewers:

```
<ul class="viewers">
<li class="viewer" type="(type)">identifier</li>
<!-- More (li) if the group has multiple viewers -->
</ul>
<br/>
```

`<!-- version v2 and greater -->`

Allowed optin:

```
<ul class="optins">
<li class="optin" type="(type)">identifier</li>
<!-- More (li) if the group has multiple optins -->
</ul>
<br/>
```

Allowed optout:

```
<ul class="optouts">
<li class="optout" type="(type)">identifier</li>
<!-- More (li) if the group has multiple optouts -->
</ul>
<br/>
```

`<!-- end v2 -->`

`members`

`</div>`

`</body>`

`</html>`

Return

201	Content is the same as for a GET.
400	Invalid xhtml in the upload data.
	Invalid regid

401	No authorization.
	Group names did not match the name in the URL
412	If-Match header in the request.

Notes

1. The creator of a group is always an administrator of the new group. This administrator can be removed with a subsequent PUT.
2. The response always includes an ETag header, which can be used on subsequent GETs and must be used on subsequent updates or deletes.
3. The "emailenabled" attribute is ignored in a PUT request. Use the [affiliate resource](#) to update this.