



GroupWise Admin ReST API

Dan Christensen
Software Engineer
February 8th 2017



Overview

- Brief introduction to REST
- Admin Service URL layout
- Browsing the Admin Service
- Anatomy of a CURL command
- Create, Modify, Delete GroupWise objects
- Querying and exporting data
- Looping constructs

REST

- **RE**presentational **S**tate **T**ranfer
- Objects represented by documents that “live” at a specific URL endpoint
 - Documents are encoded as JSON or XML
- Two* types of endpoints
 - Collection (i.e. /users)
 - Object (i.e. /users/joe)
- HTTP verbs are used transfer documents to/from endpoints to affect the state of the object.

HTTP Verbs – Object endpoints

- GET /users/joe
 - Retrieve joe's current state
- PUT /users/joe
 - Modify joe's state (i.e. disable account)
- DELETE /users/joe
 - Delete joe from the system
- POST /users/joe ???
 - Not really applicable to object endpoints

HTTP Verbs – Collection endpoints

- GET /users
 - Get the list of all users
- POST /users
 - Create a new user in the users collection
- PUT /users (limited support)
 - Modify multiple objects in the user's collection
- DELETE /users (limited support)
 - Delete all users

Admin Service REST Endpoints

/domains

Hierarchical collection of domains, post offices, users, resources, groups, nicknames, libraries, agents, etc.

/system

System settings and records such as LDAP servers and directories, internet domains.

/list

Provides a view of all objects (regardless of containment) by type. Useful for querying and reporting.

Domain API - /domains

Hierarchical URL pattern alternating between **collection** and **object** endpoints

/domains – All domains on the system

/dom1 – Info about dom1

/postoffices – All post offices on dom1

/po1 – Info about po1 on dom1

/users – All users on dom1.po1

/user1 – Info about dom1.po1.user1

System API - /system

/system – System information

/directories – All directories

/edir1 – Info about the edir1 LDAP directory

List API - /list

- View all objects in system by type:
- /list/user – View all users in the system
- /list/poa – View all poa's in the system
- /list/directory – View all directories in the system
- ...

Browsing the Admin Service REST API

Anatomy of a CURL command

```
curl -k --user admin:password -X PUT {BASEURL}/{ENDPOINT} -H  
"Content-Type: application/json" --data {DATA}
```

-k : skip SSL certificate validation

--user admin:password : Admin credentials

-X PUT

HTTP verb (GET, PUT, POST, DELETE). Defaults to GET if omitted.

{BASEURL}/{ENDPOINT}

<https://dom.mycompany.com:9710/gwadmin-service>

</domains/dom1/postoffice/po1/users/user1>

-H "Content-Type: application/json"

Input document will be JSON encoded

{DATA}: JSON object

{**"givenName"**: "Dan"}

Create a New User

```
curl -k --user admin:password  
  -H "Content-Type: application/json"  
  -X POST  
  https://gw.x.com:9710/gwadmin-  
service/domains/dom1/postoffices/po1/users  
  --data "{\"name\": \"jdoe\", \"givenName\": \"John\",  
  \"surname\": \"Doe\"}"
```

Modify a User

```
curl -k --user admin:password  
  -H "Content-Type: application/json"  
  -X PUT  
  https://gw.x.com:9710/gwadmin-  
service/domains/dom1/postoffices/pol/users/jdoe  
  --data "{\"department\": \"Finance\"}"
```

Upload a User Photo

```
curl -k --user admin:password  
  -H "Content-Type: application/octet-stream"  
  -X PUT  
  https://gw.x.com:9710/gwadmin-  
service/domains/dom1/postoffices/pol/users/jdoe/picture  
  --data-binary @/path/to/photo.jpg
```

Change a password

```
curl -k --user admin:password  
  -H "Content-Type: application/json"  
  -X PUT  
  https://gw.x.com:9710/gwadmin-  
service/domains/dom1/postoffices/pol/users/jdoe/clientoptions  
  --data "{\"userPassword\": {\"value\": \"secret\"}}"
```

Disable a User Account

```
curl -k --user admin:password  
  -H "Content-Type: application/json"  
  -X PUT  
  https://gw.x.com:9710/gwadmin-  
service/domains/dom1/postoffices/pol/users/jdoe  
  --data "{\"loginDisabled\": true}"
```


Delete a User Account

```
curl -k --user admin:password
```

```
-X DELETE
```

```
https://gw.x.com:9710/gwadmin-  
service/domains/dom1/postoffices/po1/users/jdoe
```

Create a New Group

```
curl -k --user admin:password  
  -H "Content-Type: application/json"  
  -X POST  
  https://gw.x.com:9710/gwadmin-  
service/domains/dom1/postoffices/po1/groups  
  --data "{\"name\": \"ttp\"}"
```

Add a group member

```
curl -k --user admin:password  
  -H "Content-Type: application/json"  
  -X POST  
  https://gw.x.com:9710/gwadmin-  
service/domains/dom1/postoffices/po1/groups/ttp/members  
  --data "{\"id\": \"USER.dom1.po1.jdoe\", \"participation\":  
  \"PRIMARY\"}"
```

Delete a group member

```
curl -k --user admin:password
```

```
-X DELETE
```

```
https://gw.x.com:9710/gwadmin-  
service/domains/dom1/postoffices/po1/groups/ttp/members/USER.  
dom1.po1.jdoe
```

Querying and Exporting Data

- Use the `/list/{type}.csv` endpoint to export data in comma delimited format.
- `?attrs=domain,postoffice,name`
 - Specify a comma separated list of attributes to include in CSV. All attributes included by default.
- `&department=finance`
 - Specify any attribute as a query parameter to do a “starts with” search with that parameter.
- `&filter=visibility ne 'system'`
 - Use filter for more complex queries.
 - Operators: `eq`, `ne`, `lt`, `lte`, `gt`, `gte`, `match`

User query examples

- Restricted visibility
 - `/list/user?filter=visibility ne 'system'`
- Has Preferred Email Address
 - `/list/user?filter=preferredemailid ne null`
- Hasn't logged in over a week (run audit first)
 - `/list/user?filter=lastClientLoginTime lt now(-1,week)`
- Mailbox larger than 1 GB
 - `/list/user?filter=mailboxsizemb gt 1000`

Looping Constructs: CSV, AWK, Curl

```
urls=`  
  
curl -k -user admin:password "https://gw.x.com/gwadmin-  
service/list/user.csv?attrs=domain,postoffice,name&filter=lastClientLoginTime%2  
0lt%20now(-1,month)" |  
  
gawk --field-separator=, 'NR != 1 {print "https://gw.x.com:9710/gwadmin-  
service/domains/"$1"/postoffices/"$2"/users/"$3}'  
  
`  
  
for url in $urls; do curl -k --user admin:novell -X PUT $url -H "Content-Type:  
application/json" --data "{\"loginDisabled\": true}"
```



www.microfocus.com