MICRO FOCUS

Filr REST APIs

Getting Started and Tutorial

October 2016

Legal Notice

Contents

About This Guide

Audience

This guide is intended for Filr developers.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the **comment on this topic** link at the bottom of each page of the online documentation.

Documentation Updates

For the most recent version of the this guide, visit the REST API Documentation web site (http://www.novell.com/documentation/rest-api).

Additional Documentation

For documentation on the Filr product, see the File Documentation web site (http://www.novell.com/documentation/).

1 Introduction

1.1 What Is REST?

REST stands for Representation State Transfer. The client initiates HTTP requests to the server, which processes the requests and returns appropriate responses.

REST is a stateless protocol. In other words, the server does not store or track any client context (session) between request. Clients do not log in or log out. Instead, they supply basic authentication credentials with each request.

1.2 Resources and Responses

Requests and responses are built around the transfer of representations of resources. A resource is any meaningful entity or concept that may be acted upon. For example, in Filr, folders, files and users are resources.

There are several different logical categories of resources:

- Object
  - For example, a folder
- Collection
  - For example, the set of files in a folder
  - Each file by itself is an object resource, but the set of files in a folder is a collection resource that can also be acted upon.
- Attribute
  - For example, a folder’s parent

REST clients act on these resources using standard HTTP verbs (methods):

- GET: retrieve the resource from the server
- POST: create a new resource on the server/modify an existing resource
- PUT: modify a resource
- DELETE: delete a resource

Depending on the type of resource, each HTTP method has a different meaning:

- Folder (object resource)
  - GET: retrieves a representation of the folderl
  - PUT (or perhaps POST): modifies the folder
  - DELETE: deletes the folder
- Set of files in a particular folder (collection resource)
  - GET: retrieves the list of files
  - POST: add an additional file to the collection (create a new file in the folder)
  - PUT: replace the entire collection with a new list of files
  - DELETE: deletes all files in the folder
- Folder's parent (attribute resource)
  - GET: Retrieves the parent of the folder
  - PUT (or perhaps POST): Updates the parent folder attribute (moves the folder to a different location)

    DELETE: Removed the parent folder attribute (in this example, it would not make sense to support the DELETE method, because the parent folder attribute is mandatory)

1.3 Accessing the REST Interface

You can access the Filr REST interface on each Filr appliance using the following URL:

`https://filr_address:port/rest`

where filr_address is the IP address or DNS name of the Filr appliance and port is ???

1.4 Authenticating for Access

Each request to the Filr REST interface must be authenticated using basic authentication (see http://tools.ietf.org/html/rfc1945#section-11.1). If you enter your Filr REST URL in your browser address bar, it will prompt you to enter your Filr username and password.

Likewise, if you try to make an unauthenticated HTTP request using the curl command-line tool, the server responds with a 401 Unauthorized HTTP error:

[Request]

```
> curl -k -v https://amethyst.provo.novell.com:8443/rest
> GET /rest HTTP/1.1
> User-Agent: curl/7.35.0
> Host: amethyst.provo.novell.com:8443
> Accept: */*
```

[Response]

```
< HTTP/1.1 401 Unauthorized
< WWW-Authenticate: Basic realm="Novell Filr"
```

[Request]

```
> curl -k -v -u dlewis:novell https://amethyst.provo.novell.com:8443/rest
> GET /rest HTTP/1.1
> Authorization: Basic ZGxld2lzOm5vdmVsbA==
> User-Agent: curl/7.35.0
> Host: amethyst.provo.novell.com:8443
> Accept: */*
```

[Response]

```
< HTTP/1.1 200 OK
{...}
```

1.5 Navigation

The Filr REST Interface is fully navigable. In other words, resources contain links to other related resources, so clients can navigate from one resource to another without needing to understand the URL scheme of the REST interface.

For example, a folder resource contains links to the parent folder attribute, the lists of files and folders, etc.

The starting point for the REST Interface is `https://[server_address]:[port]/rest`.

This response consists of a list of related links. Each link has a “rel” (relation) attribute and an “href” attribute. All href values are relative to the base REST URL. After I request the root resource (/rest), I can retrieve the “self” resource as follows (append the href, /self, to https://amethyst.wal.novell.com:8443/rest):

The “self” resource is a User object representing the user making the REST request. It contains a number of additional related links (not shown in the example above), which allow the client to continue navigating the REST interface.

1.6 Object Model

In order to use the REST Interface, it's important to understand some things about the underlying object model and architecture that are not necessarily apparent when using the Filr web, desktop or mobile applications. Filr is built on the same architecture as Novell Vibe. While not so visible in the end-user Filr applications, some of the Vibe-isms appear in the REST interface.

1.6.1 Workspaces and Folders (Binders)

Filr consists of a hierarchy of workspaces and folders. The top level of this hierarchy is a workspace called “Home Workspace”. It has two child workspaces that are relevant to Filr:

- **Personal Workspaces:** Each Filr user has a workspace in this location containing the user's personal storage and link to the user's home directory.
- **Net Folders:** This workspace contains the Net Folder references that the has configured.

The top level folders that users see when logging into Filr (such as My Files, Shared with Me and Net Folders) do not exist in this workspace hierarchy. They are virtual views where files and folders are aggregated from different parts of the Filr workspace hierarchy.

Together, workspaces and folders are known as binders. That is, a binder is a generic container that can have children. A workspace is a binder that can contain other workspaces and also folders. A folder is a binder that can contain other folders and also folder entries (see below).

1.6.2 Folder Entries and Attachments

Files are represented in Filr as a folder entry with a single attachment. The attachment contains important metadata about the file itself, such as the size, last modification time and file name. The folder entry contains the Filr-related information, such as the location in the folder/workspace hierarchy and information about sharing and comments.

The REST interface tries to blend the folder entry and attachment into a single object called FileProperties where possible, but when developing a REST client, it's still important to understand that it really is a folder entry with an attachment in the underlying architecture.

1.6.3 Replies

A Reply represents a user comment on a file. In the underlying architecture, it's actually a folder entry, but the REST interface treats it as a separate type of object. The text of the user's comment is contained in the description property of the reply object.

A user can reply to the folder entry (the file) or to another reply. So, a reply has a parent entry, which can either be a folder entry or another reply. It also has a top entry, which is always a folder entry.

Users and Groups (Principals)

Together, users and groups are known as principals.

1.6.4 Library Folders

In Vibe, there are a number of different types of folders. Some folder types mirror file system folders and require that files in the folder have unique file names. Others do not have this same requirement. Those folders that require unique file names are called “library folders”.

In Filr, all folders require unique file names and are therefore “library folders”. This is only important for Filr developers to understand because the term “library” appears throughout the Filr REST interface.

2 Admin APIs

2.1 Overview

The REST administration interface can be used to configure settings defined in the Filr administration console. To access the admin interface, find the “admin” link in the root REST resource:

```
[Request]
> curl –k –u admin:novell https://amethyst.wal.novell.com:8443/rest

[Response]
{
   "links":[{
      "rel":"admin",
      "href":"/admin"
   },{
      ...
   },...]
}
```

The root “admin” link will only be present in the root resource if the user whose credentials are supplied to the REST API has permission to access the administration console. Any attempt to make requests to the admin REST resources will result in an HTTP 403 Unauthorized error.

```
[Request]
> curl –k –u admin:novell <a target="_top" href="https://
amethyst.wal.novell.com:8443/rest/admin">
https://amethyst.wal.novell.com:8443/rest/admin
</a>
[Response]
{
   "links":[{
      "rel":"net_folder_servers",
      "href":"/admin/net_folder_servers"
   },{
      "rel":"net_folders",
      "href":"/admin/net_folders"
   },{
```

```
      "rel":"personal_storage",
      "href":"/admin/personal_storage"
   },{
      "rel":"shares",
      "href":"/admin/shares"
   },{
      "rel":"share_settings",
      "href":"/admin/share_settings"
   },{
      "rel":"user_sources",
      "href":"/admin/user_sources"
   },{
      "rel":"user_sources_sync",
      "href":"/admin/user_sources/sync"
   },{
      "rel":"user_sources_sync_config",
      "href":"/admin/user_sources/sync_config"
   },{
      "rel":"web_application",
      "href":"/admin/web_application"
   }]
}
```

2.1.1 Resource Representations

Access

```
{
   "role": string ("NONE", "VIEWER", "EDITOR", "CONTRIBUTOR", "ACCESS"],
   "sharing": SharingPermission Resource
}
```

AssignedRight

```
{
   "principal": Recipient Resource,
   "access": Access Resource
}
```

KeyValuePair

```
{
   "key": string,
   "value": string
}
```

LongIdLinkPair

```
{
   "id": string,
   "href": string
}
```

Schedule

```
{
   "enabled": boolean,
   "when": string ("daily" or "selected_days"),
   "selected_days": SelectedDays Resource,
   "at": Time,
   "every": Time
}
```

SelectedDays

```
{
   "sun": boolean,
   "mon": boolean,
   "tue": boolean,
   "wed": boolean,
   "thu": boolean,
   "fri": boolean,
   "sat": boolean
}
```

SharingPermission

```
{
   "internal": boolean,
   "external": boolean,
   "all_internal": boolean,
   "all_external": boolean,
   "public": boolean,
   "public_link": boolean,
   "grant_reshare": boolean
}
```

Time

```
{
   "hour": integer,
   "minute" integer (0, 15, 30 or 45)
}
```

2.2 Desktop Application

2.2.1 Resource Representations

DesktopAppConfig

```
{
   "enabled": boolean,
   "allow_cached_password": boolean,
   "sync_interval_mins": integer,
   "max_file_size_mbs": integer
}
```

2.2.2 Get Settings

To get the Desktop Application settings, use the desktop_application related link (href: “/admin/desktop_application") from the root admin resource (href: "/admin"):

```
[Request]
> curl -k -u admin:novell https://amethyst.provo.novell.com:8443/rest/admin/
desktop_application

[Response]
{
   "enabled":true,
   "href":"/admin/desktop_application",
   "allow_cached_password":true,
   "sync_interval_mins":15,
   "max_file_size_mbs":50,
}
```

2.2.3 Modify Settings

To update the Web Application settings, send a PUT request to the web_application related link (href: "/admin/desktop_application") from the root admin resource (href: "/admin"). Only the attributes that are included in the request body are updated:

```
[Request]
> curl -k -u admin:novell https://amethyst.provo.novell.com:8443/rest/admin/
desktop_application /
  -X PUT -H "Content-Type: application/json" -d '{"max_file_size_mbs":1024}'

[Response]
{
   "enabled":true,
   "href":"/admin/desktop_application",
   "allow_cached_password":true,
   "sync_interval_mins":15,
   "max_file_size_mbs":1024
}
```

2.3 LDAP

2.3.1 Resource Representations

GroupSynchronization

```
{
   "register": boolean,
   "sync_profiles": boolean,
   "sync_membership": boolean,
   "delete_removed_groups": boolean
}
```

LdapHomeDirConfig

```
{
   "type": string ("custom_net_folder", "home_dir_attribute", "custom_attribute" or
"none"),
   "net_folder_server": LongIdLinkPair Resource,
   "path": string,
   "ldap_attribute": string
}
```

LdapSearchInfo

```
{
   "base_dn": string,
   "filter": string,
   "search_subtree": boolean,
   "home_dir_config": LdapHomeDirConfig Resource
}
```

LdapSyncResults

```
{
   "status": string ("STATUS_COLLECT_RESULTS", "STATUS_COMPLETED",
"STATUS_STOP_COLLECTING_RESULTS",
                     "STATUS_ABORTED_BY_ERROR" or
"STATUS_SYNC_ALREADY_IN_PROGRESS"),
   "added_users": [ string ],
   "modified_users": [ string ],
   "deleted_users": [ string ],
   "disabled_users": [ string ],
   "added_groups": [ string ],
   "modified_groups": [ string ],
   "deleted_groups": [ string ]
}
```

LdapUserSource

```
{
   "id": string,
   "url": string,
   "username_attribute": string,
   "guid_attribute": string,
   "attribute_map": [
      KeyValuePair Resource
   ],
   "user_contexts": [
      LdapSearchInfo Resource
   ],
   "group_contexts": [
      LdapSearchInfo Resource
   ],
   "username": string,
   "password": string
}
```

UserSourceSynchronization

```
{
   "users": UserSynchronization Resource,
   "groups": GroupSynchronization Resource,
   "schedule": Schedule Resource
}
```

UserSynchronization

```
{
   "register": boolean,
   "sync_profiles": boolean,
   "remove_account_action: string ("disable" or "delete"),
   "delete_workspace": boolean,
   "default_timezone": string,
   "default_locale": string
}
```

2.3.2 List

To list all LDAP user sources, use the “user_sources” related link (href: "/admin/user_sources") from the root admin resource (href: "/admin"):

```
[Request]
> curl -k -u admin:novell https://amethyst.provo.novell.com:8443/rest/admin/
user_sources

[Response]
{
   "first":0,
   "count":1,
   "total":1,
   "items":[{
      "id":"09c1c3fb518cf1a001518d01a2b70014",
      "url":"ldap://intlab79.lab.novell.com:389",
      "type":"ldap",
      "href":"/admin/user_sources/09c1c3fb518cf1a001518d01a2b70014",
      "username_attribute":"cn",
      "guid_attribute":"GUID",
      "attribute_map":[{
         "key":"gn",
         "value":"firstName"
      },{
         "key":"mail",
         "value":"emailAddress"
      },{
         "key":"sn",
         "value":"lastName"
      },{
         "key":"description",
         "value":"description"
      },{
         "key":"telephoneNumber",
         "value":"phone"
      },{
         "key":"surname",
         "value":"lastName"
```

```
      },{
         "key":"givenName",
         "value":"firstName"
      }],
      "user_contexts":[{

"filter":"(&amp;(!(objectClass=computer))(|(objectClass=Person)(objectClass=orgPer
son)(objectClass=inetOrgPerson)))",
         "base_dn":"o=novell",
         "search_subtree":false,
         "home_dir_config":{
            "type":"home_dir_attribute"
         }
      }],
      "group_contexts":[{

"filter":"(|(objectClass=group)(objectClass=groupOfNames)(objectClass=groupOfUniqu
eNames))",
         "base_dn":"o=novell",
         "search_subtree":false
      }],
      "username":"cn=superuser,o=novell"
   }]
}
```

2.3.3 Create

To create a new LDAP user source, POST an LdapUserSource object to the “user_sources” related link (href: "/admin/user_sources") from the root admin resource (href: "/admin"):

```
[Request]
> curl -k -u admin:novell https://amethyst.provo.novell.com:8443/rest/admin/
user_sources \
  -X POST -H "Content-Type: application/json" \
  -d '{"url":"ldap://
intlab79.lab.novell.com:389","username_attribute":"cn","guid_attribute":"GUID",
       "username":"cn=superuser,o=novell","password":"pwd",

"user_contexts":[{"base_dn":"o=novell","home_dir_config":{"type":"home_dir_attribu
te"}}],
       "group_contexts":[{"base_dn":"o=novell"}]}'

[Response]
{
...(Full LdapUserSource object)
}
```

2.3.4 Delete

```
[Request]
> curl -k -u admin:novell https://amethyst.provo.novell.com:8443/rest/admin/
user_sources/09c1c3fb518cf1a001518d01a2b70014 -X DELETE

[Empty Response]
```

2.3.5 Sync Config

Get Settings

To get the LDAP Sync Config settings, use the user_source_sync_config related link (href: “/admin/user_sources/sync_config") from the root admin resource (href: "/admin"):

```
[Request]
> curl -k -u admin:novell https://amethyst.provo.novell.com:8443/rest/admin/
user_sources/sync_config

[Response]
{
   "users":{
      "register":true,
      "sync_profiles":true,
      "removed_account_action":"disable",
      "default_timezone":"GMT",
      "default_locale":"en_US"
   },
   "groups":{
      "register":true,
      "sync_profiles":true,
      "sync_membership":true,
      "delete_removed_groups":false
   },
   "schedule":{
      "enabled":false,
      "at":{
         "hour":12,
         "minute":15
      },
      "when":"daily"
   }
}
```

Modify Settings

To modify the LDAP Sync Config settings, PUT a UserSourceSynchronization object to the user_source_sync_config related link (href: “/admin/user_sources/sync_config") from the root admin resource (href: "/admin"). Only the attributes that are specified in the request body are updated.

```
[Request]
> curl -k -u admin:novell https://amethyst.provo.novell.com:8443/rest/admin/
user_sources/sync_config \
  -X PUT -H "Content-Type: application/json" -d '{"schedule":{"enabled":true}}'

[Response]
{
...(Full UserSourceSynchronization object)
}
```

2.3.6 Sync

To sync all LDAP user sources, make a POST request to the "user_sources_sync" related link (href: "/admin/user_sources/sync") from the root admin resource (href: "/admin"):

```
[Request]
> curl -k -u admin:novell https://amethyst.provo.novell.com:8443/rest/admin/
user_sources/sync -X POST

[Response]
{
   "status":"STATUS_COLLECT_RESULTS",
   "added_users":[],
   "modified_users":[],
   "deleted_users":[],
   "disabled_users":[],
   "added_groups":[],
   "modified_groups":[],
   "deleted_groups":[]
}
```

2.4 Net Folder Servers

2.4.1 Resource Representations

NetFolderServer

```
{
   "id": long,
   "name": string,
   "driver_type": string ("windows_server", "oes" or "netware"),
   "server_path": string,
   "proxy_dn": string,
   "proxy_password": string,
   "auth_type": string ("nmas", "ntlm", "kerberos", "kerberos_then_ntlm"),
   "sync_schedule": Schedule Resource,
   "index_content": boolean,
   "jits_enabled": boolean,
   "jits_max_age": long,
   "jits_max_acl_age": long,
   "allow_client_initiated_sync": boolean
}
```

2.4.2 List

To list net folder servers, use the net_folder_servers related link (href: “/admin/net_folder_servers”) from the root admin resource (href: “/admin”):

```
[Request]
> curl –k –u admin:novell https://amethyst.wal.novell.com:8443/rest/admin/
net_folder_servers

[Response]
{
   "first":0,
   "count":2,
   "total":2",
   "items":[{
      "id":1,
      "name":"intlab79.lab.novell.com-HOME",
      "href":"/admin/net_folder_servers/1",
      "links":[{
         "rel":"net_folders",
         "href":"/admin/net_folder_servers/1/net_folders"
      }],
      "driver_type":"oes",
      "auth_type":"nmas",
      "server_path":"\\\\intlab79.lab.novell.com\\HOME",
      "proxy_dn":"cn=superuser,o=novell",
      "index_content": false,
      "jits_enabled": true,
      "jits_max_age": 60000,
      "jits_max_acl_age": 3600000
   },{
      ...
   }]
}
```

Some attributes on net folder servers (the sync schedule) require additional database lookups, so the REST interface does not return them by default in the “net_folder_servers” response. You can override this behavior by adding the include_full_details=true query parameter:

```
[Request]
> curl –k –u admin:novell https://amethyst.wal.novell.com:8443/rest/admin/
net_folder_servers?include_full_details=true

[Response]
{
   "first":0,
   "count":2,
   "total":2",
   "items":[{
      "id":1,
      "name":"intlab79.lab.novell.com-HOME",
      "href":"/admin/net_folder_servers/1",
      "links":[{
         "rel":"net_folders",
         "href":"/admin/net_folder_servers/1/net_folders"
      }],
      "driver_type":"oes",
      "auth_type":"nmas",
      "server_path":"\\\\intlab79.lab.novell.com\\HOME",
      "proxy_dn":"cn=superuser,o=novell",
```

```
      "index_content": false,
      "jits_enabled": true,
      "jits_max_age": 60000,
      "jits_max_acl_age": 3600000,
      "sync_schedule":{
         "enabled":true,
         "when":"daily",
         "at":{
            "hour":0,
            "minute":0
         }
      }
   },{
      ...
   }]
}
```

2.4.3 Create

To create a new net folder server, POST a net folder server object to the net_folder_servers resource:

```
[Request]
> curl -k -u admin:novell https://amethyst.wal.novell.com:8443/rest/admin/
net_folder_servers \
 -X POST -H "Content-Type: application/json" \
 -d
'{"name":"myserver","driver_type":"oes","server_path":"\\\\sisaacson2.provo.novell
.com\\vol1",

"auth_type":"ntlm","proxy_dn":"cn=admin,o=novell","proxy_password":"novell"}'

[Response]
{
 ... (Full net folder server object)
}
```

2.4.4 Modify

To modify an existing net folder server, PUT a net folder server object to the individual net folder server resource. Only the net folder server attributes that are included in the request body are updated.

```
[Request]
> curl -k -u admin:novell https://amethyst.wal.novell.com:8443/rest/admin/
net_folder_servers/1 \
 -X PUT -H "Content-Type: application/json" \
 -d '{"sync_schedule":{"enabled":true,"when":"daily","at":{"hour":1}}}'

[Response]
{
 ... (Full net folder server object)
}
```

2.4.5 Delete

To delete a net folder server, send a DELETE request to the individual net folder server resource:

```
[Request]
> curl -k -u admin:novell https://amethyst.wal.novell.com:8443/rest/admin/
net_folder_servers/1 -X DELETE

[Response body is empty]
```

2.5 Net Folders

2.5.1 Resource Representations

NetFolder

```
{
   "id": long,
   "name": string,
   "server": LongIdLinkPair Resource,
   "relative_path": string,
   "home_dir": boolean,
   "inherit_sync_schedule": boolean,
   "sync_schedule": Schedule Resource,
   "assigned_rights": [
      AssignedRight Resource
   ],
   "inherit_index_context": boolean,
   "index_content": boolean,
   "inherit_jits_settings": boolean,
   "jits_enabled": boolean,
   "jits_max_age": long,
   "jits_max_acl_age": long,
   "inherit_client_sync_settings": boolean,
   "allow_desktop_sync": boolean,
   "allow_client_initiated_sync": boolean
}
```

NetFolderSyncStatus

```
{
   "status": string ("ready", "taken", "started", "stopped", "finished", "aborted",
"canceled", or "deleting"),
   "node_ip_address": string,
   "start_date": dateTime,
   "end_date": dateTime,
   "directory_only": boolean,
   "directory_enumeration_failure": boolean,
   "files_found": integer,
   "files_added": integer,
   "files_expunged": integer,
   "files_modified": integer,
   "files_with_modified_acl": integer,
   "files_with_modified_owner: integer,
   "folder_found": integer,
   "folders_added": integer,
   "folders_expunged": integer,
   "folders_with_modified_acl": integer,
   "folders_with_modified_owner": integer,
   "entries_expunged": integer,
   "failures": integer,
   "folders_processed": integer
}
```

2.5.2 List

List All Net Folders

To list all net folders (including home folders), use the “net_folders” link from the root admin resource:

```
[Request]
> curl –k –u admin:novell https://amethyst.wal.novell.com:8443/rest/admin/
net_folders

[Response]
{
   "first":0,
   "count":2,
   "total":2",
   "items":[{
      "id":99,
      "name":"Home",
      "href":"/admin/net_folders/99",
      "server":{
         "id":1,
         "href":"/admin/net_folder_servers/1"
      },
      "relative_path":"dlewis",
      "home_dir":true,
      "links":[{
         "rel":"sync",
```

```
         "href":"/admin/net_folders/99/sync"
      }],
      "inherit_client_sync_settings": true,
      "inherit_index_content": true,
      "inherit_jits_settings": true,
      "inherit_sync_schedule": true,
      "index_content": false,
      "jits_enabled": true,
      "jits_max_age": 60000,
      "jits_max_acl_age": 3600000,
      "allow_client_initiated_sync": false,
      "allow_desktop_sync": true,
   },{
      ...
   }]
}
```

To list only home folders, include the "type=home" query parameter:

```
[Request]
> curl –k –u admin:novell https://amethyst.wal.novell.com:8443/rest/admin/
net_folders?type=home
```

To exclude home folders from the results, add the "type=net" query parameter:

```
[Request]
> curl –k –u admin:novell https://amethyst.wal.novell.com:8443/rest/admin/
net_folders?type=net
```

List Net Folders associated with Net Folder Server

To list net folders associated with a particular net folder server, use the “net_folders” related link from the individual net folder server object (href: "/net_folder_servers/{id}/net_folders"):

```
[Request]
> curl –k –u admin:novell https://amethyst.wal.novell.com:8443/rest/admin/
net_folder_servers/1/net_folders

[Response similar to previous response]
```

Listing Full Details of Net Folders

Some attributes of net folders (assigned rights and sync schedule) require additional database lookups, so they are not returned by default. To return this information in the listings, add the “include_full_details=true” query parameter:

```
[Request]
> curl –k –u admin:novell \
  https://amethyst.wal.novell.com:8443/rest/admin/
net_folders?include_full_details=true

[Alternate Request]
> curl -k -u admin:novell \
  https://amethyst.wal.novell.com:8443/rest/admin/net_folder_servers/1/
net_folders?include_full_details=true

[Response]
{
   "first":0,
   "count":2,
   "total":2",
   "items":[{
      "id":99,
      "name":"Home",
      "href":"/admin/net_folders/99",
      "server":{
         "id":1,
         "href":"/admin/net_folder_servers/1"
      },
      "relative_path":"dlewis",
      "home_dir":true,
      "links":[{
         "rel":"sync",
         "href":"/admin/net_folders/99/sync"
      }],
      "inherit_client_sync_settings": true,
      "inherit_index_content": true,
      "inherit_jits_settings": true,
      "inherit_sync_schedule": true,
      "index_content": false,
      "jits_enabled": true,
      "jits_max_age": 60000,
      "jits_max_acl_age": 3600000,
      "allow_client_initiated_sync": false,
      "allow_desktop_sync": true,
      "assigned_rights":[{
         "access":{
            "role":"ACCESS",
            "sharing":{
               "internal":true,
               "external":false,
               "public":false,
               "public_link":false,
               "grant_reshare":true
            }
         },
         "principal":{
            "id":116,
            "type":"group",
            "href":"/groups/116"
```

```
         }
      }],
      "sync_schedule":{
         "enabled":true,
         "when":"daily",
         "at":{
            "hour":0,
            "minute":0
         }
      }
   },{
      ...
   }]
}
```

2.5.3 Create

You can create a new folder by POSTing a net folder object to either of the net_folders resources:

```
[Request]
> curl -k -u admin:novell https://amethyst.wal.novell.com:8443/rest/admin/
net_folder_servers/5/net_folders \
  -X POST -H "Content-Type: application/json" -d
'{"name":"testing5","relative_path":"testing"}'

[Alternate Request]
> curl -k -u admin:novell https://amethyst.wal.novell.com:8443/rest/admin/
net_folders -X POST \
  -H "Content-Type: application/json" -d
'{"name":"testing6","server":{"id":5},"relative_path":"testing"}'

 [Response]
{
 ... (Full net folder object)
}
```

Notice that the net folder server (id: 5) must be specified in either the request URL or the POST body.

2.5.4 Modify

To modify an existing net folder, PUT a net folder object to the individual net folder server resource. Only the net folder attributes that are included in the request body are updated.

```
[Request]
> curl -k -u admin:novell https://amethyst.wal.novell.com:8443/rest/admin/
net_folders/1 \
 -X PUT -H "Content-Type: application/json" \
 -d '{"sync_schedule":{"enabled":true,"when":"daily","at":{"hour":1}}}'

[Response]
{
 ... (Full net folder object)
}
```

2.5.5 Delete

```
[Request]
> curl -k -u admin:novell https://amethyst.wal.novell.com:8443/rest/admin/
net_folders/99 -X DELETE

[Empty Response]
```

2.5.6 Net Folder Sync

Get Sync Status

To get the sync status for a net folder, use the “sync” related link from the net folder object:

```
[Request]
 > curl -u admin:novell -k https://amethyst.wal.novell.com:8443/rest/admin/
net_folders/99/sync

[Response]
{
   "status":"finished",
   "failures":0,
   "node_ip_address":"164.99.211.53",
   "start_date":"2014-10-22T00:00:00Z",
   "end_date":"2014-10-22T00:00:05Z",
   "directory_only":false,
   "directory_enumeration_failure":false,
   "files_found":54,
   "files_added":0,
   "files_expunged":0,
   "files_modified":0,
   "files_with_modified_acl":0,
   "files_with_modified_owner":0,
   "folders_found":10,
   "folders_added":0,
   "folders_expunged":0,
   "folders_with_modified_acl":0,
   "folders_with_modified_owner":0,
   "entries_expunged":0,
   "folders_processed":11
}
```

Status is one of “none”, ”ready”, ”taken”, ”started”, ”stopped”, ”finished”, “aborted”, “canceled”, or “deleting”. Of those, “none, “ready”, “started” and “finished” are the most common.

Trigger Net Folder Sync

To cause Filr to synchronize a net folder, POST to the net folder’s “sync” link:

```
[Request]
> curl -u admin:novell -k https://amethyst.wal.novell.com:8443/rest/admin/
net_folders/99/sync -X POST

[Response]
{
... (Same as above)
}
```

Stop Net Folder Sync

You can stop an in-progress sync by sending a DELETE request to the sync link:

```
[Request]
> curl -u admin:novell -k https://amethyst.wal.novell.com:8443/rest/admin/
net_folders/99/sync -X DELETE

[Response]
{
... (Same as above)
}
```

2.6 Personal Storage

2.6.1 Resource Representations

PersonalStorage

```
{
   "allow_personal_storage": boolean
}
```

2.6.2 Get Settings

To get the current personal storage settings, use the personal_storage related link (href: "/admin/personal_storage") from the root admin resource (href: "/admin"):

```
[Request]
> curl -k -u admin:novell https://amethyst.provo.novell.com:8443/rest/admin/
personal_storage

[Response]
{
   "href":"/admin/personal_storage",
   "allow_personal_storage":false
}
```

2.6.3 Modify Settings

To update the current personal storage settings, send a PUT request to the personal_storage related link (href: "/admin/personal_storage") from the root admin resource (href: "/admin"):

```
[Request]
> curl -k -u admin:novell https://amethyst.provo.novell.com:8443/rest/admin/
personal_storage \
  -X PUT -H "Content-Type: application/json" -d '{"allow_personal_storage":true}'

[Response]
{
   "href":"/admin/personal_storage",
   "allow_personal_storage":true
}
```

2.7 Share Settings

2.7.1 Resource Representations

AssignedSharingPermission

```
{
   "principal": Recipient Resource,
   "sharing": SharingPermission Resource
}
```

ExternalSharingRestriction

```
{
   "mode": string ("none", "whitelist" or "blacklist"),
   "email_list": [ string ],
   "domain_list": [ string ]
}
```

Recipient

```
{
   "type": string ("user" or "group")
   "id": long,
   "href": string
}
```

ShareSettings

```
{
   "allow_sharing_with_ldap_groups": boolean,
   "sharing_permissions": [
      AssignedSharingPermission Resource
   ],
   "external_restrictions": ExternalSharingRestrictions Resource
}
```

2.7.2 Get Settings

```
[Request]
> curl -k -u admin:novell https://amethyst.provo.novell.com:8443/rest/admin/
share_settings

[Response]
{
   "href":"/admin/share_settings",
   "links":[{
      "rel":"permissions",
      "href":"/admin/share_settings/permissions"
   },{
      "rel":"external_restrictions",
      "href":"/admin/share_settings/external_restrictions"
   }],
   "allow_sharing_with_ldap_groups":true,
   "sharing_permissions":[{
      "principal":{
         "id":1,
         "type":"user",
         "href":"/users/1"
      },
      "sharing":{
         "internal":true,
         "external":true,
         "public":true,
         "all_internal":true,
         "all_external":false,
         "public_link":true,
         "grant_reshare":true
      }
   },{
      "principal":{
         "id":2,
         "type":"group",
         "href":"/groups/2"
      },
      "sharing":{
         "internal":true,
```

```
         "external":true,
         "public":true,
         "all_internal":true,
         "all_external":false,
         "public_link":true,
         "grant_reshare":true
      }
   }],
   "external_restrictions":{
      "mode":"none",
      "email_list":[],
      "domain_list":[]
   }
}
```

2.7.3 Modify Settings

To modify the share settings, PUT a ShareSettings object to the share_settings related link (href: "/admin/share_settings"). Only the attributes that are included in the request body are updated.

```
[Request]
> curl -k -u admin:novell https://amethyst.provo.novell.com:8443/rest/admin/
share_settings \
  -X PUT -H "Content-Type: application/json" -d
'{"allow_sharing_with_ldap_groups":false}'

[Response]
{
...(ShareSettings Resource)
}
```

- “Modify Sharing Permissions” on page 32
- “Modify External Sharing Restrictions” on page 33

Modify Sharing Permissions

The permissions related link (href: "/admin/sharing/permissions") allows you to deal with just the permissions portion of Share Settings.

```
[Request]
> curl -k -u admin:novell https://amethyst.provo.novell.com:8443/rest/admin/
share_settings/permissions

[Response]
[{
   "principal":{
      "id":1,
      "type":"user",
      "href":"/users/1"
   },
   "sharing":{
      "internal":true,
      "external":true,
      "public":true,
      "all_internal":true,
      "all_external":false,
      "public_link":true,
      "grant_reshare":true
```

```
   }
},{
   "principal":{
      "id":2,
      "type":"group",
      "href":"/groups/2"
   },
   "sharing":{
      "internal":true,
      "external":true,
      "public":true,
      "all_internal":true,
      "all_external":false,
      "public_link":true,
      "grant_reshare":true
   }
}]
```

To add a new permission, POST an AssignedSharingPermission to the permissions related link:

```
[Request]
> curl -k -u admin:novell https://amethyst.provo.novell.com:8443/rest/admin/
share_settings/permissions \
  -X POST -H "Content-Type: application/json" -d
'{"principal":{"id":20,"type":"user"},"sharing":{"internal":true}}'

[Response]
[
...(full list of AssigneSharingPermission objects)
]
```

To replace the whole list of permissions, PUT a list of AssignedSharingPermission objects to the permissions related link:

```
[Request]
> curl -k -u admin:novell https://amethyst.provo.novell.com:8443/rest/admin/
share_settings/permissions \
  -X PUT -H "Content-Type: application/json" \
  -d '[{"principal":{"id":20,"type":"user"},"sharing":{"internal":true}},
       {"principal":{"id":21,"type":"user"},"sharing":{"internal":true}}]'

[Response]
[
...(full list of AssignedSharingPermission objects)
]
```

To delete all permissions, make a DELETE request to the permissions related link:

```
[Request]
> curl -k -u admin:novell https://amethyst.provo.novell.com:8443/rest/admin/
share_settings/permissions -X DELETE

[Empty Response]
```

Modify External Sharing Restrictions

The external_restrictions related link (href: "/admin/sharing/external_restrictions") allows you to deal with just the external sharing restrictions of the Share Settings.

```
[Request]
&gt; curl -k -u admin:novell https://amethyst.provo.novell.com:8443/rest/admin/
share_settings/external_restrictions

[Response]
{"mode":"none","email_list":[],"domain_list":[]}
```

To update the external sharing restrictions settings, PUT an ExternalSharingRestrictions object to the external_restrictions related link.

```
[Request]
> curl -k -u admin:novell https://amethyst.provo.novell.com:8443/rest/admin/
share_settings/external_restrictions \
  -X PUT -H "Content-Type: application/json" \
  -d '{"mode":"whitelist", "domain_list":["novell.com","microfocus.com"]}'

[Response]
{"mode":"whitelist","email_list":[],"domain_list":["microfocus.com","novell.com"]}
```

2.8 Shares

2.8.1 Resource Representations

Share

```
{
   "id": long,
   "comment": string,
   "sharer": LongIdLinkPair Resource,
   "sharing_date": dateTime,
   "days_to_expire": integer,
   "expiration": dateTime,
   "recipient": ShareRecipient Resource,
   "shared_entity": EntityId Resource,
   "role": string ("VIEWER", "CONTRIBUTOR" or "EDITOR"),
   "access": Access Resource
}
```

ShareRecipient

```
{
   "type": string ("user", "group", "external_user", "public", or "public_link")
   "id": long,
   "email": string
}
```

2.8.2 List

To list all shares in the system, use the shares related link (href: "/admin/shares") from the root admin resource (href: "/admin"):

```
[Request]
> curl -k -u admin:novell https://amethyst.provo.novell.com:8443/rest/admin/shares

[Response]
{
   "first":0,
   "count":1,
   "total":1,
   "items":[{
      "id":1,
      "comment":"",
      "sharer":{
         "id":20,
         "href":"/users/20"
      },
      "recipient":{
         "id":140,
         "type":"external_user",
         "href":"/users/140",
         "email":"siguat@gmail.com"
      },
      "role":"VIEWER",
      "access":{
         "role":"VIEWER",
         "sharing":{
            "internal":false,
            "external":false,
            "public":false,
            "public_link":false,
            "grant_reshare":false
         }
      },
      "href":"/admin/shares/1",
      "sharing_date":"2015-12-22T17:31:12Z",
      "shared_entity":{
         "id":10,
         "type":"folderEntry",
         "href":"/folder_entries/10"
      },
      "can_share":false
   }]
}
```

List By Sharer

Use the shared_by query parameter to list the shares by sharer:

```
[Request]
> curl -k -u admin:novell https://amethyst.provo.novell.com:8443/rest/admin/
shares?shared_by=20

[Response]
Similar to above
```

List By Recipient

Use the shared_with query parameter to list the shares by recipient:

```
[Request]
> curl -k -u admin:novell https://amethyst.provo.novell.com:8443/rest/admin/
shares?shared_with=140

[Response]
Similar to above
```

List Public Shares

To list all public shares in the system, use the public_shares related link (href: "/admin/shares/public") from the root admin resource (href: "/admin"):

```
[Request]
> curl -k -u admin:novell https://amethyst.provo.novell.com:8443/rest/admin/
shares/public

[Response]
Similar to above
```

2.8.3 Delete

To delete a share, send a DELETE request to the individual share resource:

```
[Request]
> curl -k -u admin:novell https://amethyst.wal.novell.com:8443/rest/admin/shares/
24 -X DELETE

[Empty Response]
```

2.9 Web Application

2.9.1 Resource Representations

WebAppConfig

```
{
   "enabled": boolean,
   "allow_guest_access": boolean,
   "read_only_guest": boolean,
   "allow_open_id": boolean,
   "allow_downloads": boolean
}
```

2.9.2 Get Settings

To get the Web Application settings, use the web_application related link (href: “/admin/web_application") from the root admin resource (href: "/admin"):

```
[Request]
> curl -k -u admin:novell https://amethyst.provo.novell.com:8443/rest/admin/
web_application

[Response]
{
   "enabled":true,
   "href":"/admin/web_application",
   "allow_guest_access":true,
   "read_only_guest":true,
   "allow_open_id":true,
   "allow_downloads":true
}
```

2.9.3 Modify Settings

To update the Web Application settings, send a PUT request to the web_application related link (href: "/admin/web_application") from the root admin resource (href: "/admin"). Only the attributes that are included in the request body are updated:

```
[Request]
> curl -k -u admin:novell https://amethyst.provo.novell.com:8443/rest/admin/
web_application /
  -X PUT -H "Content-Type: application/json" -d '{"allow_guest_access":false}'

[Response]
{
   "enabled":true,
   "href":"/admin/web_application",
   "allow_guest_access":false,
   "read_only_guest":true,
   "allow_open_id":true,
   "allow_downloads":true
}
```

3 Client APIs

3.1 Overview

The entry point to the REST client API is https://[serveraddress]/rest. This root resource contains a list of related links.

```
[Request]
> curl –k –u admin:novell https://amethyst.wal.novell.com:8443/rest

[Response]
{
   "links":[{
      "rel":"binders",
      "href":"/binders"
   },{
      "rel":"library_entities",
      "href":"/workspaces/1/library_entities"
   },{
      "rel":"net_folders",
      "href":"/net_folders"
   },{
      "rel":"principals",
```

```
      "href":"/principals"
   },{
      "rel":"release_info",
      "href":"/release_info"
   },{
      "rel":"self",
      "href":"/self"
   },{
      "rel":"zone_config",
      "href":"/zone_config"
   },{
     ...
   },...]
}
```

3.1.1 Resource Representations

BaseRestObject

```
{
  "href":string,
  "links":[
    Link Resource
  ]
}
```

Binder

Extends DefinableEntity

```
{
  "path": string,
  "library_info": LibraryInfo Resource
}
```

BinderBrief

Extends DefinableEntityBrief

```
{
  "path": string,
  "library": boolean,
  "mirrored": boolean,
  "home_dir": boolean,
  "library_info": LibraryInfo Resource
}
```

DefinableEntity

Extends SearchableObject

```
{
  "id": long,
  "parent_binder": ParentBinder Resource,
  "title": string,
  "description": Description Resource,
  "entity_type": string ("folder", "folder_entry", "group", "user", or
"workspace"),
  "permalink": string,
  "creation": HistoryStamp Resource,
  "modification": HistoryStamp Resource
}
```

DefinableEntityBrief

Extends SearchableObject

```
 "id": long,
 "parent_binder": ParentBinder Resource,
 "title": string,
 "description": Description Resource,
 "entity_type": string ("folder", "folder_entry", "group", "user", or "workspace"),
 "permalink": string,
 "creation": HistoryStamp Resource,
 "modification": HistoryStamp Resource
```

Description

```
{
  "text": string,
  "format": integer (1=html, 2=text),
  "format_str": string ("html" or "text")
}
```

DesktopAppConfig

```
{
  "enabled": boolean,
  "allow_cached_password", boolean,
  "auto_update_url": string,
  "max_file_size": long,
  "sync_interval": integer,
  "process_config": DesktopAppProcessConfigApp Resource
}
```

DesktopAppProcessConfig

```
{
  "allow_unlisted_processes": boolean,
  "allow_unlisted_process_override": boolean,
  "allowed_processes":[ string ],
  "blocked_processes":[ string ]
}
```

EntityId

```
{
  "id": long,
  "href": string,
  "type": string ("folder", "folderEntry", "user", "group" or "workspace")
}
```

FileProperties

```
{
  "id": string,
  "name": string,
  "doc_type": "file",
  "creation": HistoryStamp Resource,
  "modification": HistoryStamp Resource,
  "length": long,
  "md5": string,
  "version_number": integer,
  "owning_entity": EntityId Resource,
  "parent_binder": ParentBinder Resource,
  "permalink": string,
  "links": [
    Link Resource
  ]
}
```

Folder

Extends Binder

```
{
  "library": boolean,
  "mirrored": boolean
}
```

HistoryStamp

```
{
  "principal": LongIdLinkPair,
  "date": iso_8601_date
}
```

LibraryInfo

```
{
  "mod_date": iso_8601_date,
  "disk_space": long,
  "file_count": integer,
  "folder_count": integer,
  "mirrored_sync_date": iso_8601_date
}
```

Link

```
{
  "rel": string,
  "href": string
}
```

Locale

```
{
  "country": string,
  "language": string
}
```

LongIdLinkPair

```
{
  "id": long,
  "href": string
}
```

MobileAppConfig

```
{
  "enabled": boolean,
  "allow_cached_password": boolean,
  "allow_cached_content": boolean,
  "allow_play_with_other_apps": boolean,
  "force_pin_code": boolean,
  "sync_interval": integer,
  "allow_cut_copy": boolean,
  "allow_screen_capture": boolean,
  "allow_rooted_device": boolean,
  "allow_open_in_apps": string ("all", "none" or "selected"),
  "android_app_whitelist": [
    string
  ],
  "ios_app_whitelist": [
    string
  ]
}
```

ParentBinder

```
{
  "id": long,
  "href": string,
  "path": string
}
```

Principal

Extends DefinableEntity

```
{
  "disabled": boolean,
  "email": string,
  "name": string
}
```

SearchableObject

Extends BaseRestObject

```
{
  "doctype": string ("binder", "file" or "entry")
}
```

StringIdLinkPair

```
{
  "id": string,
  "href": string
}
```

User

Extends Principal

```
{
  "first_name": string,
  "middle_name": string,
  "last_name": string,
  "organization": string,
  "phone": string,
  "locale": Locale Resource,
  "time_zone": string,
  "avatar": StringIdLinkPair resource,
  "disk_space_quota": long,
  "disk_space_used": long,
  "file_size_limit": long,
  "hidden_files_folder": LongIdLinkPair Resource,
  "workspace": LongIdLinkPair Resource,
  "mobile_app_config": MobileAppConfig Resource,
  "desktop_app_config": DesktopAppConfig Resource,
  "groups": [
    LongIdLinkPair Resource
  ]
}
```

Workspace

Extends Binder

```
{
}
```

3.2 Comments

The examples that follow illustrate how to view, add, edit and delete comments on this file:

```
{
  "id":"09c1c3fb5256e2e4015256e8691c003b",
  "creation":{"principal":{"id":20,"href":"/users/20"},"date":"2016-01-
18T22:43:37Z"},
  "modification":{"principal":{"id":20,"href":"/users/20"},"date":"2014-04-
24T17:05:40Z"},
  "length":14,
  "name":"test.txt",
  "href":"/files/09c1c3fb5256e2e4015256e8691c003b/metadata",
  "links":[...],
  "doc_type":"file",
  "version_number":1,
  "owning_entity":{"id":15,"type":"folderEntry","href":"/folder_entries/15"},
  "parent_binder":{"id":44,"href":"/binders/44"}
}
```

The owning folder entry (found in the <code>owning_entity</code> property) is the starting point for comments:

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/
folder_entries/15

[Response]
{
  "id":15,
  "title":"test.txt",
  "href":"/folder_entries/15",
  "doc_type":"entry",
  "entity_type":"folderEntry",
  "entry_type":"entry",
  "creation":{"principal":{"id":20,"href":"/users/20"},"date":"2016-01-
18T22:43:37Z"},
  "modification":{"principal":{"id":20,"href":"/users/20"},"date":"2016-01-
18T22:43:37Z"},
  "parent_binder":{"id":44,"href":"/binders/44"},
  "attachments":[{"id": "09c1c3fb5256e2e4015256e8691c003b",...}],
  "links":[{
    "rel":"replies",
    "href":"/folder_entries/15/replies"
  },{
    "rel":"reply_tree",
    "href":"/folder_entries/15/reply_tree"
  },...],
  "reply_count":2,
  "total_reply_count":3
}
```

The <code>reply_count</code> property indicates that there are two direct replies to this folder entry. Including replies to those replies, there are 3 replies in all (<code>total_reply_count</code>).

3.2.1 List comments (replies)

To list one level of comments, use the <code>replies</code> related link of the FolderEntry or Reply object.

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/
folder_entries/15/replies

[Response]
{
  "first":0,
  "count":2,
  "total":2,
  "items":[{
    "id":67,
    "href":"/replies/67",
    "title":"Re: test.txt",
    "description":{
      "text":"Nice work!",
      "format_str":"text"
    },
    "creation":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
29T16:15:41Z"},
    "modification":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
29T16:15:41Z"},
    "links":[
      {"rel":"replies","href":"/replies/67/replies"},
      {"rel":"reply_tree","href":"/replies/67/reply_tree"}
    ],
    "doc_type":"entry",
    "entity_type":"folderEntry",
    "entry_type":"reply",
    "parent_entry":{"id":15,"href":"/folder_entries/15"},
    "top_entry":{"id":15,"href":"/folder_entries/15",
    "parent_binder":{"id":44,"href":"/binders/44"},
    "reply_count":1,
    "total_reply_count":1
  },{
    "id":106,
    "href":"/replies/106",
    "title":"Re: test.txt",
    "description":{
      "text":"I made a few more changes.",
      "format_str":"text"
    },
    "creation":{"principal":{"id":34,"href":"/users/34"},"date":"2016-02-
29T16:17:00Z"},
    "modification":{"principal":{"id":34,"href":"/users/34"},"date":"2016-02-
29T16:17:00Z"},
    "links":[
      {"rel":"replies","href":"/replies/106/replies"},
      {"rel":"reply_tree","href":"/replies/106/reply_tree"}
    ],
    "doc_type":"entry",
    "entity_type":"folderEntry",
    "entry_type":"reply",
    "parent_binder":{"id":44,"href":"/binders/44"},
    "parent_entry":{"id":15,"href":"/folder_entries/15"},
    "top_entry":{"id":15,"href":"/folder_entries/15"},
    "reply_count":0,
    "total_reply_count":0
  }]
}
```

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/replies/67/
replies

[Response]
{
  "first":0,
  "count":1,
  "total":1,
  "items":[{
    "id":105,
    "href":"/replies/105",
    "title":"Re: Re: test.txt",
    "description":{
      "text":"Thanks.",
      "format_str":"text"
    },
    "creation":{"principal":{"id":34,"href":"/users/34"},"date":"2016-02-
29T16:16:53Z"},
    "modification":{"principal":{"id":34,"href":"/users/34"},"date":"2016-02-
29T16:16:53Z"},
    "links":[
      {"rel":"replies","href":"/replies/105/replies"},
      {"rel":"reply_tree","href":"/replies/105/reply_tree"}
    ],
    "doc_type":"entry",
    "entity_type":"folderEntry",
    "entry_type":"reply",
    "parent_entry":{"id":67,"href":"/folder_entries/67"},
    "top_entry":{"id":15,"href":"/folder_entries/15",
    "parent_binder":{"id":44,"href":"/binders/44"},
    "reply_count":0,
    "total_reply_count":0
  }]
}
```

3.2.2 List comment tree (reply_tree)

To get the all comments and their comments in a single request, use the <code>reply_tree</code> related link of the FolderEntry or Reply object:

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/
folder_entries/15/reply_tree

[Response]
{
  "children":[{
    "item":{
      "id":67,
      "title":"Re: test.txt",
      "description":{"text":"Nice work!","format_str":"text"},
      ...
    },
    "children":[{
      "item":{
        "id":105,
```

```
        "title":"Re: Re: test.txt",
        "description":{"text":"Thanks.","format_str":"text"},
        ...
      }
    }]
  },{
    "item":{
      "id":106,
      "title":"Re: test.txt",
      "description":{"text":"I made a few more changes.","format_str":"text"},
      ...
    }
  }]
}
```

3.2.3 Add comment

To add a comment, POST a Reply object to the "replies" related link of the desired FolderEntry or Reply object. Only the description text is required.

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/
folder_entries/15/replies \
  -H "Content-Type: application/json" -X POST -d '{"description":
{"text":"Great!"}}'

[Response]
{
  "id":107,
  "title":"Re: test.txt",
  "description":{"text":"Great!","format_str":"text"},
  "creation":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
29T22:31:28Z"},
  "modification":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
29T22:31:28Z"},
  "href":"/replies/107",
  "links":[...],
  "parent_binder":{"id":44,"href":"/binders/44"},
  "entry_type":"reply",
  "reply_count":0,
  "total_reply_count":0,
  "parent_entry":{"id":15,"href":"/folder_entries/15"},
  "top_entry":{"id":15,"href":"/folder_entries/15"}
}
```

3.2.4 Edit comment

To modify a comment, PUT a Reply object to the href of the Reply to update.

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/replies/107
\
  -H "Content-Type: application/json" -X PUT -d '{"description": {"text":"Really
great!"}}'

[Response]
{
  "id":107,
  "title":"Re: test.txt",
  "description":{"text":"Really great!","format_str":"text"},
  "creation":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
29T22:31:28Z"},
  "modification":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
29T22:34:37Z"},
  "href":"/replies/107",
  "links":[...],
  "parent_binder":{"id":44,"href":"/binders/44"},
  "entry_type":"reply",
  "reply_count":0,
  "total_reply_count":0,
  "parent_entry":{"id":15,"href":"/folder_entries/15"},
  "top_entry":{"id":15,"href":"/folder_entries/15"}
}
```

3.2.5 Delete comment

To delete a comment and all it's child comments, send a DELETE request to the the href of the Reply to update. If the delete is successful, the server responds with an HTTP 204 (No Content) status and an empty response body.

```
[Request]
> curl -v -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/replies/
107 -X DELETE

[Response]
< HTTP/1.1 204 No Content
```

3.3 Files

3.3.1 Download file

Each file resource has a related link named "content" (href: "/files/{id}") that can be used to download the file content. For example, take the following folder listing:

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/self/
my_files/library_files

[Response]
{
  "first": 0,
  "count": 15,
  "total": 15,
  "items": [{
    "id": "09c1c3fb5256e2e4015256e8691c003b",
    "doc_type": "file",
    "href": "/files/09c1c3fb5256e2e4015256e8691c003b/metadata",
    "name": "test.txt",
    "length": 14,
    "links":[{
      "rel": "content",
      "href": "/files/09c1c3fb5256e2e4015256e8691c003b"
    },...],
    ...
  },...
}
```

The following illustrates how to download the file:

```
[Request]
> curl -k -u dlewis:novell \
  https://amethyst.provo.novell.com:8443/rest/files/
09c1c3fb5256e2e4015256e8691c003b -o test.txt

[Response]
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    14  100    14    0     0     28      0 --:--:-- --:--:-- --:--:--    28
```

3.3.2 Upload new file

To upload a new file, use the "child_library_folders" link of the parent folder where the file should reside. For example, take the following folder listing:

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/self/
my_files/library_children

[Response]
{
  "first": 0,
  "count": 24,
  "total": 24,
  "items": [{
    "id": 48,
    "doc_type": "binder",
    "entity_type": "folder",
    "href": "/folders/48",
    "title": "A",
    "links":[{
      "rel": "child_library_files",
      "href": "/folders/48/library_files"
    },...],
    ...
  },...
}
```

The "child_library_files" location for this folder is "/folders/48/library_files". To upload a file send a POST request to that resource. You must use the file_name query parameter to specify the name of the file. If successful, the response is the FileProperties representation of the newly added file.

```
[Request]
>  curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/folders/
48/library_files?file_name=aaa.txt \
   -X POST -H "Content-Type: application/octet-stream" --upload-file ./test.txt

[Response]
{
  "id":"09c1c3fb530f562401531018f4270000",
  "creation":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
23T21:46:23Z"},
  "modification":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
23T21:46:23Z"},
  "length":14,
  "md5":"07b4778d21a1673c60bb650a7c1a4055",
  "name":"aaa.txt",
  "href":"/files/09c1c3fb530f562401531018f4270000/metadata",
  "links":[...],
  "doc_type": "file",
  "version_number": 1,
  "owning_entity": {"id":61,"type":"folderEntry","href":"/folder_entries/61"},
  "parent_binder": {"id":48,"href":"/binders/48"}
}
```

The REST interface also supports "Content-Type: multipart/form-data".

If a file already exists with that name, the server will return an HTTP 409 Conflict status code. The response body will be an ErrorInfo object with a "FILE_EXISTS" code and the existing FileProperties resource:

```
[Request]
>  curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/folders/
48/library_files?file_name=aaa.txt \
   -X POST -H "Content-Type: application/octet-stream" --upload-file ./test.txt

[Response]
{
  "code":"FILE_EXISTS",
  "message":"A file with the name already exists.",
  "data":{
    "id":"09c1c3fb530f562401531018f4270000",
    "creation":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
23T21:46:23Z"},
    "modification":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
23T21:46:23Z"},
    "length":14,
    "md5":"07b4778d21a1673c60bb650a7c1a4055",
    "name":"aaa.txt",
    "href":"/files/09c1c3fb530f562401531018f4270000/metadata",
    "links":[...],
    "doc_type": "file",
    "version_number":1,
    "owning_entity":{"id":61,"type":"folderEntry","href":"/folder_entries/61"},
    "parent_binder":{"id":48,"href":"/binders/48"}
  }
}
```

3.3.3 Replace existing file

There are two ways to overwrite an existing file. The first option is upload it as a new file to the parent folder's "child_library_files" link and add the "overwrite_existing=true" query parameter:

```
[Request]
>  curl -k -u dlewis:novell \
   "https://amethyst.provo.novell.com:8443/rest/folders/48/
library_files?file_name=aaa.txt&overwrite_existing=true" \
   -X POST -H "Content-Type: application/octet-stream" --upload-file ./test.txt

[Response]
{
  "id":"09c1c3fb530f562401531018f4270000",
  "creation":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
23T21:46:23Z"},
  "modification":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
23T21:59:24Z"},
  "length":24,
  "md5":"7eda124afb39fddc0b2e11ce45662e75",
  "name":"aaa.txt",
  "href":"/files/09c1c3fb530f562401531018f4270000/metadata",
  "links":[...],
  "doc_type": "file",
  "version_number": 2,
  "owning_entity": {"id":61,"type":"folderEntry","href":"/folder_entries/61"},
  "parent_binder": {"id":48,"href":"/binders/48"}
}
```

If no file exists in the folder with the given name, the file will add to the folder as a new file.

The other option is to send a POST request to the "content" related link of the existing FileProperties object (href: "/files/{id}"). You must specify one of two query parameters: "force_overwrite=true" or "last_version={version_number}". The "force_overwrite=true" option will overwrite the current version of the file without doing any version conflict checks:

```
[Request]
>  curl -k -u dlewis:novell \
   "https://amethyst.provo.novell.com:8443/rest/files/
09c1c3fb530f562401531018f4270000?force_overwrite=true" \
   -X POST -H "Content-Type: application/octet-stream" --upload-file ./test.txt

[Response]
{
  "id":"09c1c3fb530f562401531018f4270000",
  "creation":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
23T21:46:23Z"},
  "modification":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
23T22:07:13Z"},
  "length":45,
  "md5":"919fd4f5a0cd091d34aeb09583068834",
  "name":"aaa.txt",
  "href":"/files/09c1c3fb530f562401531018f4270000/metadata",
  "links":[...],
  "doc_type": "file",
  "version_number": 3,
  "owning_entity": {"id":61,"type":"folderEntry","href":"/folder_entries/61"},
  "parent_binder": {"id":48,"href":"/binders/48"}
}
```

However, if the client application can track file version numbers, specifying "last_version={version_number}" is the better option because it prevents users from overwriting each others' modifications to the file. If user A downloads and edits version 3 of the file in the client application, the client application can upload the new version of the file as follows:

```
[Request]
>  curl -k -u dlewis:novell \
   "https://amethyst.provo.novell.com:8443/rest/files/
09c1c3fb530f562401531018f4270000?last_version=3" \
   -X POST -H "Content-Type: application/octet-stream" --upload-file ./test.txt

[Response]
{
  "id":"09c1c3fb530f562401531018f4270000",
  "creation":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
23T21:46:23Z"},
  "modification":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
23T22:17:24Z"},
  "length":43,
  "md5":"99a3b991e1b7e7ddfd6be1251402fd1f",
  "name":"aaa.txt",
  "href":"/files/09c1c3fb530f562401531018f4270000/metadata",
  "links":[...],
  "doc_type": "file",
  "version_number": 4,
  "owning_entity": {"id":61,"type":"folderEntry","href":"/folder_entries/61"},
  "parent_binder": {"id":48,"href":"/binders/48"}
}
```

This request succeeds because the file version at the time of the upload matches the version in "last_version". If user B also downloads and edits version 3 of the file, the upload will fail with and HTTP 409 (Conflict) status code because user A already modified it and the version number doesn't match the "last_version" query parameter. The response body is an ErrorInfo object with a "FILE_VERSION_CONFLICT" code and the latest file metadata:

```
[Request]
>  curl -k -u dlewis:novell \
   "https://amethyst.provo.novell.com:8443/rest/files/
09c1c3fb530f562401531018f4270000?last_version=3" \
   -X POST -H "Content-Type: application/octet-stream" --upload-file ./test.txt

[Response]
{
 "code":"FILE_VERSION_CONFLICT",
 "message":"A file with the name already exists.",
 "data":{
   "id":"09c1c3fb530f562401531018f4270000",
   "creation":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
23T21:46:23Z"},
   "modification":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
23T22:17:24Z"},
   "length":43,
   "md5":"99a3b991e1b7e7ddfd6be1251402fd1f",
   "name":"aaa.txt",
   "href":"/files/09c1c3fb530f562401531018f4270000/metadata",
   "links":[...],
   "doc_type": "file",
   "version_number":4,
   "owning_entity":{"id":61,"type":"folderEntry","href":"/folder_entries/61"},
   "parent_binder":{"id":48,"href":"/binders/48"}
 }
}
```

3.3.4 Rename file

To rename a file, use the "name" related link of the FileProperties object (href: "/files/{id}/name"). The Content-Type must be "application/x-www-form-urlencoded" and the new name of the file is specified by the "name" form parameter. The REST interface returns the full FileProperties resource in the response.

```
[Request]
> curl -k -u dlewis:novell "https://amethyst.provo.novell.com:8443/rest/files/
09c1c3fb530f562401531018f4270000/name"
  -X POST -H "Content-Type: application/x-www-form-urlencoded" -d
"name=aaa%20renamed.txt"

[Response]
{
  "id":"09c1c3fb530f562401531018f4270000",
  "creation":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
23T21:46:23Z"},
  "modification":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
23T22:17:24Z"},
  "length":43,
  "md5":"99a3b991e1b7e7ddfd6be1251402fd1f",
  "name":"aaa renamed.txt",
  "href":"/files/09c1c3fb530f562401531018f4270000/metadata",
  "links":[...],
  "doc_type": "file",
  "version_number":4,
  "owning_entity":{"id":61,"type":"folderEntry","href":"/folder_entries/61"},
  "parent_binder":{"id":48,"href":"/binders/48"}
}
```

The name in the POST form data should be a UTF-8 string that has been URL encoded.

3.3.5 Move file

To move a file, use the "parent_folder" related link of the FileProperties object (href: "/files/{id}/parent_folder"). The Content-Type must be "application/x-www-form-urlencoded" and the parent folder to which the file should be moved is specified by the "folder_id" form parameter. The REST interface returns the full FileProperties resource in the response.

```
[Request]
> curl -k -u dlewis:novell \
  "https://amethyst.provo.novell.com:8443/rest/files/
09c1c3fb530f562401531018f4270000/parent_folder" \
  -X POST -H "Content-Type: application/x-www-form-urlencoded" -d "folder_id=49"

[Response]
{
  "id":"09c1c3fb530f56240153105273a40009",
  "creation":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
23T21:46:23Z"},
  "modification":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
23T22:17:24Z"},
  "length":43,
  "md5":"99a3b991e1b7e7ddfd6be1251402fd1f",
  "name":"aaa renamed.txt",
  "href":"/files/09c1c3fb530f56240153105273a40009/metadata",
  "links":[...],
  "doc_type": "file",
  "version_number":4,
  "owning_entity":{"id":62,"type":"folderEntry","href":"/folder_entries/62"},
  "parent_binder":{"id":49,"href":"/binders/49"}
}
```

It's important to note that the file ID and the owning_entity ID have changed in the above response. In certain cases, for example when moving a file from personal storage to a net folder or moving a file between net folders, the move operation is implemented as a copy operation and a delete operation. For this reason, the file and it's associated folder entry are assigned new IDs.

Optionally, you can specify a new name for the file as well with the "name" form parameter:

```
[Request]
> curl -k -u dlewis:novell \
  "https://amethyst.provo.novell.com:8443/rest/files/
09c1c3fb530f562401531018f4270000/parent_folder" \
  -X POST -H "Content-Type: application/x-www-form-urlencoded" -d
"folder_id=48&name=aaa.txt"

[Response]
...
```

The name in the POST form data should be a UTF-8 string that has been URL encoded.

3.3.6 Copy file

The "child_library_files" related link of a Folder can be used to copy a file into that folder. The Content-Type must be "application/x-www-form-urlencoded" and the source file which is to be copied into the folder is specified by the "source_id" form parameter. The name of the new file is specified by the "file_name" form parameter. The REST interface returns the FileProperties representation of the new file in the response.

For example, the "child_library_files" related link of the My Files top level folder is "/self/my_files/library_files". The following copes a file into that folder:

```
[Request]
> curl -k -u dlewis:novell "https://amethyst.provo.novell.com:8443/rest/self/
my_files/library_files" \
  -X POST -H "Content-Type: application/x-www-form-urlencoded" \
  -d "source_id=09c1c3fb530f56240153105273a40009&file_name=aaa%20copied.txt"

[Response]
{
  "id":"09c1c3fb530f562401531070137b000e",
  "creation":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
23T23:21:33Z"},
  "modification":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
23T22:18:18Z"},
  "length":43,
  "md5":"99a3b991e1b7e7ddfd6be1251402fd1f",
  "name":"aaa copied.txt",
  "href":"/files/09c1c3fb530f562401531070137b000e/metadata",
  "links":[...],
  "doc_type": "file",
  "version_number":1,
  "owning_entity":{"id":63,"type":"folderEntry","href":"/folder_entries/63"},
  "parent_binder":{"id":-100,"href":"/self/my_files"}
}
```

3.3.7 Delete file

To delete a file, send DELETE request to the "content" related link of the FilrProperites resource:

```
[Request]
> curl -v -k -u dlewis:novell \
  https://amethyst.provo.novell.com:8443/rest/files/
09c1c3fb530f562401531070137b000e -X DELETE

[Response]
< HTTP/1.1 204 No Content
```

If successful, the REST interface returns an HTTP 204 (No Content) status code with no response body.

3.4 Folders

3.4.1 Create folder

The "child_library_folders" link of the parent folder is used to create a new folder. For example, take the following folder listing:

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/self/
my_files/library_children

[Response]
{
  "first": 0,
  "count": 24,
  "total": 24,
  "items": [{
    "id": 48,
    "doc_type": "binder",
    "entity_type": "folder",
    "href": "/folders/48",
    "title": "A",
    "links":[{
      "rel": "child_library_folders",
      "href": "/folders/48/library_folders"
    },...],
    ...
  },...
}
```

The "child_library_folders" location for this folder is "/folders/48/library_folders". To create a subfolder, POST a Folder object to that resource. The only required field in the Folder object is the title:

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/folders/48/
library_folders \
  -X POST -H "Content-Type: application/json" -d '{"title": "AA"}'

[Response]
{
  "id":82,
  "title":"AA",
  "creation":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
24T22:30:10Z"},
  "modification":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
24T22:30:10Z"},
  "path":"/Home Workspace/Personal Workspaces/David Lewis (dlewis)/Home/A/AA",
  "library":true,
  "mirrored":true,
  "href":"/folders/82",
  "links":[...],
  "doc_type":"binder",
  "parent_binder":{"id":48,"href":"/binders/48"},
  "entity_type":"folder"
}
```

The response is the newly created Folder object.

If a subfolder with that name already exists, the server will return an HTTP 409 Conflict status code. The response body will be an ErrorInfo object with a "TITLE_EXISTS" code and the existing Folder resource:

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/folders/48/
library_folders \
  -X POST -H "Content-Type: application/json" -d '{"title": "AA"}'

[Response]
{
  "code": "TITLE_EXISTS",
  "message: "Title already exists: AA",
  "data": {
    "id":82,
    "title":"AA",
    "creation":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
24T22:30:10Z"},
    "modification":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
24T22:30:10Z"},
    "path":"/Home Workspace/Personal Workspaces/David Lewis (dlewis)/Home/A/AA",
    "library":true,
    "mirrored":true,
    "href":"/folders/82",
    "links":[...],
    "doc_type":"binder",
    "parent_binder":{"id":48,"href":"/binders/48"},
    "entity_type":"folder"
  }
}
```

3.4.2 Rename folder

To rename a folder, use the "title" related link of the Binder object (href: "/folders/{id}/title"). The Content-Type must be "application/x-www-form-urlencoded" and the new name of the folder is specified by the "title" form parameter. The REST interface returns the full Binder resource in the response.

```
[Request]
> curl -k -u dlewis:novell "https://amethyst.provo.novell.com:8443/rest/folders/
82/title"
  -X POST -H "Content-Type: application/x-www-form-urlencoded" -d
"title=AA%20renamed"

[Response]
{
  "id":82,
  "title":"AA renamed",
  "creation":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
24T22:30:10Z"},
  "modification":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
24T22:46:49Z"},
  "path":"/Home Workspace/Personal Workspaces/David Lewis (dlewis)/Home/A/AA
renamed",
  "library":true,
  "mirrored":true,
  "href":"/folders/82",
  "links":[...],
  "doc_type":"binder",
  "parent_binder":{"id":48,"href":"/binders/48"},
  "entity_type":"folder"
}
```

The title in the POST form data should be a UTF-8 string that has been URL encoded.

3.4.3 Move folder

To move a folder, use the "parent_binder" related link of the Binder object (href: "/folders/{id}/parent_binder"). The Content-Type must be "application/x-www-form-urlencoded" and the target folder is specified by the "binder_id" form parameter. The REST interface returns the full Binder resource in the response.

```
[Request]
> curl -k -u dlewis:novell "https://amethyst.provo.novell.com:8443/rest/folders/
82/parent_binder"
  -X POST -H "Content-Type: application/x-www-form-urlencoded" -d "binder_id=49"

[Response]
{
  "id":82,
  "title":"AA renamed",
  "creation":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
24T22:30:10Z"},
  "modification":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
24T22:46:49Z"},
  "path":"/Home Workspace/Personal Workspaces/David Lewis (dlewis)/Home/A/AA
renamed",
  "library":true,
  "mirrored":true,
  "href":"/folders/82",
  "links":[...],
  "doc_type":"binder",
  "parent_binder":{"id":49,"href":"/binders/49"},
  "entity_type":"folder"
}
```

3.4.4 Delete folder

To delete a folder, send DELETE request to the Folder resource's href location:

```
[Request]
> curl -v -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/folders/
82 -X DELETE

[Response]
< HTTP/1.1 204 No Content
```

If successful, the REST interface returns an HTTP 204 (No Content) status code with no response body.

For personal storage folders, the default behavior is to move the folder to the trash. To delete the folder permanently, use the "purge=true" query parameter:

```
[Request]
> curl -v -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/folders/
82?purge=true -X DELETE

[Response]
< HTTP/1.1 204 No Content
```

Trash is not supported for external storage folders (mirrored/net folder). They are always deleted permanently.

By default, this will delete the folder and its children. To only delete the folder if it is empty, use the "only_if_empty=true" query parameter. If the folder is not empty, the server will return an HTTP 409 Conflict status code. The response body will be an ErrorInfo object with a "BINDER_NOT_EMPTY" code.

```
[Request]
> curl -v -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/folders/
82?only_if_empty=true -X DELETE

[Response]
< HTTP/1.1 409 Conflict
{"code":"BINDER_NOT_EMPTY","message":"The binder is not empty."}
```

3.4.5 Copy folder

The "child_library_folders" related link of a Folder can be used to copy another into that folder. The Content-Type must be "application/x-www-form-urlencoded" and the source folder to be copied is specified by the "source_id" form parameter. The name of the new folder is specified by the "title" form parameter. The REST interface returns the Folder representation of the new folder in the response.

For example, the "child_library_folders" related link of the My Files top level folder is "/self/my_files/library_folders". The following copies a folder there:

```
[Request]
> curl -k -u dlewis:novell "https://amethyst.provo.novell.com:8443/rest/self/
my_files/library_folders" \
  -X POST -H "Content-Type: application/x-www-form-urlencoded" \
  -d "source_id=49&title=B%20copied"

[Response]
{
  "id":83,
  "title":"B copied",
  "description":{"text":"","format":2,"format_str":"text"},
  "family":"file",
  "creation":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
24T23:15:33Z"},
  "modification":{"principal":{"id":20,"href":"/users/20"},"date":"2016-02-
24T23:15:33Z"},
  "attachments":[],
  "path":"/Home Workspace/Personal Workspaces/David Lewis (dlewis)/Home/B copied",
  "library":true,
  "mirrored":true,
  "href":"/folders/83",
  "links":[...],
  "doc_type":"binder",
  "parent_binder":{"id":44,"href":"/binders/44"},
  "entity_type":"folder"
}
```

3.5 Sharing

3.5.1 Resource Representations

Share

```
{
  "id": long,
  "href": string,
  "shared_entity": EntityId Resource,
  "sharer": LongIdLinkPair Resource,
  "recipient": ShareRecipient Resource,
  "days_to_expire": long,
  "expiration": iso_8601_date,
  "comment": string,
  "access": Access Resource
}
```

ShareRecipient

```
{
  "type": string ("user", "group", "external_user", "public", "public_link"),
  "id": long,
  "email": string
}
```

3.5.2 List shares for file or folder

Each file and folder has a "shares" related link that can be used to get sharing information by the authenticated user. For folders, the href is "/folders/{id}/shares" and for files, it is "/folder_entries/{owning_folder_entry_id}/shares".

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/
folder_entries/15/shares

[Response]
{
  "first":0,
  "count":1,
  "total":1,
  "items":[{
    "id":8,
    "href":"/shares/8",
    "comment":"",
    "shared_entity":{"id":15,"type":"folderEntry","href":"/folder_entries/15"},
    "sharer":{"id":20,"href":"/users/20"},
```

```
    "recipient":{"id":34,"type":"user","href":"/users/34"},
    "sharing_date":"2016-02-29T16:16:31Z",
    "access":{
      "role":"EDITOR",
      "sharing":{
        "internal":false,
        "external":false,
        "public":false,
        "public_link":false,
        "grant_reshare":false
      }
    }
  }]
}
```

3.5.3 Sharing permissions and system settings

Each file and folder has an "access" related link that can be used to determine whether the authenticated user can share the file or folder. For folders, the href is "/folders/{id}/access" and for files, it is "/folder_entries/{owning_folder_entry_id}/access".

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/
folder_entries/15/access

[Response]
{
  "role":"EDITOR",
  "sharing":{
    "internal":true,
    "external":true,
    "public":true,
    "public_link":true,
    "grant_reshare":true
  }
}
```

The Access object above indicates that the user is allowed to share the file with internal users, external users and the public. The user is also allowed to share the file's public link and grant permission to others to reshare the file.

There are also system settings that affect who the user can share with. These settings can be found in the zone config (href: "/zone_config"):

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/zone_config

[Response]
{
  ...
  "allow_sharing_with_ldap_groups":true,

"external_sharing_restrictions":{"mode":"none","email_list":[],"domain_list":[]},
  ...
}
```

These values indicate that the user is allowed to share with LDAP groups and that there are no restrictions in place for sharing with external users. The other possible external sharing restriction modes are "whitelist" and "blacklist". "whitelist" means that the "email_list" and "domain_list"

properties contain a list of allowed email addresses and allowed email domains. Users cannot share with any other email addresses. "blacklist" means that the "email_list" and "domain_list" properties contain a list of disallowed email addresses and disallowed email domains. Users may share with all other email addresses.

3.5.4 Share

To share a file or folder with a user or group, POST a Share object to the file or folder's "shares" related link. At a minimum, you must specify the recipient and the access role ("VIEWER", "EDITOR" or "CONTRIBUTOR"):

```
[Request]
> curl -k -u dlewis:novell \
  https://amethyst.provo.novell.com:8443/rest/folder_entries/15/shares \
  -H "Content-Type: application/json" -X POST \
  -d '{"recipient":{"type":"user","id":32},"access":{"role":"EDITOR"}}'

[Response]
{
  "id":9,
  "href":"/shares/9",
  "shared_entity":{"id":15,"type":"folderEntry","href":"/folder_entries/15"},
  "sharer":{"id":20,"href":"/users/20"},
  "recipient":{"id":32,"type":"user","href":"/users/32"},
  "sharing_date":"2016-02-29T23:20:06Z",
  "access":{
    "role":"EDITOR",
    "sharing":{
      "internal":false,
      "external":false,
      "public":false,
      "public_link":false,
      "grant_reshare":false
    }
  },
}
```

You can grant the recipient permission to reshare the item (assuming you have permission to grant reshare) by including the desired sharing permissions in the Access property.

Expiration

You can set an expiration time on the share in one of two ways. You can specify the "days_to_expire" property in the share that you post, or you can set the exact expiration date and time in the "expiration" property. For example:

```
[Request]
> curl -k -u dlewis:novell \
  https://amethyst.provo.novell.com:8443/rest/folder_entries/15/shares \
  -H "Content-Type: application/json" -X POST \
  -d
'{"recipient":{"type":"user","id":32},"access":{"role":"EDITOR"},"days_to_expire":
10}'

[Response]
{
  "id":11,
  "href":"/shares/11",
  "shared_entity":{"id":15,"type":"folderEntry","href":"/folder_entries/15"},
  "sharer":{"id":20,"href":"/users/20"},
  "recipient":{"id":32,"type":"user","href":"/users/32"},
  "sharing_date":"2016-02-29T23:28:52Z",
  "days_to_expire":10,
  "expiration":"2016-03-10T23:28:52Z"
  "access":{
    "role":"EDITOR",
    "sharing":{
      "internal":false,
      "external":false,
      "public":false,
      "public_link":false,
      "grant_reshare":false
    }
  },
}
```

Notice that when you specify the "days_to_expire", both the "days_to_expire" and "expiration" properties are set in the response. The expiration time is calculated based on the supplied "days_to_expire" value and the "days_to_expire" value is no longer relevant. It is preserved in the Share object for informational purposes only.

Notify Recipient

To notify the recipient when creating the share, specify the "notify=true" query parameter:

```
[Request]
> curl -k -u dlewis:novell \
  https://amethyst.provo.novell.com:8443/rest/folder_entries/15/shares?notify=true
\
  -H "Content-Type: application/json" -X POST \
  -d '{"recipient":{"type":"user","id":32},"access":{"role":"EDITOR"}}'

[Response]
(Same as above, and the server will send an email to the recipient.)
```

External users

Sharing with an external user is the same as sharing with internal users, except the recipient is a little different. The recipient type is "external_user" and the external user's email address is specified.

```
[Request]
> curl -k -u dlewis:novell \
  https://amethyst.provo.novell.com:8443/rest/folder_entries/15/shares?notify=true
\
  -H "Content-Type: application/json" -X POST \
  -d
'{"recipient":{"type":"external_user","email":"user@fakedomain.com"},"access":{"ro
le":"EDITOR"}}'

[Response]

{
  "id":12,
  "href":"/shares/9",
  "shared_entity":{"id":15,"type":"folderEntry","href":"/folder_entries/15"},
  "sharer":{"id":20,"href":"/users/20"},
  "recipient":{"id":141,"type":"external_user","href":"/users/
141","email":"user@fakedomain.com"},
  "sharing_date":"2016-02-29T23:20:06Z",
  "access":{...}
}
```

The attempt to share with the external user will fail if the user does not have permission to share the file or folder with external users, or if the email address is not allowed based on the system external sharing restriction settings.

Share with public

Sharing with the public is the same as sharing with internal users, except the recipient type is "public". When sharing publicly, the access role must be "VIEWER".

```
[Request]
> curl -k -u dlewis:novell \
  https://amethyst.provo.novell.com:8443/rest/folder_entries/15/shares?notify=true
\
  -H "Content-Type: application/json" -X POST \
  -d '{"recipient":{"type":"public"},"access":{"role":"VIEWER"}}'

[Response]
{
  "id":13,
  "href":"/shares/9",
  "shared_entity":{"id":15,"type":"folderEntry","href":"/folder_entries/15"},
  "sharer":{"id":20,"href":"/users/20"},
  "recipient":{"type":"public"},
  "sharing_date":"2016-02-29T23:20:06Z",
  "access":{...}
}
```

Create public link

To create a public link for a file, the recipient type should be "public_link". You do not need to specify the access in this case. Only "VIEWER" with no sharing permissions is allowed.

The public URLs that can be used by anyone to access the file are returned in the "permalinks" property. The "download" URL is a direct link to download the file. The "view" URL goes to the file details page in the web application.

```
[Request]
> curl -k -u dlewis:novell \
  https://amethyst.provo.novell.com:8443/rest/folder_entries/15/shares?notify=true
\
  -H "Content-Type: application/json" -X POST \
  -d '{"recipient":{"type":"public_link"}}'

[Response]
{
  "id":13,
  "href":"/shares/9",
  "shared_entity":{"id":15,"type":"folderEntry","href":"/folder_entries/15"},
  "sharer":{"id":20,"href":"/users/20"},
  "recipient":{"type":"public"},
  "sharing_date":"2016-02-29T23:20:06Z",
  "access":{...},
  "permalinks":[{
    "rel":"download",
    "href":"https://amethyst.provo.novell.com:8443/ssf/s/readFile/share/15/-
6714533406876790678/publicLink/test.txt"
  },{
    "rel":"view",
    "href":"https://amethyst.provo.novell.com:8443/ssf/s/readFile/share/15/-
6714533406876790678/publicLinkHtml/test.txt"
  }]
}
```

3.5.5 Modify share

To modify a share, POST the update share object to the share's href value ("/shares/{id}"). You are only allowed to change the share's comment, access and expiration. Modifying the recipient or shared file or folder is not supported.

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/shares/13 \
  -H "Content-Type: application/json" -X POST \
  -d '{"days_to_expire":30,"access":{"role":"VIEWER"}}'

[Response]
{
  "id":17,
  "href":"/shares/17",
  "sharer":{"id":20,"href":"/users/20"},
  "shared_entity":{"id":15,"type":"folderEntry","href":"/folder_entries/15"},
  "recipient":{"type":"public"},
  "sharing_date":"2016-03-01T00:03:20Z",
  "days_to_expire":30,
  "expiration":"2016-03-31T00:03:22Z",
  "access":{
    "role":"VIEWER",
    "sharing":{...}
  }
}
```

Note: Modifying a share changes its ID. The new ID is returned in the response.

The recipient of the share will notified by email if the "notify=true" query parameter is specified.

You can also modify a share by creating a new share. When creating a new share, if the recipient, sharer and shared_entity match an existing share, that share is replaced with the new share.

3.5.6 Delete share

To delete a share, send a DELETE request to the share's href value ("/shares/{id}").

```
[Request]
> curl -v -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/shares/
17 -X DELETE

[Response]
< HTTP/1.1 204 No Content
```

3.6 Get global settings (zone_config)

The zone config resource contains system settings defined by the administrator for the authenticated user.

3.6.1 Resource Representations

DiskQuotasConfig

```
{
  "enabled": boolean,
  "user_default": integer,
  "highwater_percentage": integer
}
```

ExternalSharingRestrictions

```
{
   "mode": string ("none", "whitelist" or "blacklist"),
   "email_list": [ string ],
   "domain_list": [ string ]
}
```

ZoneConfig

```
{
  "id": long,
  "guid": string,
  "disk_quotas": DiskQuotasConfig Resource,
  "desktop_app_config": DesktopAppConfig Resource,
  "mobile_app_config": MobileAppConfig Resource,
  "file_upload_size_limit: long,
  "allow_sharing_with_ldap_groups": boolean,
  "external_sharing_restrictions": ExternalSharingRestrictions Resource
}
```

3.6.2 Get Settings

To get the user's system settings, use the zone_config related link (href: "/zone_config") from the root REST resource:

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/zone_config

[Response]
{
  "id":1,
  "guid":"09c1c3fb52569424015256945e76000f",
  "binder_quotas":{
    "initialized":false,
    "enabled":false,
    "allow_owner":false
  },
  "disk_quotas":{
    "enabled":false,
    "user_default":100,
    "highwater_percentage":90
  },
  "desktop_app_config":{
    "enabled":true,
    "allow_cached_password":true,
    "sync_interval":15,
    "auto_update_url":"",
    "max_file_size":52428800
  },
  "mobile_app_config":{
    "enabled":true,
    "allow_cached_password":true,
    "allow_cached_content":true,
    "allow_play_with_other_apps":true,
    "force_pin_code":false,
    "sync_interval":15,
    "allow_cut_copy":true,
    "allow_screen_capture":true,
    "allow_rooted_devices":false,
    "allowed_open_in_apps":"all"
  },
  "allow_sharing_with_ldap_groups":true,
  "external_sharing_restrictions":{
    "mode":"none",
    "email_list":[],
    "domain_list":[]
  }
}
```

3.7 Get authenticated user information (self)

To get the User resource for the authenticated user, use the self related link (href: "/self") of the root REST resource:

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/self

[Response]
{
  "id":20,
  "title":"David Lewis",
  "name":"dlewis",
  "description":{"text":"","format":2,"format_str":"text"},
  "creation":{
    "principal":{"id":20,"href":"/users/20"},
    "date":"2016-01-18T21:14:18Z"
  },
  "modification":{
    "principal":{"id":20,"href":"/users/20"},
    "date":"2016-01-18T22:43:25Z"
  },
  "disabled":false,
  "reserved":false,
  "person":true,
  "organization":"",
  "phone":"",
  "locale":{"country":"US","language":"en"},
  "workspace":{"id":43,"href":"/workspaces/43"},
  "href":"/self",
  "links":[{
    "rel":"shares_with",
    "href":"/shares/with_user/20"
  },{
    "rel":"shares_by",
    "href":"/shares/by_user/20"
  },{
    "rel":"password",
    "href":"/users/20/password"
  },{
    "rel":"roots",
    "href":"/self/roots"
  },{
    "rel":"my_files",
    "href":"/self/my_files"
  },{
    "rel":"net_folders",
    "href":"/self/net_folders"
  },{
    "rel":"shared_with_me",
    "href":"/self/shared_with_me"
  },{
    "rel":"shared_by_me",
    "href":"/self/shared_by_me"
  },{
  "rel":"public_shares",
  "href":"/self/public_shares"
```

```
  }],
  "permalink":"...",
  "email":"dlewis@novell.com",
  "first_name":"David",
  "middle_name":"",
  "last_name":"Lewis",
  "time_zone":"America/Boise",
  "disk_quota":0,
  "disk_space_used":0,
  "disk_space_quota":104857600,
  "hidden_files_folder":{"id":78,"href":"/folders/78"},
  "mobile_app_config":{...},
  "desktop_app_config":{...}
}
```

3.8 Browse

3.8.1 List top level folders

The /self User resource has a related link named <code>roots</code> (href: /self/roots) that can be used to retrieve the top level folders that are available to the authenticated user. The response is a SearchResultsList of BinderBrief objects.

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/self/roots

[Response]
{
  "first":0,
  "count":5,
  "total":5,
  "items":[{
    "id": -100,
    "title": "My Files",
    "path": "/Home Workspace/Personal Workspaces/David Lewis (dlewis)/Home",
    "href": "/self/my_files",
    "doc_type": "binder",
    "links": [...]
  },{
    "id": -101,
    "title": "Shared with Me",
    "href": "/self/shared_with_me",
    "doc_type": "binder",
    "links": [...]
  },{
    "id": -102,
    "title": "Shared by Me",
    "href": "/self/shared_by_me",
    "doc_type": "binder",
    "links": [...]
```

```
},{
  "id": -103,
  "title": "Net Folders",
  "path": "/Home Workspace/Net Folders",
  "href": "/self/net_folders",
  "doc_type": "binder",
  "links": [...]
},{
  "id": -104,
  "title": "Public",
  "href": "/self/public",
  "doc_type": "binder",
  "links": [...]
}]
```

Filr has five top level folders:

- My Files (/self/my_files)
- Shared with Me (/self/shared_with_me)
- Shared by Me (/self/shared_by_me)
- Net Folders (/self/net_folders)
- Public (/self/public)

However, not every top level folder is available to every user. The /self/roots response will only include those that are available.

Each of the top level folders have an <code>href</code> that can be used to retrieve the BinderBrief object individually. The /self User resource also has related links for each of the top level folders that are available to the user.

The titles of the top level folders are not localized in the response. The English name of the folder is always returned. Applications that are displaying the top level folders to users should localize the titles if the English name is not acceptable.

3.8.2 List folder contents (library_children)

- “Paged Results” on page 75
- “Listing multiple binders at once” on page 76
- “Last-Modified/If-Modified-Since” on page 78

Every Binder and BinderBrief resource has three related links that can be used to list the children of the binder:

- library_children: retrieve all children of the binder
- child_library_folders: retrieve the child folders of the binder
- child_library_files: retrieve the child files of the binder

For example, the BinderBrief resource representing the "My Files" top level folder has the following related links:

```
{
  "id": -100,
  "title": "My Files",
  "links": [{
    "rel": "library_children",
    "href": "/self/my_files/library_children"
  },{
    "rel": "child_library_folders",
    "href": "/self/my_files/library_folders"
  },{
    "rel": "child_library_files",
    "href": "/self/my_files/library_files"
  }]
}
```

So, to list all of the children of the My Files folder, you find the "library_children" related link (href: "/self/my_files/library_children") and make a GET request for that resource. The response is a SearchResultList of SearchableObjects (BinderBrief objects to represent the folders and FileProperties objects for files).

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/self/
my_files/library_children

[Response]
{
  "first":0,
  "count":24,
  "total":24,
  "items":[{
    "id": 48,
    "doc_type": "binder",
    "entity_type": "folder",
    "href": "/folders/48",
    "parent_binder": {
      "href": "/self/my_files",
      "id": -100
    },
    "title": "A",
    "path": "/Home Workspaces/Personal Workspaces/David Lewis (dlewis)/Home/A",
    "links": [{
      "rel": "library_children",
      "href": "/folders/48/library_children"
    },{
      "rel": "child_library_folders",
      "href": "/folders/48/library_folders"
    },{
      "rel": "child_library_files",
      "href": "/folders/48/library_files"
```

```
    }]
  },{
    "id": "09c1c3fb5256e2e4015256e8691c003b",
    "doc_type": "file",
    "href": "/files/09c1c3fb5256e2e4015256e8691c003b/metadata",
    "parent_binder": {
      "href": "/self/my_files",
      "id": -100
    },
    "name": "test.txt",
  },
  ...22 more results...
  ]
}
```

Notice that the BinderBrief resource representing the child folder named A also has "library_children", "child_library_folders" and "child_library_files" related links. Those links can then be used to list the children of folder A:

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/folders/48/
library_children

[Response]
{
  "first": 0,
  "count": 12,
  "total": 12,
  "items": [...]
}
```

Paged Results

Most of the library_children resources only return the first 100 results. The SearchResultList response indicates how many children were returned (the "count" field) and the total number of children in the folder (the "total" field). If a partial list is returned, the SearchResultList will also include a "next" field containing an Href that can be used to retrieve the next page of children. This is done using "first" and "count" query parameters.

```
[Request]
> curl -k -u dlewis:novell \
  https://amethyst.provo.novell.com:8443/rest/self/my_files/
library_children?count=20

[Response]
{
  "first":0,
  "count":20,
  "total":24,
  "next":"/self/my_files/library_children?first=20&count=20",
  "items":[...First 20 results...]
}

[Request]
> curl -k -u dlewis:novell \
  "https://amethyst.provo.novell.com:8443/rest/self/my_files/
library_children?first=20&count=2"

[Response]
{
  "first":20,
  "count":4,
  "total":24,
  "items":[...Last 4 results...]
}
```

The /shares/*/library_children, /shares/*/library_folders and /shares/*/library_files resources do not support the "first" and "count" query parameters. All results are returned regardless of the values specified by "first" and "count".

Listing multiple binders at once

The root resource has a "binder_library_children" related link that can be used to list the children of multiple binders in a single request:

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest

[Response]
{
  "links":[
    ...
    {
      "rel": "binder_library_children",
      "href": "/binders/library_children",
    },
    ...
}
```

The "binder_library_children" resource allows you to specific a number of binder IDs using the "id" query parameter. It supports top level folder IDs (-100, -101, etc.) as well as regular folder IDs (48, 103, 567). The response is a list of BinderChildren objects.

```
[Request]
> curl -k -u dlewis:novell \
  "https://amethyst.provo.novell.com:8443/rest/binders/library_children?id=-
100&id=48&id=49&count=10"

[Response]
[{
  "binder_id": 49,
  "children": {
    "first": 0,
    "count": 2,
    "total": 2,
    "items": [...2 children...]
  }
},{
  "binder_id": 48,
  "children": {
    "first": 0,
    "count": 8,
    "total": 12,
    "items": [...8 children...]
}]
```

The "count" query parameter specifies the total number of children to return. In this case, 10 results were returned: both children of folder 49 and the first 8 children of folder 48. The following illustrates how to retrieve the next 10 results:

```
[Request]
> curl -k -u dlewis:novell \
  "https://amethyst.provo.novell.com:8443/rest/binders/library_children?\
   id=-100&id=48&count=10&first_id=48&first=8"

[Response]
[{
  "binder_id": 48,
  "children": {
    "first": 8,
    "count": 4,
    "total": 12,
    "items": [...4 children...]
  }
},{
  "binder_id": -100,
  "children": {
    "first": 0,
    "count": 6,
    "total": 24,
    "items": [...6 children...]
}]
```

Folder 49 was removed from the request because the previous request returned all of the results. Since the order of the IDs is non-deterministic, the "first_id=48" parameter indicates that folder 48 should be processed first, and the "first=8" parameter specifies that the results should begin with the 9th child of folder 49 (the indexing is 0-based). The response contains the final 4 results of folder 48 and the first 6 results of folder -100.

To get the next 10 results, the query parameters would be "id=-100&first_id=-100&first=6&count=10".

Last-Modified/If-Modified-Since

The "library_children", "library_folders" and "library_files" resources support the "If-Modified_Since" HTTP header. The -v curl option displays verbose information, including request and response headers.

```
[Request]
> curl -v -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/folders/
48/library_children

[Response]
> GET /rest/folders/48/library_children HTTP/1.1
> Authorization: Basic ****
> User-Agent: curl/7.35.0
> Host: amethyst.provo.novell.com:8443
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: Apache-Coyote/1.1
< Last-Modified: Thu, 11 Feb 2016 17:46:28 GMT
< Content-Type: application/json
< Transfer-Encoding: chunked
< Date: Fri, 19 Feb 2016 05:45:02 GMT
<
{
  "first": 0,
  "count": 12,
  "total": 12,
  "items": [...]
}
```

The "Last-Modified" response header indicates the time when the list of children last changed. If you make that same request, but include this time in the "If-Modified-Since" request header, the server will return an HTTP 304 Not Modified response with no response body.

```
[Request]
> curl -v -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/folders/
48/library_children \
  -H "If-Modified-Since: Thu, 11 Feb 2016 17:46:28 GMT"

[Response]
> GET /rest/folders/48/library_children HTTP/1.1
> Authorization: Basic ****
> User-Agent: curl/7.35.0
> Host: amethyst.provo.novell.com:8443
> Accept: */*
> If-Modified-Since: Thu, 11 Feb 2016 17:46:28 GMT
>
< HTTP/1.1 304 Not Modified
< Server: Apache-Coyote/1.1
< Date: Fri, 19 Feb 2016 05:48:21 GMT
<
```

If the list of children has changes after the date in the "If-Modified-Since" header, the response will contain the full list of children, or the full page of children as specified by the "first" and "count" query parameters.

```
[Request]
> curl -v -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/folders/
48/library_children
  -H "If-Modified-Since: Thu, 11 Feb 2016 17:46:28 GMT"

[Response]
> GET /rest/folders/48/library_children HTTP/1.1
> Authorization: Basic ****
> User-Agent: curl/7.35.0
> Host: amethyst.provo.novell.com:8443
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: Apache-Coyote/1.1
< Last-Modified: Thu, 18 Feb 2016 14:23:54 GMT
< Content-Type: application/json
< Transfer-Encoding: chunked
< Date: Fri, 19 Feb 2016 05:45:02 GMT
<
{
  "first": 0,
  "count": 13,
  "total": 13,
  "items": [...all 13 children...]
}
```

3.9 What's New (recent_activity)

The "recent_activity" related link can be used to get folder entries that have changed recently. To get site-wide recent activity, use the root rest object's "recent_activity" link.

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest

[Response]
{
  "links":[...,{
    "rel":"recent_activity",
    "href":"/workspaces/1/recent_activity"
  },...]
}
```

The response contains a SearchResultList of RecentActivityEntry objects. A RecentActivityEntry is a FolderEntry with the 2 most recent replies to that folder entry included.

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/workspaces/
1/recent_activity?count=5

[Response]
{
  "first":0,
  "count":5,
  "total":53,
  "items":[{
    "id":15,
    "entity_type":"folderEntry",
    "href":"/folder_entries/15",
    "title":"test.txt",
```

```
    "parent_binder":{"id":57,"href":"/binders/57"},
    ...,
    "primary_file":{
      "id","09c1c3fb5256e2e4015256e8691c003b",
      "href":"/files/09c1c3fb5256e2e4015256e8691c003b/metadata",
      "name":"test.txt",
      "length":14,
      "mod_date":"2014-04-24T17:05:40Z"
    },
    "replies":[{
      "id":106,
      "href":"/replies/106",
      "title":"Re: test.txt",
      "description":{
        "text":"I made a few more changes",
        "format_str":"text"
      },
      ...
    },{
      "id":106,
      "href":"/replies/106",
      "title":"Re: test.txt",
      "description":{
        "text":"I made a few more changes",
        "format_str":"text"
      },
      ...
    }]
  },
  ...
  ]
}
```

Each binder object, including the top level folders such as My Files (-100) and Shared with Me (-101), also has a "recent_activity" related link that can be used to get recently changed folder entries in that particular location.

3.10 Searching (library_entities)

The "library_entities" related link can be used to search for files and folders by keyword. To search the entire site, use the root rest object's "library_entities" link.

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest

[Response]
{
  "links":[...,{
    "rel":"library_entities",
    "href":"/workspaces/1/library_entities"
  },...]
}
```

The following query parameters are supported:

- recursive: whether to search a single level (false) or the entire sub-tree (true).
- binders: whether to include binders in the results.
- folder_entries: whether to include folder entries in the results.

- files: whether to include files in the results.
- replies: whether to include replies in the results.
- keyword: search term. May include wildcards (*, ?), but not as the first character in the keyword ("Test*" is allowed but "*Test" is not).

The response is a SearchResultList of BinderBrief and FileProperties objects.

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/workspaces/
1/library_entities?recursive=true&keyword=H*

[Response]
{
  "first":0,
  "count":100,
  "total":514,
  "items":[...]
}
```

Each binder object, including the top level folders such as My Files (-100) and Shared with Me (-101), also has a "library_entities" related link that can be used to search that particular location.

3.11 Users and Groups (principals)

The root REST resource has a "principals" related link that is used to get users and groups.

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest

[Response]
{
  "links":[...,{
    "rel":"principals",
    "href":"/principals"
  },...]
}
```

3.11.1 Get by ID

The principals resource can be used to look up multiple users and groups by ID.

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/
principals?id=20&id=32&id=46

[Response]
{
  "first":0,
  "count":3,
  "total":3,
  "items":[...]
}
```

The response is a SearchResultList of UserBrief and GroupBrief objects.

3.11.2 Search

The principals resource can also be used to search for users and groups by keyword. The keyword query parameter can include wildcards and matches on the full name, email address and login name fields.

```
[Request]
> curl -k -u dlewis:novell https://amethyst.provo.novell.com:8443/rest/
principals?keyword=da*

[Response]
{
  "first":0,
  "count":8,
  "total":8,
  "items":[...]
}
```

The following query parameters are supported:

- keyword: search term. May include wildcards (*, ?), but not as the first character in the keyword ("Test*" is allowed but "*Test" is not).
- included_groups: "none", "all", "local" or "ldap"
- included_users: "none", "all", "local" or "ldap"
- first/count: for paging the results

A Glossary

Term	Definition
Attachment	A file attached to a folder entry. In Filr, a logical file consists of two parts: a folder entry with an attachment.
Binder	A container object. There are two types: folders and workspaces.
Entity	Any kind of first-class object: binder, folder entry, file or reply.
Entry	An abbreviated form of Folder Entry
File	Can either refer specifically to an attachment, or more generally to a folder entry and its attachment.
Folder Entry	The encapsulation of a file and Filr-related extensions (comments, shares). Each logical file in Filr consists of a folder entry with an attachment.
Folder	A type of binder that can contain other folders as well as folder entries (files).
Library	A property of a folder indicating that all files in the folder must have unique names. All folders in Filr are library folders. A library entity or library file is an entity or file that resides in a library folder.
Mirrored	A property of a folder or file that indicates that the folder or file that is stored outside of Filr. Net and home folders are “mirrored”.
Principal	A user or group.
Reply	A comment that a user has made on a file.
Share	An object encapsulating information about the action a user took to share a folder or folder entry.
Top Level Folder	A virtual folder, such as “My Files”, “Shared with Me” or “Net Folders”.
Workspace	A type of binder that can contain other workspaces as well as folders. No workspaces are exposed to the end users in Filr.

B About cURL

cURL is an open source command line tool for making HTTP requests. It typically comes installed on on Linux and Mac computers and can be downloaded for other platforms, including Windows, from the cURL Download page (https://curl.haxx.se/download.html).

The following describes the options that are used throughout the Filr REST tutorial. Full documentation is available on the cURL Documentation page (https://curl.haxx.se/docs/).

The general syntax is: `curl [options] <url>`

Option	Description
`-k`	Do not verify SSL certificates for HTTPS requests
`-u [username]:[password]`	User credentials for basic authentication
`-v`	Verbose; show additional output, including the request and response headers
`-X`	Specify the HTTP command (GET, PUT, POST or DELETE)
`-H "[header-name]: header-value"`	Specify a custom request header
`-d "[DATA]"`	Data to POST in the request body
`-o [file-name]`	Write output to the file instead of stdout
`-T [file-name]`	Upload the specified file in the request body