

Arvados - Feature #10268

[SDK] Convenience method for resolving project/file paths

10/13/2016 06:24 PM - Peter Amstutz

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assigned To:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			

Description

Wrote this up as a code sample for customer, but may want to incorporate it into SDK:

```
import arvados
from arvados.collection import RichCollectionBase
from arvados.arvfile import ArvadosFile

api = arvados.api('v1')

def follow_path(path, parent=None):
    if isinstance(path, basestring):
        path = path.split("/")

    if parent is None:
        scope = path.pop(0)
        current_user = api.users().current().execute()
        if scope == "home":
            parent = current_user["uuid"]
        elif scope == "shared":
            startproject = path.pop(0)
            projects = api.groups().list(filters=[["name", "=", startproject]]).execute()
            if len(projects["items"]) == 1:
                parent = projects["items"][0]["uuid"]
            elif len(projects["items"]) == 0:
                raise Exception("Starting project '%s' not found" % startproject)
            else:
                raise Exception("Found multiple starting projects named '%s'" % (startproject))
        else:
            raise Exception("First path segment must be one of 'home' or 'shared'")

    if len(path) == 0:
        return parent

    # Look for projects with the next name in the path list
    name = path.pop(0)
    l = api.groups().list(filters=[["owner_uuid", "=", parent], ["name", "=", name]].execute()
    if l["items"]:
        return follow_path(path, l["items"][0]["uuid"])

    # Look for collections with the next name in the path list
    l = api.collections().list(filters=[["owner_uuid", "=", parent], ["name", "=", name]].execute()
    if l["items"]:
        cr = arvados.collection.CollectionReader(l["items"][0]["uuid"])
        if path:
            f = cr.find("/".join(path))
            if f:
                return f
            else:
                raise Exception("path '%s' not found in collection %s" % ("/".join(path), cr.manifest_locator()))
        else:
            return cr
```

```
    raise Exception("Can't find subproject or collection named '%s' under project %s" % (name, parent))

#path = "home/Tools/Examples and templates/Config template for example pipeline/example.config"
#path = "home/Tools/Examples and templates/Config template for example pipeline"
#path = "home/Tools/Examples and templates"
#path = "home/Tools"
#path = "shared/New project"
#path = "shared/Examples and templates"

leaf = follow_path(path)

if isinstance(leaf, basestring):
    print "Resolved to project uuid %s" % leaf

if isinstance(leaf, RichCollectionBase):
    print "Resolved to directory '%s' within collection %s" % (leaf.stream_name(), leaf.root_collection().manifest_locator())

if isinstance(leaf, ArvadosFile):
    print "Resolved to file '%s/%s' within collection %s" % (leaf.parent.stream_name(), leaf.name, leaf.parent.root_collection().manifest_locator())
```

History

#1 - 10/13/2016 06:25 PM - Peter Amstutz

- Description updated

#2 - 10/14/2016 12:32 AM - Tom Morris

So this is basically extending Collection.find() to deal with the full path in the way that arv-mount does? What method does arv-mount use?

<http://doc.arvados.org/sdk/python/arvados/arvados.collection.RichCollectionBase-class.html#find>

#3 - 08/29/2017 01:54 PM - Tom Morris

- Target version set to Arvados Future Sprints

#4 - 07/07/2021 06:22 PM - Ward Vandewege

- Target version deleted (Arvados Future Sprints)