Grouper Remedy integration

This integration can be two integrations, to Remedy (SaaS) and to Digital Marketplace which is a helper application for remedy.

This integration is a change log consumer or a message listener for events that should be published to Remedy. In Remedy, groups should be created in remedy, and Grouper, and memberships will be synced over. In Digital marketplace, a group created in Remedy will be created automatically in digital marketplace.

- Grouper Remedy Digital Marketplace integration

Common configuration

Configure logging in log4j.properties

```
log4j.appender.grouperRemedy                            = org.apache.log4j.DailyRollingFileAppender
log4j.appender.grouperRemedy.File                       = logs/grouperRemedy.log
log4j.appender.grouperRemedy.DatePattern                = '.'yyyy-MM-dd
log4j.appender.grouperRemedy.layout                     = org.apache.log4j.PatternLayout
log4j.appender.grouperRemedy.layout.ConversionPattern   = %d{ISO8601}: %m%n

log4j.logger.edu.internet2.middleware.grouperRemedy.GrouperRemedyLog =  DEBUG, grouperRemedy
log4j.additivity.edu.internet2.middleware.grouperRemedy.GrouperRemedyLog = false
```

Configure the grouper.client.properties

```
#########################################################
## REMEDY
#########################################################

# base url for remedy
remedyGrouperClient.webService.url = https://school-restapi.onbmc.com

# username for remedy
remedyGrouperClient.webService.username =

# password for remedy
remedyGrouperClient.webService.password =

# put groups in here which go to remedy, the name in remedy will be the extension here
grouperRemedy.folder.name.withRemedyGroups = remedy:groups

# put the comma separated list of sources to send to remedy
grouperRemedy.sourcesForSubjects = jdbc

# either have id for subject id or an attribute for the remedy username (e.g. netId)
grouperRemedy.subjectAttributeForRemedyUsername = email

# if there is a suffix (e.g. @institution.edu then append that to the subject attribute to make the remedy
login id
grouperRemedy.subjectIdSuffix =

# if there is a require group that users must be in to be a user in remedy
grouperRemedy.requireGroup = remedy:remedyUser

# how long to cache remedy users in the requireGroup in grouper
grouperRemedy.cacheGrouperUsersForMinutes = 60

# is grouper the true system of record, delete remedy groups which dont exist in grouper
# note, if you delete the remedy group, if it is recreated, then shares wont exist
grouperRemedy.deleteGroupsInRemedyWhichArentInGrouper = false

#the quartz cron is a cron-like string.
```

```
# http://www.quartz-scheduler.org/documentation/quartz-1.x/tutorials/crontrigger
grouperRemedy.fullSync.quartzCron = 0 0 5 * * ?

# should log in the event log if no messages
grouperRemedy.logIfNoMessages = false

# messaging config for incremental changes, blank to use default
grouperRemedy.messaging.systemName = grouperBuiltinMessaging
# queueName is required for incremental provisioning
grouperRemedy.messaging.queueName = remedy_queue

# if you want to perform a full sync with each message received (note, assumes only applicable messages are
sent)
# note, will wait X 30? seconds, then mark subsequent messages as complete for those 30 seconds
grouperRemedy.fullSyncOnMessage = false

# note, this must be at least 5 seconds
grouperRemedy.fullSyncOnMessageWaitSeconds = 30

#the quartz cron is a cron-like string.
# http://www.quartz-scheduler.org/documentation/quartz-1.x/tutorials/crontrigger
# this defaults to every 30 seconds since the messaging long polls for 20 seconds.
grouperRemedy.incrementalSync.quartzCron = 0/30 * * * * ?

# if a user is not in the grouperRemedy.requireGroup group, then set the user's status to inactive,
cannot_delete_edit, or cannot_delete_edit_upload
# if this is blank then dont worry about it
# be careful that you dont lock out your admin account(s), whitelist below
grouperRemedy.statusDeprovisionedUsers =

# if a user is not in the grouperRemedy.requireGroup group, then set is_sync_enabled to false
grouperRemedy.deprovisionDisableSync = false

# if a user is in the grouperRemedy.requireGroup group, then set the user's status to active
# if this is blank then dont worry about it
grouperRemedy.statusUndeprovisionedUsers =

# if a user is in the grouperRemedy.requireGroup group, then set is_sync_enabled to true
grouperRemedy.undeprovisionEnableSync = false

# these could be administrative id's to never invalidate, comma separated
grouperRemedy.whitelistRemedyIds = a@b.c, b@c.d
```

Configure as a message listener

Send messages from the Remedy folder to the remedy message queue. Note: edit the filter and subject attributes as needed

```
changeLog.consumer.remedyEsb.class = edu.internet2.middleware.grouper.changeLog.esb.consumer.EsbConsumer
changeLog.consumer.remedyEsb.quartzCron = 0 * * * * ?
# carefully adjust this filter e.g. for sourceId and groupName
changeLog.consumer.remedyEsb.elfilter = (event.sourceId == null || event.sourceId eq 'jdbc') && (event.
groupName =~ '^remedy\\:groups\\:.*$' || event.groupName eq 'remedy:remedyUser' || event.name =~ '^remedy\\:
groups\\:.*$' || event.name eq 'remedy:remedyUser') && (event.eventType eq 'GROUP_DELETE' || event.eventType eq
'GROUP_ADD' || event.eventType eq 'GROUP_UPDATE' || event.eventType eq 'MEMBERSHIP_DELETE' || event.eventType
eq 'MEMBERSHIP_ADD' || event.eventType eq 'MEMBERSHIP_UPDATE')
# if messaging
changeLog.consumer.remedyEsb.publisher.class = edu.internet2.middleware.grouper.changeLog.esb.consumer.
EsbMessagingPublisher
# if change log
# changeLog.consumer.remedyEsb.publisher.class = edu.internet2.middleware.grouperRemedy.RemedyEsbPublisher
changeLog.consumer.remedyEsb.publisher.messagingSystemName = grouperBuiltinMessaging
# queue or topic
changeLog.consumer.remedyEsb.publisher.messageQueueType = queue
changeLog.consumer.remedyEsb.publisher.queueOrTopicName = remedy_queue
# this is optional if not using "id" for subjectId, need to be a subject attribute in the sources.xml
changeLog.consumer.remedyEsb.publisher.addSubjectAttributes = email
```

Copy the grouper remedy jar and all jars in the lib dir to the same directory as the grouper client jar

Run the service

```
java8/bin/java -cp lib/*:classes edu.internet2.middleware.grouperRemedy.GrouperRemedySync
```

Configure as change log consumer

You might want to use a change log consumer since it runs in the same process as the grouper daemon, and the grouper status will monitor to make sure the process is running successfully.

Copy the grouper remedy jar to the grouper daemon lib dir. Edit the configs

Edit the grouper-loader.properties

```
changeLog.consumer.remedyEsb.class = edu.internet2.middleware.grouper.changeLog.esb.consumer.EsbConsumer
changeLog.consumer.remedyEsb.quartzCron = 0 * * * * ?
# carefully adjust this filter e.g. for sourceId and groupName
changeLog.consumer.remedyEsb.elfilter = (event.sourceId == null || event.sourceId eq 'jdbc') && (event.
groupName =~ '^remedy\\:groups\\:.*$' || event.groupName eq 'remedy:remedyUser' || event.name =~ '^remedy\\:
groups\\:.*$' || event.name eq 'remedy:remedyUser') && (event.eventType eq 'GROUP_DELETE' || event.eventType eq
'GROUP_ADD' || event.eventType eq 'GROUP_UPDATE' || event.eventType eq 'MEMBERSHIP_DELETE' || event.eventType
eq 'MEMBERSHIP_ADD' || event.eventType eq 'MEMBERSHIP_UPDATE')
# if change log
changeLog.consumer.remedyEsb.publisher.class = edu.internet2.middleware.grouperRemedy.RemedyEsbPublisher
# this is optional if not using "id" for subjectId, need to be a subject attribute in the sources.xml
changeLog.consumer.remedyEsb.publisher.addSubjectAttributes = email


otherJob.remedy.class = edu.internet2.middleware.grouperRemedy.RemedyOtherJob
otherJob.remedy.quartzCron = 0 0 8 * * ?
```

Bounce the loader

Sample logs

```
2018-09-24 23:26:00,898: method: RemedyEsbPublisher.dispatchEvent, elapsedMillis: 482
2018-09-24 23:29:00,296: method: retrieveRemedyGroups, getMillis: 152ms, size: 6, elapsedMillis: 153
2018-09-24 23:29:00,296: method: processMessage, eventType: MEMBERSHIP_ADD, groupName: penn:isc:ait:apps:remedy:
groups:pg_hire_IT_cl
ients, groupInRemedy: false, elapsedMillis: 153
2018-09-24 23:29:00,296: method: RemedyEsbPublisher.dispatchEvent, elapsedMillis: 156
2018-09-24 23:32:00,476: method: retrieveRemedyGroups, authnMillis: 191ms, getMillis: 115ms, size: 6,
elapsedMillis: 308
2018-09-24 23:32:00,476: method: processMessage, eventType: MEMBERSHIP_DELETE, groupName: penn:isc:ait:apps:
remedy:groups:pg_hire_IT
_clients, groupInRemedy: false, elapsedMillis: 308
```

Internal technical details

Remedy rest API

Remedy API guide: https://docs.bmc.com/docs/ars91/en/developing-an-api-program-609071507.html

Use Chrome Advanced Rest Client to test calls: go to: chrome ://apps

URL encode params here for testing: https://meyerweb.com/eric/tools/dencoder/

https://docs.bmc.com/docs/itsm91/bmc-remedy-itsm-integrations-608491969.html

https://www.scribd.com/doc/2551868/White-Paper-BMC-Service-Request-Management-2-0-Architecture

Authentication

- https://docs.bmc.com/docs/display/public/ars9000/Login+information
- https://www.youtube.com/watch?v=xue9Gx-dbEA&feature=youtu.be
- https://school-env-restapi.onbmc.com

```
HEADER:
Content-Type: application/x-www-form-urlencoded

REQUEST
/api/jwt/loginjwt/login
POST with username and password form param

RESPONSE
200
Body:
eyJhbXXXXXXXXXXX
```

All users

```
REQUEST
GET /api/arsys/v1/entry/CTM:People?fields=values(Person%20ID,Remedy%20Login%20ID)     ( do not URL encode)
authorization AR-JWT eyJhbGXXXXXXXXXXXXX

RESPONSE
{
   "entries" : [
     {
       "values" : {
         "Person ID" :  "PPL000000000306",
         "Remedy Login ID" :  "foundationdataadmin",
         "Assignee Groups" :  "1000000023;"
       },
       "_links" : {
         "self" : [
           {
             "href" :  "https://school-dev-restapi.onbmc.com/api/arsys/v1/entry/CTM:People/PPL000000000306"
           }
         ]
       }
     }
   ],
   "_links" : {
     "self" : [
       {
         "href" :  "https://school-dev-restapi.onbmc.com/api/arsys/v1/entry/CTM:People"
       }
     ]
   }
}
```

One user

```
REQUEST
GET /api/arsys/v1/entry/CTM:People?q=%27Remedy%20Login%20ID%27%20%3D%20%22foundationdataadmin%22&fields=value
s(Person%20ID,Remedy%20Login%20ID)
URL encoded from:  'Remedy Login ID' =  "foundationdataadmin"
authorization AR-JWT eyJhbGXXXXXXXXXXXXX

RESPONSE
{
   "entries" : [
     {
       "values" : {
         "Person ID" :  "PPL000000000306",
         "Remedy Login ID" :  "foundationdataadmin"
       },
       "_links" : {
         "self" : [
           {
             "href" :  "https://school-dev-restapi.onbmc.com/api/arsys/v1/entry/CTM:People/PPL000000000306"
           }
         ]
       }
     }
   ],
   "_links" : {
     "self" : [
       {
         "href" :  "https://school-dev-restapi.onbmc.com/api/arsys/v1/entry/CTM:People?q=%27Remedy%20Login%20ID%
27%20%3D%20%22foundationdataadmin%22&fields=values(Person%20ID,Remedy%20Login%20ID)"
       }
     ]
   }
}
```

All groups

```
REQUEST
GET /api/arsys/v1/entry/ENT:SYS-Access%20Permission%20Grps?fields=values(Status,Permission%20Group,Permission%
20Group%20ID)

RESPONSE
{
   "entries" : [
     {
       "values" : {
         "Status" :  "Enabled" ,
         "Permission Group" :  "Sample Entitlement Group 1" ,
         "Permission Group ID" :  2000000001
       },
       "_links" : {
         "self" : [
           {
             "href" :  "https://school-env-restapi.onbmc.com/api/arsys/v1/entry/ENT:SYS-Access%20Permission%20Grps
/EGP000000000001"
           }
         ]
       }
     }
   ],
   "_links" : {
     "self" : [
       {
         "href" :  "https://school-env-restapi.onbmc.com/api/arsys/v1/entry/ENT:SYS-Access%20Permission%20Grps?
fields=values(Status,Permission%20Group,Permission%20Group%20ID)"
       }
     ]
   }
}
```

All memberships

```
REQUEST
GET /api/arsys/v1/entry/ENT:SYS%20People%20Entitlement%20Groups?fields=values(People%20Permission%20Group%20ID,
Permission%20Group,Permission%20Group%20ID,Person%20ID,Remedy%20Login%20ID,Status)

RESPONSE
{
   "entries" : [
     {
       "values" : {
         "People Permission Group ID" :  "EPG000000000002" ,
         "Permission Group" :  "2000000001" ,
         "Permission Group ID" :  2000000001 ,
         "Person ID" :  "PPL000000000405" ,
         "Remedy Login ID" :  "jsmith" ,
         "Status" :  "Enabled"
       },
       "_links" : {
         "self" : [
           {
             "href" :  "https://school-env-restapi.onbmc.com/api/arsys/v1/entry/ENT:SYS%20People%20Entitlement%
20Groups/EPG000000000002"
           }
         ]
       }
     }
   ],
   "_links" : {
     "self" : [
       {
         "href" :  "https://school-env-restapi.onbmc.com/api/arsys/v1/entry/ENT:SYS%20People%20Entitlement%
20Groups?fields=values(People%20Permission%20Group%20ID,Permission%20Group,Permission%20Group%20ID,Person%20ID,
Remedy%20Login%20ID,Status)"
       }
     ]
   }
}
```

Create membership

```
FIRST, SEE IF MEMBERSHIP EXISTS AND IS DISABLED

IF DOESNT EXIST:

REQUEST
POST /api/arsys/v1/entry/ENT:SYS%20People%20Entitlement%20Groups

    {
      "values" : {
        "Permission Group ID" :  2000000001 ,
        "Permission Group" :  "2000000001" ,
        "Person ID" :  "PPL000000000616" ,
        "Remedy Login ID" :  "jsmith"
      }
    }

RESPONSE
201

IF DOES EXIST:

REQUEST
PUT /api/arsys/v1/entry/ENT:SYS%20People%20Entitlement%20Groups/EPG000000000101


    {
      "values" : {
        "Permission Group ID" :  2000000001 ,
        "Permission Group" :  "2000000001" ,
        "Person ID" :  "PPL000000000616" ,
        "Remedy Login ID" :  "jsmith" ,
        "Status" :  "Enabled"
      }
    }
```

Delete membership

```
  REQUEST
PUT /api/arsys/v1/entry/ENT:SYS%20People%20Entitlement%20Groups/EPG000000000101



    {
      "values" : {
        "Permission Group ID" :  2000000001 ,
        "Permission Group" :  "2000000001" ,
        "Person ID" :  "PPL000000000616" ,
        "Remedy Login ID" :  "jsmith" ,
        "Status" :  "Delete"
      }
    }
```