Provisioning Consumer

Overview

This project aims to expose a simple API for creating "provisioning connectors." A "provisioning connector" **watches for changes** in Grouper and then **modifies an external system** based on the changes it sees happening in Grouper. This is implemented as a "changelog consumer" which digests the changelog data and then initiates your connector, passing the data in a simple-to-use data structure. Your connector can then contact the external system to push changes (such as group membership adds or removes) to that external system.

The Provisioning Consumer is based on the "EsbConsumer" that comes with the standard Grouper distribution.

Because the Provisioning Consumer handles most of the "grunt work" and allows the developer to work with a simpler API and focus only on the code necessary to communicate with the external system, we believe that it can significantly lower the bar of complexity for implementing a connector, decreasing the cost of integration.

Areas that need improvement

Currently this does **not** yet provide either "real-time provisioning" or "reconciliation" (handling incoming data from the external system for bi-directional synchronization).

License

This software is released under the open source Apache 2 License (the same license used by Grouper itself).

Download

[Download the attachment here](#)

Description

A few features:

1. The straight-forward EventProvisioningConnector interface makes it really simple to implement new connectors.
2. If you want to send messages in blocks, you can extend BufferedEventProvisioningConnector instead of implementing EventProvisioningConnector.
3. You can use the @ConfigItem annotation on connector attributes to read configuration settings from the grouper-loader.properties file.
4. Reflection is used to copy values from ChangeLogEntry to ChangeEvent, which will make it simple to adapt to any new values added to ChangeLogEntry.

I also included the source code for our Siebel connector as an example, but you won't be able to build it, since I didn't include the many dependencies. It's just there for reference.

How Create a Connector

Steps:

1. Implement EventProvisioningConnector
   a. Implement the "dispatchEvent()" method, which dispatches a single event.
   b. Implement the "close()" method, which cleans up resources (e.g. database connections) when the connector has finished its work.
   c. Add the "@ConfigItem" annotation to any attributes of your implementation that should be "injected" from grouper-loader.properties
2. Compile and create a jar file containing your implementation and the Provisioning Consumer class files
3. Copy the jar to .../grouper/lib/custom
4. Edit .../grouper/conf/grouper-loader.properties as described below.
5. Run grouper loader with the following command: gsh -loader

Configure your connector in grouper-loader.properties:

```
changeLog.consumer.<connector-name>.quartzCron = <cron interval>
changeLog.consumer.<connector-name>.class = edu.internet2.middleware.grouper.changeLog.provisioning.
ProvisioningConsumer
changeLog.consumer.<connector-name>.elfilter = event.eventType eq 'MEMBERSHIP_DELETE' \|\| event.eventType eq
'MEMBERSHIP_ADD'
changeLog.consumer.<connector-name>.connector.class = <your-connector-class>
changeLog.consumer.<connector-name>.connector.<prop1-name> = <prop1-value>
changeLog.consumer.<connector-name>.connector.<prop2-name> = <prop2-value>
changeLog.consumer.<connector-name>.connector.<prop3-name> = <prop3-value>
```

Example:

```
changeLog.consumer.myconnector.quartzCron = 0 * * * * ?
changeLog.consumer.myconnector.class = edu.internet2.middleware.grouper.changeLog.provisioning.
ProvisioningConsumer
changeLog.consumer.myconnector.elfilter = event.eventType eq 'MEMBERSHIP_DELETE' \|\| event.eventType eq
'MEMBERSHIP_ADD'
changeLog.consumer.myconnector.connector.class = org.ccci.idm.groupersiebelpc.SiebelConnector
changeLog.consumer.myconnector.connector.username = myusername
changeLog.consumer.myconnector.connector.password = my-password
changeLog.consumer.myconnector.connector.url = <siebel-url-here>
```