GrouperClient failover client API

The Grouper Failover Client API is included in the Grouper client v2.1+

This is an API where you can specify a list of endpoints and some logic and it will run against them until it finds one that returns a successful response before a timeout has occurred.

This can be used for any type of endpoint: web service, database, ldap, etc.

The failover client will remember which connnections have how many errors to prefer not to use connections with more errors than others.

If the failover is used in command-line mode, the state can be saved periodically to disk so each invocation remembers the state of the previous.

This API is used in the Grouper client optionally for the discovery service and always available web services and ldap.

Usage

Include the grouperClient jar, and call the failover client:

```
String result = FailoverClient.failoverLogic("myEndPointType", new FailoverLogic<String>() {

        /**
         * @see FailoverLogic#logic
         */
        @Override
        public String logic(FailoverLogicBean failoverLogicBean) {
          ... do some logic based on this connection: failoverLogicBean.getConnectionName()
          return theResult which is a String since it was declared this type above: FailoverLogic<String>;
        }
      });
```

Configuration

There is a configuration bean (which you would likely read from a config file, but could be programmatic), and you register the configuration for this type before using it (e.g. on startup)

```
FailoverConfig failoverConfig = new FailoverConfig();

//if there are no errors in connections, then use the same connection for 30 minutes
failoverConfig.setAffinitySeconds(2400);

//if there are no errors in connections, it will use the 1st tier connection names before the 2nd tier
//note that this configuration example will only work for readonly queries, if it is a readwrite query,
//then there would not be any readonlyConnections available
failoverConfig.setConnectionNames(GrouperClientUtils.toSet("readWriteConnection1", "readWriteConnection2");
failoverConfig.setConnectionNamesSecondTier(GrouperClientUtils.toSet("readonlyConnection1",
"readonlyConnnection2", readonlyConnection3");

//this is a label to identify this "pool" of connections
failoverConfig.setConnectionType("myEndPointType");

//if there are no errors in connections, and if there is no affinity, this is the strategy to pick which
connection
//active/active will pick one at random from the 1st tier connections, active/standby will use the connections
in order
failoverConfig.setFailoverStrategy(FailoverStrategy.activeActive);

//this is how long it will try on connection and wait for a response until giving up and trying another
connection
failoverConfig.setTimeoutSeconds(120);

//after you have cycled through all the connections you can wait a little longer for one of the connections to
finish
failoverConfig.setExtraTimeoutSeconds(50);

//this is how much time to remember that a connection had errors.  If an error hasnt occurred in a certain
amount of time,
//it will be forgotten
failoverConfig.setMinutesToKeepErrors(5);

//if you have a lot of hibernate mapped classes or something, it will give a buffer for the first X seconds for
the JVM
//to load and initialize
failoverConfig.setSecondsForClassesToLoad(20);

//register this configuration
FailoverClient.initFailoverClient(failoverConfig);
```

Here are some settings in grouper.client.properties. Note saving to disk takes a few milliseconds

```
########################################
## Misc settings
########################################

# path of a writable directory where files can be created or stored
# for example, cache of discovery configuration, or failover state
# dot is the current directory...  note, this directory must exist
# or it will be created (attempted)
# if this is blank, none of these features will be used, and
# no files will be saved
grouperClient.cacheDirectory = .

# this will save the failover state to a file so if the JVM is stopped, it
# will be there when it starts again.
# Set to 0 to store on every use (recommended if used command line)
# or set to -1 to not store or read ever
# grouperClient.cacheDirectory must be set
grouperClient.saveFailoverStateEverySeconds = 60
# if the failover client should use threads.  If it doesnt then you cant detect timeouts
grouperClient.failoverClientUseThreads = true
```

To do

- Could provide retry after sleeping if not success...
- Could have error count where below error count does not change connection affinity

See Also

Grouper Client

Discovery Client