Advanced Usage: REST, PSQL, and SSH

While Selenium access to the XNAT webapp is great and flexible, there are certain operations which simply cannot be done simply in the webapp. To address this, the NRG_Selenium framework allows REST calls (via REST-Assured), XNAT database access, and SSH connections to perform commands directly on the test server.

REST

Since most of XNAT requires authentication, one important aspect is allowing that in the framework. To obtain a credentials object (more specifically, a **RequestSpecification**), you can simply make a call to **Settings.credentials(...)**. With these credentials, the usual PUTs, GETs, POSTs, etc. of REST calls are available at your fingertips. A couple of code examples:

```
if (credentials.get(String.format("%s/JSESSION", Settings.RESTURL)).getStatusCode() != 200) System.out.println
("Session invalid");
Settings.seleniumCredentials().expect().statusCode(403).put(String.format("%s/projects/%s", Settings.RESTURL,
testSpecificProject));
```

Database Commands

The NRG_Selenium framework also allows the tester to interact directly with XNAT's database. To do this, the properties which begin with **xnat.db** listed in Setup and Configuration must be set up correctly, and the database must be set up to allow the tester's IP to access it (via the **pg_hba.conf** file). Then, simply calling **XnatDatabase.performXnatDatabaseQuery(...)**, providing the SQL command as the parameter will pass the command on to the XNAT database. The method will return a **ResultSet** with the results of the command. If you wish to modify the database, instead use the **XnatDatabase.performXnatDatabaseUpdate(...)** method, which will return an **int** representing the number of database rows affected by the statement.

SSH

The framework also allows SSH connections to the XNAT machine. To do this, the **xnat.ssh** properties must be configured correctly. The property for **xnat.ssh.key** specifies the name (without full path) of the SSH private key which will be used to authenticate to the XNAT machine. The key must be placed in **~/.ssh** to be found. Then, call **SSHConnection.executeCommand(...)** with the desired command to be executed on the machine as the parameter.