Hi,  
  
I'd like each of the examples to get the URL, application name, application ID (key), username and password from a "properties" file, named "pycipres.conf".    The program should look for pycipres.conf in the user's home directory as well as in $SDK\_VERSIONS/testdata, where $SDK\_VERSIONS is an environment variable that may or may not be set on the user's computer.   If pycipres.conf exists in both places, the settings from the home directory should take precedence.  
  
The format of the properties file is a simplified version of the java properties file format.  Each line has the form: NAME=VALUE.  Comment lines start with "#".  The file may include blank lines.    If a NAME appears more than once, the last definition is the one that's used.  Java, can of course, read and write these files very easily using the Properties class.   For python, I use a library called pyjavaproperties.  I understand that in perl you can use<http://search.cpan.org/~salva/Config-Properties-1.75/lib/Config/Properties.pm>.  
  
Here's an example of ~/pycipres.conf for the python command line example in <https://svn.sdsc.edu/repo/scigap/trunk/rest_client_examples/examples/python_cipres> :  
  
URL=<http://localhost:7070/cipresrest/v1>  
APPNAME=pycipres  
APPID=pycipres-5A656A9748E24A2880489BD4DA2F43F5  
USERNAME=pycipres\_eu  
PASSWORD=test  
  
And here's an example for the java struts example in <https://svn.sdsc.edu/repo/scigap/trunk/rest_client_examples/examples/java_umbrella> :  
  
URL=<http://localhost:7070/cipresrest/v1>  
UMBRELLA\_APPNAME=guidemo  
UMBRELLA\_APPID=guidemo-CEF8403A682C44CF9435631CA01B9DD7  
ADMIN\_USERNAME=guidemo\_admin  
ADMIN\_PASSWORD=test  
  
Your examples should use "URL", but should use different names for the other properties.  For example, Anthony, your perl command line example could use:

A service (which could be anything) that listens on port number 7070 of the computer the client is running on.

URL=  
PERL\_EX\_APPNAME=  
PERL\_EX\_APPID=  
PERL\_EX\_USERNAME=  
PERL\_EX\_PASSWORD=  
  
( The reason, I want you to use different property names for different examples, is that I have our master build script put all the properties for all the examples in one big pycipres.conf, that I put in $SDK\_VERSIONS/testdata, on the machine where the REST service runs, so that I can easily test all the examples after I do a build.  I don't want all the examples to use the same appid and account because that would make looking at the server log files confusing.)  
  
EXAMPLES OF READING PYCIPRES.CONF:  
  
1. In <https://svn.sdsc.edu/repo/scigap/trunk/rest_client_examples/examples/python_cipres/python_cipres> we use pyjavaproperties.py to process properties files.  In client.py, look at the \_\_init\_\_ method of the Application class to see how we  get properties from $SDK\_VERSIONS/testdata/pycipres.conf, if it exists, and then overload them with properties from ~/pycipres.conf if it exists, and then check that the required properties have been set.  
  
2. In <https://svn.sdsc.edu/repo/scigap/trunk/rest_client_examples/examples/java_umbrella/src/main/java/org/ngbw/restclient,> see the class constructor in Application.java.   It looks for pycipres.conf first in $SDK\_VERSIONS/testdata and then in the user's home directory, only if the copy in $SDK\_VERSIONS/testdata wasn't found.   I will be changing this to be consistent with the python example, and my description of how it should work.  In other words, it should load properties from $SDK\_VERSIONS/testadata/pycipres.conf if it exists, and then overload them with the copy in the home directory if it also exists.  
  
Terri

I added one more location in which we'll look for the pycipres.conf properties file in the examples, mostly for the sake of running on windows.   If you set an environment variable named PYCIPRES to the full pathname of a properties file, we'll use that.   So, first we try to load properties from $SDK\_VERSIONS/testdata/pycipres.conf, then from $HOME/pycipres.conf and finally from $PYCIPRES.  It's ok if more than one exists; in that case, the ones loaded later override settings from the ones loaded first.  
  
Alexes, I haven't been able to test this on windows, so I'd appreciate it if you would and report back to me about whether or not it works.  Export the project:  https://svn.sdsc.edu/repo/scigap/trunk/rest\_client\_examples/examples/java\_umbrella.   Use the PYCIPRES env var and follow the instructions in the Readme.txt file to build and run.   You can use the GUIDEMO appID, username and password, as you were before.   (java\_umbrella is just restclient, renamed, and cleaned up a little).      Let me know if you're able to submit a job, see results, etc.  
  
Anthony, I think you'll find the code in java\_umbrella a good starting point for your DIRECT authentication, command line java client.

Give a printing line for each directory you are looking at for the configuration file.

Add details about the configuration file for the write up too.