

Running a Job on the Server to Generate Data through REST API

The following example demonstrates how a job created on the Repertoire Server can be invoked from the application side through the REST API.

This example would be applicable in situations where specific datasets are required on an ad-hoc basis.

The full sample class can be download [here](#).

Import the following Java references within the class environment:

```
import java.io.ByteArrayOutputStream;  
import java.io.File;
```

To talk to the server using the REST API, add the following .jar files from the "/lib" directory of the Repertoire Server installation:

commons-codec-1.3.jar

commons-httpclient-3.0.1.jar

commons-logging.jar

and import them into the class file:

```
import org.apache.commons.httpclient.Credentials;  
import org.apache.commons.httpclient.HttpClient;  
import org.apache.commons.httpclient.UsernamePasswordCredentials;  
import org.apache.commons.httpclient.auth.AuthScope;  
import org.apache.commons.httpclient.methods.ByteArrayRequestEntity;  
import org.apache.commons.httpclient.methods.PostMethod;  
import org.apache.commons.httpclient.methods.PutMethod;  
import org.apache.commons.httpclient.methods.RequestEntity;  
import org.apache.commons.httpclient.params.HttpClientParams;
```

The following are explanations and walkthroughs of the code. The full Java class can be downloaded [here](#).

```
//This method takes in the host url, port number and credentials to  
//establish a connection to the Repertoire Server
```

```

public static HttpClient ERSCoconnect(String host,Integer port,String username,String
password,String CredentialCharset)
{

    HttpClient client = new HttpClient();
    Credentials defaultCreds = new UsernamePasswordCredentials(username,password);
    client.getState().setCredentials(new AuthScope(host,port,AuthScope.ANY_REALM),
defaultCreds);
    HttpClientParams params = client.getParams();
    params.setCredentialCharset(CredentialCharset);
    params.setAuthenticationPreemptive(true);

    return client;

}

```

After creating the method mentioned above, implement the following code:

```

//Create a variable to hold the URL
String ERSURL = "http://localhost:8080";

//Instantiate a new connection to the Repertoire Server
HttpClient client = ERSCoconnect("localhost",8080,"admin","sa","UTF-8");

//Instantiate a new post method with the path of the job on the server
//and the output path of the mime-type
PostMethod post = new PostMethod(ERSURL+"/job/Elixir_Cookbook/Schedule-
Designer/GenerateData/datasources/GenerateData-Sample-001.job?
Fruit=Apple&URL=file:/c:/temp/testjob.csv");

//Execute the post method using the client connection
int ret = client.executeMethod(post);
if (ret==200) System.out.println("Successful !");
else System.out.println("Error ! " );

```