REST Clients

Overview

Documentation on REST Clients.

Details

Currently there are not Java clients implemented for the REST services but this would be a fairly straightforward thing to do. The advantage of having a Java client is that integration tests can easily be run in the dev enviornment against a deployment with known data contents. This would be the basis for full and comprehensive integration testing.

Here is a simple example of a client method that could call the Content REST Service for finding concepts by a query. This full example includes support for post request and reading the object back and unmarshalling it back into an object. In this case the "Utility" is a class that converts a string representation into an object graph using JAXB unmarshalling (with the jackson API for JSON support) - NOTE: this class is not part of the distribution but could be easily implemented.

```
public SearchResultList findConceptsForQuery(String terminology,
  String version, String searchString, PfsParameterJpa pfs, String authToken)
  throws Exception {
  Client client = Client.create();
  WebResource resource =
      client.resource(config.getProperty("base.url") + "/content/concepts/"
          + terminology + "/" + version + "/query/" + searchString);
  String pfsString =
      (pfs != null ? ConfigUtility.getStringForGraph(pfs) : null);
  ClientResponse response =
      resource.accept(MediaType.APPLICATION_XML)
          .header("Authorization", authToken)
          .header("Content-type", MediaType.APPLICATION_XML)
          .post(ClientResponse.class, pfsString);
  String resultString = response.getEntity(String.class);
  if (response.getClientResponseStatus().getFamily() == Family.SUCCESSFUL) {
    Logger.getLogger(this.getClass()).debug(resultString);
  } else {
    throw new Exception(resultString);
  }
  // converting to object
  SearchResultListJpa list =
      (SearchResultListJpa) Utility.getGraphForString(resultString,
          SearchResultListJpa.class);
  return list;
}
```

References/Links

- n/a