libJava-DAL

DAL client library for Java

This library lets you code client applications to use a DAL server and
not be concerned with the low-level details of maintaining the HTTP connection
the other assorted method calls required to GET or POST requests.

Support is also provided for automatically parsing the JSON or XML responses
and for retrieving or visiting the records in a format-independent fashion.

A pre-built copy of `JavadocDALClientDemo.java` is available as `javadoc-demo.jar`
and may be run from the project's root directory; `lib` directory is referenced
in the `Class-Path` of the jar file.

```
DALClient client = new DefaultDALClient(dalurl)
    .setAutoSwitchGroupOnLogin(true)     // auto-switch to the first group
    .setResponseType(ResponseType.JSON); // Note: "fluent" coding style is
available

try {
```

// **Step 1:** Login and show which group we got switched to.

```
    System.out.println("=== Step 1 =========");
    try {
        client.login(username, password);
    } catch (DalLoginException e) {
        System.err.println("Login failed: "+e.getMessage());
        return;
    } catch (DalResponseException e) {
        System.err.println("Login failed: "+e.getMessage());
        return;
    } catch (IOException e) {
        System.err.println("Login failed: "+e.getMessage());
        return;
    }

    System.out.println("Logged in as id="+client.getUserId()+
                   ", groupId="+client.getGroupId()+
                   ", groupName="+client.getGroupName());
```

// Output is something like:

// Logged in as id=7, groupId=2, groupName=users

// **Step 2:** Print out all of the Genus records in the database in JSON format

```
    System.out.println("=== Step 2 =========");

    DalResponse genusResponse = client.performQuery("list/genus");
    genusResponse.printOn(System.out);

    // Find the smallest and largest GenusId in the results
    final Integer[] minMaxGenusId = new Integer[2];

    DalResponseRecordVisitor visitor = new DalResponseRecordVisitor() {
        @Override
        public boolean visitResponseRecord(String tagName, DalResponseRecord
record) {

            Integer genusId = new Integer(record.rowdata.get("GenusId"));

            if (minMaxGenusId[0] == null || genusId < minMaxGenusId[0]) {
                minMaxGenusId[0] = genusId;
            }

            if (minMaxGenusId[1] == null || genusId > minMaxGenusId[1]) {
                minMaxGenusId[1] = genusId;
            }
            return true; // look at all records
        }
    };

    genusResponse.visitResults(visitor, "Genus");

    System.err.println("GenusIds in range [" + minMaxGenusId[0]
             + " to " + minMaxGenusId[1] + "]");
```

// **Step 3:** We want to display the first 5 GenotypeAlias records

// that have a *GenotypeAliasName* starting with 'MUTANT'.

// In this variation, we use a *CommandBuilder* to replace the parameters in a DAL

// operation with specific values.

```
    String cmd;
    try {
        cmd = new CommandBuilder("list/genotypealias/_nperpage/page/_num")
            .setParameter("_nperpage", "5")
            .setParameter("_num",      "1")
            .setFilterClause("GenotypeAliasName LIKE 'MUTANT%'") // this is
optional
            .build();
    } catch (DalMissingParameterException e) {
        throw new RuntimeException(e);
    }

    System.out.println("=== Step 3 =========");
    client.performQuery(cmd).printOn(System.out);
```

// Alternatively, you could do it this way:

```
    DalResponse response =
client.prepareQuery("list/genotypealias/_nperpage/page/_num")
        .setParameter("_nperpage", "5")
        .setParameter("_num",      "1")
        .setFilterClause("GenotypeAliasName LIKE 'MUTANT%'") // this is
optional
        .execute();

    response.printOn(System.out);
```

// **Step 4:** Display the details for a specific Genus using XML.

// This command is so simple we do not need to use *CommandBuilder*.

```
    System.out.println("=== Step 4 =========");

    client.setResponseType(ResponseType.XML); // change to XML
    for (Integer id : minMaxGenusId) {
        if (id != null) {
            DalResponse rsp = client.performQuery("get/genus/" + id);
            System.out.println("GenusName#" + id + "="
                + rsp.getRecordFieldValue("Genus", "GenusName"));
        }
    }

} catch (DalResponseException e) {
    System.err.println("Query failed: "+e.getMessage());
} catch (IOException e) {
    System.err.println("Query failed: "+e.getMessage());
} catch (DalMissingParameterException e) {
    System.err.println("Query failed: "+e.getMessage());
} finally {
    // Make sure that we finish off the session.
    // If we didn't get logged in, this is a NO-OP.
    client.logout();   // Note that DefaultDALClient.finalize() also does this
}
```