Sourcecode für Galaxy Beispielapplikation in Java

Table of contents

1. Beispielapplikation in Java

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Iterator;
import java.util.List;
import java.util.Map;

import com.sagadc.galaxy.api.GalaxyFactory;
import com.sagadc.galaxy.api.GalaxyRequest;

/**
 * This is a sample application to demonstrate the use of the GalaxyApi and
the M31.Galaxy
 * Eclips Plugin. To run this program, you will need to setup a local
galaxy server that
 * provides the required Containers.
 *
 * This is a simpe one shot application which does the following things:
 *
 *  - Asks Galaxy for a list of all Records defined in the sample database
 *  - Displays the fetched List
 *  - Lets the user enter the number of one of the displayed records
 *  - Runs another Galaxy container to fetch details for this record
 *  - Prints out this details
 */
public class GalaxySample
{

  //Input/Output fields, defined by Galaxy
  private static final String OUT_LASTNAME = "LastName";
  private static final String OUT_FIRSTNAME = "FirstName";
  private static final String CONTACTID = "ContactId";

  //Request names, defined by galaxy.cfg.xml
  private static final String REQUEST_TSC_USERLIST = "TSCUserList";
  private static final String REQUEST_TSC_DETAILS = "TSCUserDetails";

  public static void main(String[] args) {
    /*
     * [1] Get the request called REQUEST_TSC_USERLIST from the factory
     */
    GalaxyRequest listRequest = GalaxyFactory
                    .getFactoryInstance()
                    .getRequest(REQUEST_TSC_USERLIST);

    /*
     * [2] Run the request (No input fields are required)
     */
    listRequest.doRequest();
    checkReturncode(listRequest);
    /*
     * [3] Get the item, which represents the user table
     */
    List userTable = (List) listRequest.getOutputItem("UserTSCInc");

    /*
     * Print the list of user
     */

    System.out.println("List of all users:");
    System.out.println("------------------");
    for(int i = 0; i &lt; userTable.size(); i++)
    {
      Map currentRow = (Map) userTable.get(i);
      System.out.println(i + ": "
                 + currentRow.get(OUT_FIRSTNAME)
                 + " "
                 + currentRow.get(OUT_LASTNAME));
    }
    System.out.println("------------------");

    /*
     * [4] Get the users selection
     */
    int selection = getUserSelection(userTable.size());

    /*
     * [5] Fetch the ID and shoot another request to get the details of the
selected user
     */
    Map selectedUser = (Map) userTable.get(selection);
    Object selectedUsersId = selectedUser.get(CONTACTID);

    /*
     * [6] Get the Request called REQUEST_TSC_DETAILS
     */
    GalaxyRequest detailRequest = GalaxyFactory
                      .getFactoryInstance()
                      .getRequest(REQUEST_TSC_DETAILS);

    /*
     * [7] Add the selected users id as inputItem
     */
    detailRequest.addInputItem(CONTACTID, selectedUsersId);

    /*
     * Trigger the request (see [2])
     */
    detailRequest.doRequest();
    checkReturncode(detailRequest);

    /*
```

```
       * [8] Print out all outcomming items for the selected user
       */
    Map userDetails = detailRequest.getOutputItems();
    Iterator keys = userDetails.keySet().iterator();

    System.out.println();
    System.out.println("Details for selected User:");
    System.out.println("--------------------------");
    while(keys.hasNext())
    {
      String currentKey = keys.next().toString();
      System.out.println(currentKey + ": " + userDetails.get(currentKey));
    }
    System.out.println("--------------------------");

  }

  /**
   * Checks weather the returncode of the given GalaxyRequest is equal to
   * zero. If not, the return message is print to the console and the
progran is
   * terminated.
   *
   * @param listRequest The Request to check
   */
  private static void checkReturncode(GalaxyRequest listRequest) {
    if(listRequest.getReturncode() != 0)
    {
      System.err.println("An error occurred during request:");
      System.err.println("Returncode: " + listRequest.getReturncode());
      System.err.println("Message: " + listRequest.getMessage());
      System.exit(-1);
    }
  }

  /**
   * Utility method to read an integer from the console, will exit
   * application if reading fails...
   *
   * @param maxValue the maximum value to be entered by the user
   * @return the entered number
   */
  private static int getUserSelection(int maxValue) {
    System.out.print("Enter number to view the details: ");

    //Read the users selection from the command line
    BufferedReader reader = new BufferedReader(
                new InputStreamReader(System.in));
    int selection = 0;
    try
    {
      String selectionString = reader.readLine();
      selection = Integer.parseInt(selectionString);
    }
    catch(NumberFormatException e)
    {
      //Entered Text was not a number
      System.err.println("Please enter a valid number!");
      System.exit(-1);
    }
    catch (IOException e)
    {
      //An unhandled error occured,
      e.printStackTrace();
      System.exit(-1);
    }

    //Check for proper input, exit application if not
    if(selection < 0 || selection >= maxValue)
    {
      System.err.println("Please enter a valid number between 0 and "
          + maxValue);
      System.exit(-1);
    }

    return selection;
  }
```