

Sample Twitter AJAX based web application using Sybase ASE database.

Sample end to end web application demonstrating use of
Twitter API with Sybase ASE

TABLE OF CONTENTS

| | |
|--|-----------|
| APPLICATION INTRODUCTION..... | 3 |
| SOFTWARE REQUIRED | 3 |
| Download and install Sybase ASE Developer Edition | 3 |
| Required Java libraries and Software | 4 |
| Setup Eclipse environment | 4 |
| Setup Sybase ASE for Twitter Application | 5 |
| Login to ISQL GUI Tool | 5 |
| Using Command Line iSQL | 7 |
| SOURCE CODE CONSTITUENTS | 7 |
| Application Workflow | 7 |
| Application Business Logic | 7 |
| UseCase-I Default Mode | 9 |
| UseCase-II: Custom Query Mode | 9 |
| DETAILED SOURCE CODE | 10 |
| Listing 1. Index.JSP | 10 |
| Listing 2: TwitterClient.java | 11 |
| Listing 3: TwitterHTTPAgent.java | 12 |

This guide outlines steps required to develop end to end simple web application with Sybase ASE as underlying database without any frameworks on Windows 7 environment using Eclipse IDE for Java EE.

APPLICATION INTRODUCTION

This end to end web application demonstrates how easy is it to build a web application using Twitter API , AJAX in Java using Sybase ASE database. Business logic of application is quite simple – if a user has specified a search String for Twitter, a servlet will parse JSON response from Twitter using that search string and store response in Sybase ASE database. Of course, results can directly be displayed on web page without this intermediate step of storing specific tweets in database, but as a sample application, we will be using this convoluted way of displaying latest tweet and this remove reliance on Twitter API for future analytics on such data. Main use-case for this application is for some analytics to be performed on data brought over multiple sources and since Sybase ASE is not that well known outside financial institutions or SAP environment, application should serve as good template for steps required to build simple end to end application in Java using current available open source Java libraries.

No prior experience with Sybase ASE is assumed, and this tutorial will walk through all steps right from download, install Sybase ASE, and configure IDE and how to make changes to database. Application is built over Tomcat web server but any other web container will do equally well.

SOFTWARE REQUIRED

Download and install Sybase ASE Developer Edition

- Download Sybase ASE developer edition from here http://www.sybase.com/ase_1500devel
 - \$ASE_HOME will refer to \$ASE_INSTALL/ASE_15_0 location
- Installation guide
 - Windows
 - Linux
- Verify installation is correct by making sure ASE server starts up, and one can query sample database using iSQL interactive tool.
 - Start the server on Windows by Control Panel→ Administrative Tools→Services→Sybase _ASE_SQL_Server
 - Query pubs3 database table “authors” “select * from authors”. This should be response

```
c:\Sybase5\OCS-15_0\bin>isql -Usa
Password:
1> use pubs3
2> go
1> select * from authors
2> go
 au_id      au_lname      address      state  country      postalcode
-----
409-56-7008 Bennet
510 658-9932 6223 Bateman St.
Berkeley      CA      USA      94705
213-46-8915 Green
510 986-7020 309 63rd St. #411
Oakland      CA      USA      94618
238-95-7766 Carson
510 548-7723 589 Darwin Ln.
Berkeley      CA      USA      94705
998-72-3567 Ringer
801 826-0752 67 Seventh Av.
Salt Lake City      UT      USA      84152
899-46-2035 Ringer
801 826-0752 67 Seventh Av.
Salt Lake City      UT      USA      84152
722-51-5454 DeFrance
219 547-9982 3 Balding Pl.
Gary      IN      USA      46403
807-91-6654 Panteley
301 946-8853 1956 Arlington Pl.
Rockville      MD      USA      20853
893-72-1158 McBaden
207 440-4002 201 Bateman
```

Required Java libraries and Software

Download below jar files in one location and add them to Eclipse project class path
First install following software.

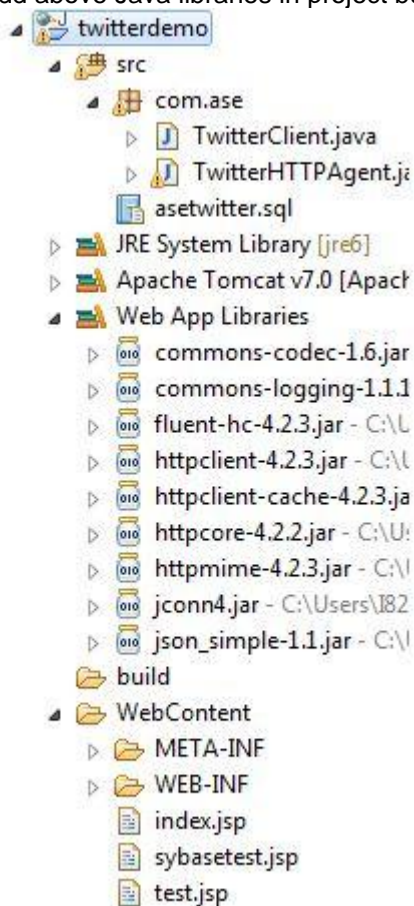
- Java 1.6 and above
- Eclipse Java EE IDE for Web Developers.- Juno Service Release 1.
- Tomcat Server 1.7

Required Java libraries for TwitterDemo project

- Sybase ASE jConnect Java JDBC Driver - located at \$ASE_HOME/jconnect/classes
- Apache HTTPClient <http://hc.apache.org/downloads.cgi>
- JSON simple jar http://code.google.com/p/json-simple/downloads/detail?name=json_simple-1.1.jar&can=2&q=
- <http://code.google.com/p/json-simple/>

Setup Eclipse environment

Create new Eclipse Dynamic web project called “twitterdemo” with Target Web Platform as Tomcat Server.
Add above Java libraries in project build path. It should now appear like this.



Web.xml for project should look like this

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.c
<display-name>Twitterdemo</display-name>
<welcome-file-list>
  <welcome-file>index.html</welcome-file>
  <welcome-file>index.htm</welcome-file>
  <welcome-file>index.jsp</welcome-file>
  <welcome-file>default.html</welcome-file>
  <welcome-file>default.htm</welcome-file>
  <welcome-file>default.jsp</welcome-file>
</welcome-file-list>
<servlet>
  <description></description>
  <display-name>TwitterClient</display-name>
  <servlet-name>TwitterClient</servlet-name>
  <servlet-class>com.ase.TwitterClient</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>TwitterClient</servlet-name>
  <url-pattern>/TwitterClient</url-pattern>
</servlet-mapping>
</web-app>
```

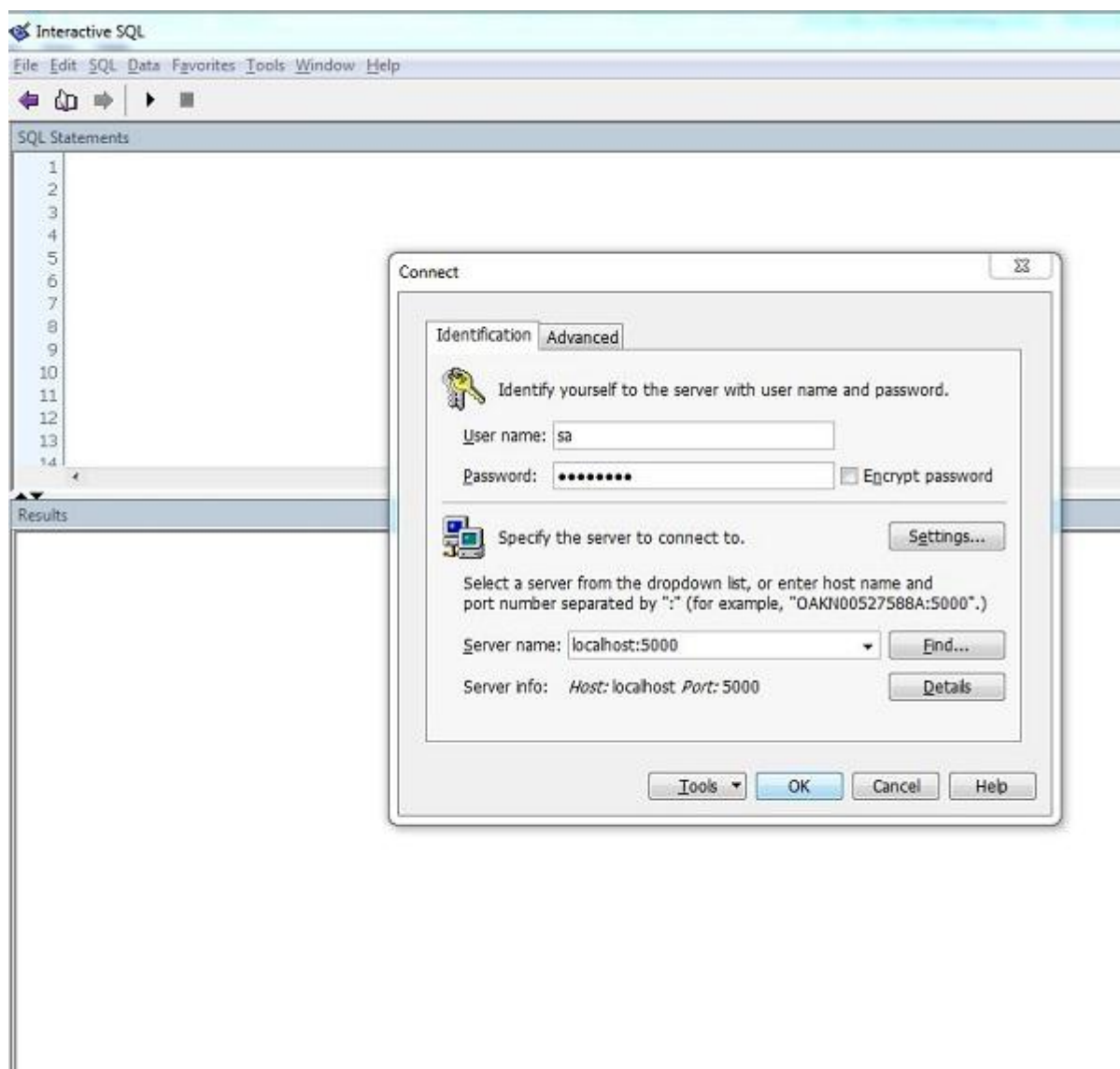
Setup Sybase ASE for Twitter Application

Using interactive SQL tool of Sybase ASE Server- either command-line or GUI, create following table under pubs3 sample database.

```
CREATE TABLE pubs3.dbo.tweets
(id numeric (7,0) identity,
twitter_user varchar(100) NULL,
dp_url varchar(100) NULL,
data varchar(200) NULL,
query varchar(200) null,
post_date bigint null,
primary key (id),created datetime default getDate());
```

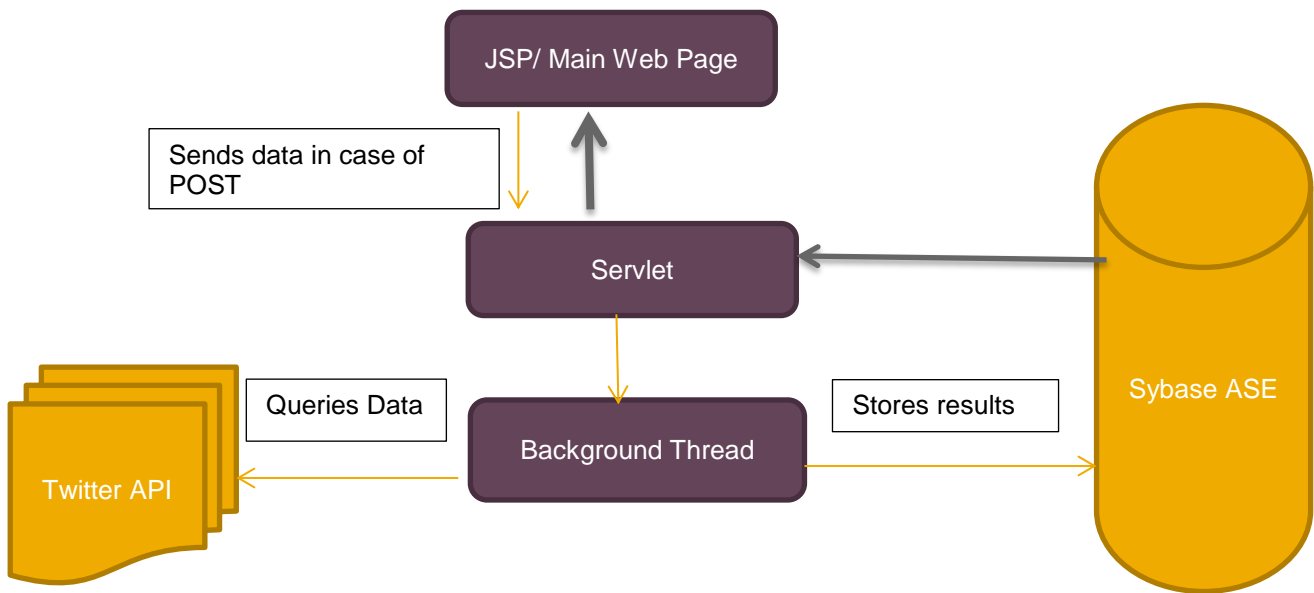
Login to ISQL GUI Tool

Sample Twitter AJAX based web application using Sybase ASE database.



Create Table command in GUI iSQL tool





There are two main use-cases of application

1. Default Mode: Application is started for the first time, and nothing is entered as query string
2. Custom Query String is entered

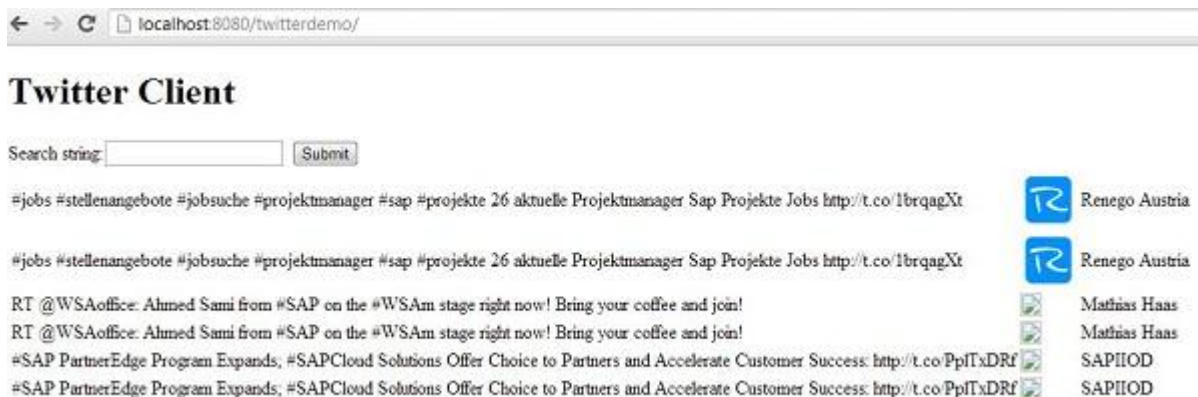
UseCase-I Default Mode

When application is invoked for the first time, database is empty, and since we haven't have given any search String, Twitterclient servlet doGet is invoked through index.jsp AJAX call which is fetching latest tweets from database

Background thread, which starts as soon as Servlet is loaded onto Tomcat or when application is first invoked, fetches results from Twitter and stores them into ASE Database. Every 20 seconds, data fetched from Twitter and inserted into ASE

TwitterClient Servlet → Initialization Method → TwitterHTTPAgent → Query Twitter → (Sample Twitter Query)
<http://search.twitter.com/search.json?q=%23HANA%20OR%20%23SAP%20OR%20%23SYBASE>
Get JSON Response → Store results in Sybase Ase → Results pulled by AJAX → TwitterClient doGet → get latest Tweets from ASE → SELECT * FROM pubs3.dbo.tweets WHERE post_date > (current_timestamp -10 seconds) ORDER BY post_date DESC → display onto web page

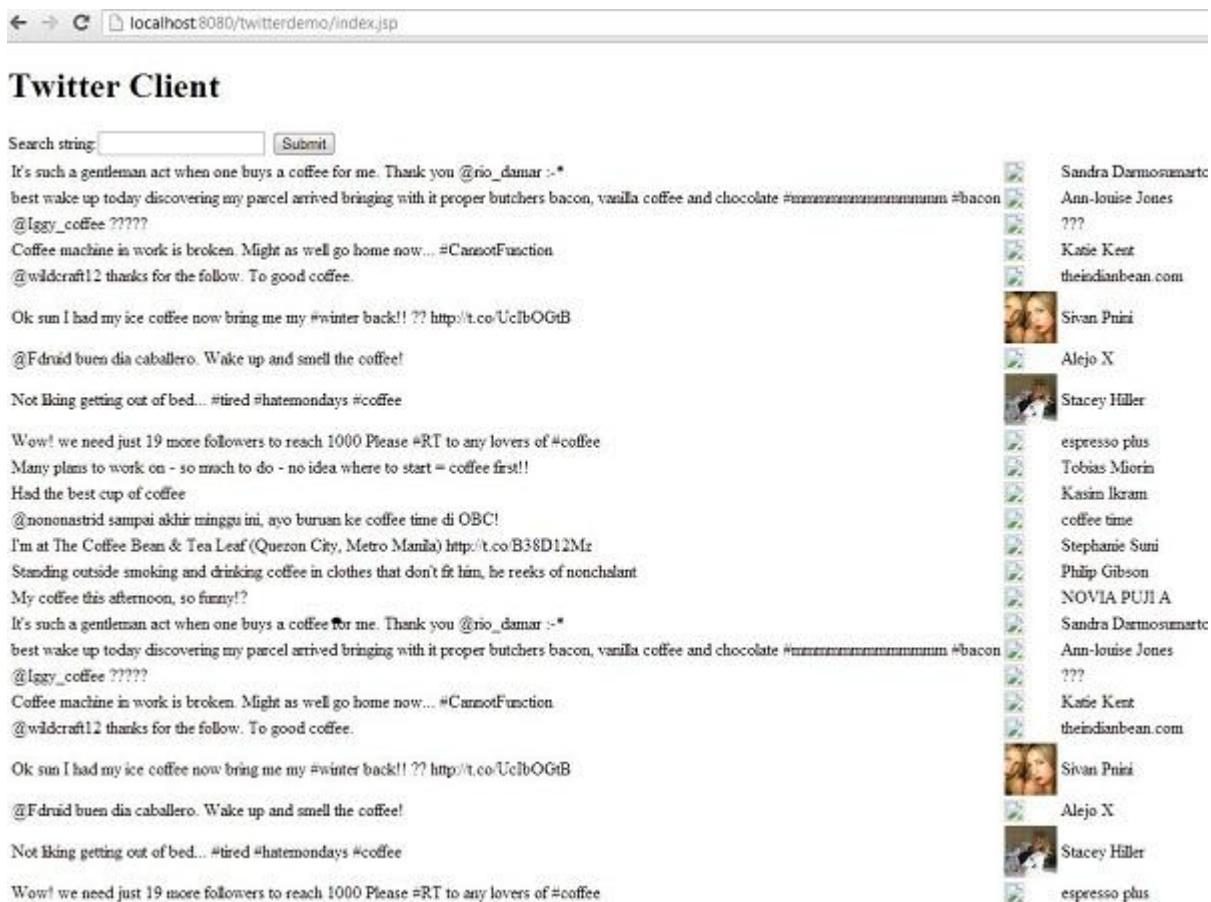
Here is output



UseCase-II: Custom Query Mode

In this scenario, background thread which polls after 20 seconds will now start storing results in Sybase ASE with JSON response from Twitter obtained based on user entered Twitter query. Once custom query string is sent over to TwitterClient servlet, index.jsp AJAX call would keep fetching and displaying results to web page after every 5 seconds automatically just like in first usecase above.

Sample output with search string as "coffee"



DETAILED SOURCE CODE.

Listing 1. Index.JSP

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
<script>
    $(document).ready(function () {
        $("#table#myTable tr:even").css("background-color", "#F4F4F8");
        $("#table#myTable tr:odd").css("background-color", "#EFF1F1");
        function poll(){
            $.getJSON("http://localhost:8080/twitterdemo/TwitterClient",
function(data){
                $.each(data, function(i) {
                    $('#myTable > tbody:first').append('<tr><td>'+data[i].data
+ '</td><td></td><td>'+data[i].twitter_user+'</td></tr>');
                });
            });
        }
    });

```

```
        });
    }

    setInterval(function(){ poll(); }, 5000);
});
</script>
<title>Twitter Client</title>
</head>
<body>
<h1>Twitter Client</h1>
<div>
<form action="TwitterClient" method="post">
<label>Search string:</label><input type="text" name="queryParam"></input>
<input type="submit" name="submit"/>
</form>
</div>
<table id=myTable>
    <tbody id=first>
    </tbody>
</table>
</body>
</html>
```

Listing 2: TwitterClient.java

```
package com.ase;
import java.io.IOException;
import java.io.PrintWriter;
import java.net.URLDecoder;
import java.net.URLEncoder;
import java.sql.SQLException;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class TwitterClient
 */
public class TwitterClient extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public TwitterClient() {
        super();
        System.out.println("TwitterClient constructor Called");
    }

    /**
     * @see Servlet#init(ServletConfig)
     */
    public void init(ServletConfig config) throws ServletException {
        System.out.println("TwitterClient init Called");
        System.out.println("Starting TwitterHTTPAgent thread");
    }
}
```

```
        TwitterHTTPAgent.getInstance().start();
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        System.out.println("TwitterClient doGet Called");
        response.setContentType("application/json");
        PrintWriter out = response.getWriter();
        try {
            System.out.println("getting latest tweets in JSON format from
Database");
            String str =
TwitterHTTPAgent.getInstance().getLatestTweets().toJSONString();
            String decodedData=URLDecoder.decode(str, "UTF-8");
            System.out.println("decodedData "+decodedData);
            out.print(decodedData);
        } catch (SQLException e) {
            e.printStackTrace();
        }
        out.flush();
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        System.out.println("TwitterClient doPost Called");
        String q = request.getParameter("queryParam");
        if(q != null)
            TwitterHTTPAgent.getInstance().setSearchString(URLEncoder.encode(q,
"UTF-8"));
        response.sendRedirect("index.jsp");
    }
}
```

Listing 3: TwitterHTTPAgent.java

```
package com.ase;

public class TwitterHTTPAgent {
    private static TwitterHTTPAgent INSTANCE = new TwitterHTTPAgent();
    private volatile static String query;
    private volatile static String refreshUrl;
    private static Connection Conn;

    private TwitterHTTPAgent() {
        query = "?q=" + "%23HANA%20OR%20%23SAP%20OR%20%23SYBASE";
        String username = "sa";
        String password = "admin123";
        String jdbcUrl = "jdbc:sybase:Tds:localhost:5000/pubs3";
```

```
        try {
            DriverManager.registerDriver((Driver)
Class.forName("com.sybase.jdbc4.jdbc.SybDriver").newInstance());
            Conn = DriverManager.getConnection(jdbcUrl, username, password);
        } catch (SQLException e) {
            e.printStackTrace();
        } catch (InstantiationException e) {
            e.printStackTrace();
        } catch (IllegalAccessException e) {
            e.printStackTrace();
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }

    public static TwitterHTTPAgent getInstance(){
        return INSTANCE;
    }

    public void setSearchString(String queryParams){

        if(queryParams == null || queryParams.length() == 0)
            query = null;
        else
            query = "?q=" + queryParams;

        refreshUrl = null;
    }

    public static void main(String[] args) throws Exception {
        getInstance().start();
    }

    public void start(){
        new Thread(new TweetCollector()).start();
    }

    private static void callTwitter() throws IOException,
        ClientProtocolException, Exception {

        System.out.println("starting TwiiterHTTPgent: callTwitter - static method
");
        DefaultHttpClient httpClient = new DefaultHttpClient();

        String twtrQuery="http://search.twitter.com/search.json" + (refreshUrl ==
null ? query : refreshUrl);

        HttpGet httpGet = new HttpGet(twtrQuery);

        System.out.println("TwitterHTTPAgent : CallTwiiter : Twitter Query
"+twtrQuery);

        HttpResponse response1 = httpClient.execute(httpGet);

        try {
            System.out.println(response1.getStatusLine());
            HttpEntity entity1 = response1.getEntity();
```

```
        String output = EntityUtils.toString(entity1);
        EntityUtils.consume(entity1);
        System.out.println(output);
        JSONParser parser = new JSONParser();
        JSONObject jsonObject = (JSONObject)parser.parse(output);
        if(jsonObject != null){
            refreshUrl = (String)jsonObject.get("refresh_url");
            System.out.println(refreshUrl);
            storeTweets(jsonObject);
        }
    } finally {
        httpGet.releaseConnection();
    }
}

//called by JSP every 5 seconds using AJAX
public JSONArray getLatestTweets() throws SQLException{

    long timeNow = System.currentTimeMillis();

    //10 seconds ago
    long timeAgo=timeNow-10000;
    java.sql.Date timeAgoStamp=new java.sql.Date(timeAgo);

    // get latest tweets stored in database.

    String readSQL = "SELECT * FROM pubs3.dbo.tweets WHERE post_date > " +
timeAgo + " ORDER BY post_date DESC";

    System.out.println("getLatestTweetsQuery -->"+readSQL);

    Statement statement = Conn.createStatement();
    ResultSet resultSet = null;
    // Result set get the result of the SQL query
    try {
        resultSet = statement.executeQuery(readSQL);
        return writeResultSetToJSON(resultSet);
    } catch (Exception e) {
        e.printStackTrace();
    }

    return null;
}

private JSONArray writeResultSetToJSON(ResultSet resultSet) throws SQLException,
UnsupportedEncodingException {
    // ResultSet is initially before the first data set
    JSONArray arr = new JSONArray();
    while (resultSet.next()) {
        JSONObject obj = new JSONObject();
        obj.put("twitter_user",
URLDecoder.decode(resultSet.getString("twitter_user"), "UTF-8"));
        obj.put("dp_url", URLDecoder.decode(resultSet.getString("dp_url"), "UTF -
8"));
        obj.put("data", URLDecoder.decode(resultSet.getString("data"), "UTF-8"));
        //obj.put("data", resultSet.getString("data"));
    }
}
```

```
        arr.add(obj);
    }
    return arr;
}

private static void storeTweets(JSONObject jsonObject) throws Exception{
    if(jsonObject.get("results") == null ||
((JSONArray)jsonObject.get("results")).size() == 0)
        return;

    PreparedStatement preparedStatement = null;

    String insertTableSQL = "INSERT INTO pubs3.dbo.tweets (twitter_user,
dp_url, data, query, post_date) VALUES " +
        "(?, ?, ?, ?, ?)";
    System.out.println("insertSQL-->" + insertTableSQL);
    preparedStatement = Conn.prepareStatement(insertTableSQL);
    JSONArray arr = ((JSONArray)jsonObject.get("results"));
    for(int i=0; i<arr.size();i++){
        JSONObject obj = (JSONObject)arr.get(i);
        preparedStatement.setString(1,
URLDecoder.decode((String)obj.get("from_user_name"), "UTF-8"));
        preparedStatement.setString(2,
URLDecoder.decode((String)obj.get("profile_image_url"), "UTF-8"));
        preparedStatement.setString(3,
URLDecoder.decode((String)obj.get("text"), "UTF-8"));
        preparedStatement.setString(4,
URLDecoder.decode((String)jsonObject.get("query"), "UTF-8"));
        preparedStatement.setLong(5, System.currentTimeMillis());

        // execute insert SQL statement
        preparedStatement.executeUpdate();
    }
    System.out.println("Record is inserted into DBUSER table!");
}

static class TweetCollector implements Runnable {

    @Override
    public void run() {
        while(query != null) {
            try {
                System.out.println("Thread run method, callTwitter
invoked");

                callTwitter();
                System.out.println("thread sleep for 20 seconds");
                Thread.sleep(20000);
            } catch (ClientProtocolException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            } catch (ParseException e) {
                e.printStackTrace();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}
```

```
    }  
  }  
}
```


© 2012 SAP AG. All rights reserved.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, SAP HANA, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company.

Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase Inc. Sybase is an SAP company.

Crossgate, m@gic EDDY, B2B 360°, and B2B 360° Services are registered trademarks of Crossgate AG in Germany and other countries. Crossgate is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

