Henry Post

ITMD 455

Lab 4: SAX XML Parsing

Contents

LoginActivity.java

```
package me.henryfbp.parser;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;


public class LoginActivity extends AppCompatActivity {

    public static final String USERNAME = "scrungo";
    public static final String PASSWORD = "chungus";
    public static final int CHANCES = 3;

    Button buttonLogin;
    FloatingActionButton floatingActionButtonLazy;

    TextView textViewProblems;
    EditText editTextUsername;
    EditText editTextPassword;

    public void populateComponents() {
        buttonLogin = findViewById(R.id.buttonLogin);
        floatingActionButtonLazy = findViewById(R.id.floatingActionButtonLazy);

        textViewProblems = findViewById(R.id.textViewProblems);
        editTextUsername = findViewById(R.id.editTextUsername);
        editTextPassword = findViewById(R.id.editTextPassword);
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {

        final Integer[] tries = {CHANCES};

        super.onCreate(savedInstanceState);
        this.setContentView(R.layout.activity_login);

        populateComponents();

        buttonLogin.setOnClickListener(new View.OnClickListener() {

            @SuppressLint("DefaultLocale")
            @Override
            public void onClick(View v) {
                if (credsValid()) {
                    Toast.makeText(getApplicationContext(), "Welcome to hell!",
Toast.LENGTH_LONG).show();

                    Intent myIntent = new Intent(getApplicationContext(), MainActivity.class);
                    startActivity(myIntent);
                } else {
                    textViewProblems.setText(String.format("Wrong login creds.\n" +
                            "Try pressing the floating action button.\n" +
                            "%d chances left.", tries[0]));
```

```
                tries[0] = tries[0] - 1;
            }

            if (tries[0] <= 0) {
                finish();
            }
        }
    });

    floatingActionButtonLazy.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            populateLogin();
        }
    });
}

private void populateLogin() {
    editTextUsername.setText(USERNAME);
    editTextPassword.setText(PASSWORD);
}

private String getUsername() {
    return editTextUsername.getText().toString();
}

private String getPassword() {
    return editTextPassword.getText().toString();
}

public boolean credsValid() {
    return (getUsername().equals(USERNAME) && getPassword().equals(PASSWORD));
}

}
```

MainActivity.java

```
package me.henryfbp.parser;

import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import org.xml.sax.InputSource;
import org.xml.sax.XMLReader;

import java.net.URL;

import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;

public class MainActivity extends AppCompatActivity {

    public static final String DATA_SOURCE = "http://www.papademas.net:81/cd_catalog3.xml";
    XMLGettersSetters data;

    /**
     * Called when the activity is first created.
     */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        new BackgroundTask().execute();
    }

    private void saxParser() {

        try {
            /* Create a new instance of the SAX parser */
            SAXParserFactory saxPF = SAXParserFactory.newInstance();
            SAXParser saxP = saxPF.newSAXParser();
            XMLReader xmlR = saxP.getXMLReader();

            // URL of the XML
            URL url = new URL(DATA_SOURCE);

            /* Create the Handler to handle each of the XML tags */
            XMLHandler myXMLHandler = new XMLHandler();
            xmlR.setContentHandler(myXMLHandler);
            xmlR.parse(new InputSource(url.openStream()));

        } catch (Exception e) {
            System.out.println(e);
        }

    }

    private void viewSaxData() {
        /* Get the view of the layout within the main layout, so that we can
         * add TextViews.
         */
        View layout = findViewById(R.id.layout);
        ViewGroup viewGroupLayout = (ViewGroup) layout;
```

```
/*Create TextView Arrays to add the retrieved data to */

TextView title[];
TextView artist[];
TextView country[];
TextView company[];
TextView price[];
TextView year[];
TextView cd[];

data = XMLHandler.data;

/* Make the TextView array length the size of the arraylist */
title = new TextView[data.getTitle().size()];
artist = new TextView[data.getArtist().size()];
country = new TextView[data.getCountry().size()];
company = new TextView[data.getCompany().size()];
price = new TextView[data.getPrice().size()];
year = new TextView[data.getYear().size()];
cd = new TextView[data.getCd().size()];


/**
 * Run a for loop to set All the TextViews with text until
 * the given size of the arraylist is reached.
 **/
for (int i = 0; i < data.getTitle().size(); i++) {


    title[i] = new TextView(this);
    title[i].setText(String.format("Title = %s", data.getTitle().get(i)));

    artist[i] = new TextView(this);
    artist[i].setText(String.format("Artist = %s", data.getArtist().get(i)));

    country[i] = new TextView(this);
    country[i].setText(String.format("Country = %s", data.getCountry().get(i)));

    company[i] = new TextView(this);
    company[i].setText(String.format("Company = %s", data.getCompany().get(i)));

    price[i] = new TextView(this);
    price[i].setText(String.format("Price = %s", data.getPrice().get(i)));

    year[i] = new TextView(this);
    year[i].setText(String.format("Year = %s", data.getYear().get(i)));

    cd[i] = new TextView(this);
    cd[i].setText(String.format("CD? %s", data.getCd().get(i)));

    if (data.getCd().get(i).equalsIgnoreCase("yes")) { // Only add ones which are sold out.
        viewGroupLayout.addView(title[i]);
        viewGroupLayout.addView(artist[i]);
        viewGroupLayout.addView(country[i]);
        viewGroupLayout.addView(company[i]);
        viewGroupLayout.addView(price[i]);
        viewGroupLayout.addView(year[i]);
        viewGroupLayout.addView(cd[i]);
    }

}
```

```
        setContentView(layout);
    }

    public class BackgroundTask extends AsyncTask<Void, Integer, Void> {

        @Override
        protected void onPreExecute() {
            super.onPreExecute();
        }

        @Override
        protected void onProgressUpdate(Integer... values) {
            super.onProgressUpdate(values);
        }

        @Override
        protected void onPostExecute(Void aVoid) {
            super.onPostExecute(aVoid);
            viewSaxData();
        }

        @Override
        protected Void doInBackground(Void... params) {

            try {
                synchronized (this) {
                    saxParser();
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
            return null;
        }
    }
}
```

XMLGettersSetters.java

```
package me.henryfbp.parser;

import android.util.Log;

import java.util.ArrayList;


public class XMLGettersSetters {
    /*This class contains all getter and setter methods to set and retrieve data.*/


    private ArrayList<String> title = new ArrayList<>();
    private ArrayList<String> artist = new ArrayList<>();
    private ArrayList<String> country = new ArrayList<>();
    private ArrayList<String> company = new ArrayList<>();
    private ArrayList<String> price = new ArrayList<>();
    private ArrayList<String> year = new ArrayList<>();
    private ArrayList<String> cd = new ArrayList<>();

    public ArrayList<String> getCompany() {
        return company;
    }

    public void addCompany(String company) {
        this.company.add(company);
        Log.i("This is the company:", company);
    }

    public ArrayList<String> getPrice() {
        return price;
    }

    public void addPrice(String price) {
        this.price.add(price);
        Log.i("This is the price:", price);
    }

    public ArrayList<String> getYear() {
        return year;
    }

    public void addYear(String year) {
        this.year.add(year);
        Log.i("This is the year:", year);
    }

    public ArrayList<String> getTitle() {
        return title;
    }

    public void addTitle(String title) {
        this.title.add(title);
        Log.i("This is the title:", title);
    }

    public ArrayList<String> getArtist() {
        return artist;
    }

    public void addArtist(String artist) {
        this.artist.add(artist);
```

```
        Log.i("This is the artist:", artist);
    }

    public ArrayList<String> getCountry() {
        return country;
    }

    public void addCountry(String country) {
        this.country.add(country);
        Log.i("This is the country:", country);
    }

    public ArrayList<String> getCd() {
        return cd;
    }

    public void addCd(String cd) {
        Log.i("Sold out?: ", cd);
        this.cd.add(cd);
    }
}
```

XMLHandler.java

```
package me.henryfbp.parser;

import android.util.Log;

import org.xml.sax.Attributes;
import org.xml.sax.SAXException;
import org.xml.sax.helpers.DefaultHandler;

public class XMLHandler extends DefaultHandler {
    public static XMLGettersSetters data = null;
    String elementValue = null;
    Boolean elementOn = false;

    public static XMLGettersSetters getXMLData() {
        return data;
    }


    public static void setXMLData(XMLGettersSetters data) {
        XMLHandler.data = data;
    }

    /**
     * This will be called when the tags of the XML starts.
     **/
    @Override
    public void startElement(String uri, String localName, String qName,
                             Attributes attributes) throws SAXException {

        elementOn = true;

        if (localName.equals("CATALOG")) {
            data = new XMLGettersSetters();
        } else if (localName.equals("CD")) {
            try {
                data.addCd(attributes.getValue("attr"));
            } catch (Exception e) {
                Log.i("err on handler   ", e.getMessage());
            }
        }
    }

    /**
     * This will be called when the tags of the XML end.
     **/
    @Override
    public void endElement(String uri, String localName, String qName)
            throws SAXException {

        elementOn = false;

        /**
         * Sets the values after retrieving the values from the XML tags
         * */
        if (localName.equalsIgnoreCase("title")) {
            data.addTitle(elementValue);
        } else if (localName.equalsIgnoreCase("artist")) {
            data.addArtist(elementValue);
        } else if (localName.equalsIgnoreCase("country")) {
            data.addCountry(elementValue);
        } else if (localName.equalsIgnoreCase("company")) {
```

```
            data.addCompany(elementValue);
        } else if (localName.equalsIgnoreCase("price")) {
            data.addPrice(elementValue);
        } else if (localName.equalsIgnoreCase("year")) {
            data.addYear(elementValue);
        }
    }

    /**
     * This is called to get the tags value
     **/
    @Override
    public void characters(char[] ch, int start, int length)
            throws SAXException {

        if (elementOn) {
            elementValue = new String(ch, start, length);
            elementOn = false;
        }

    }
}
```