Henry Post

ITMD 455

Lab 2: Temperature Converter Part Deux

Contents

Lab 2

MainActivity.java

```
package me.henryfbp.myapplication;

import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.view.ViewStub;
import android.widget.Button;
import android.widget.ListView;
import android.widget.SeekBar;
import android.widget.SeekBar.OnSeekBarChangeListener;
import android.widget.TextView;
import android.widget.Toast;

import java.math.BigDecimal;

import me.henryfbp.library.TemperatureSolver;


public class MainActivity extends AppCompatActivity {

    TemperatureSolver ts = new TemperatureSolver();
    SeekBar seekBar; //declare seekbar object
    TextView textView;
    //declare member variables for SeekBar
    int discrete = 0;
    int start = 50;
    int start_position = 50; //progress tracker
    int temp = 0;
    //declare objects for ViewStub
    ViewStub stub;
    Button button;
    ListView lv; //declare Listview object

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Intent intent = getIntent();
        Bundle extras = intent.getExtras();

        if (extras != null && extras.containsKey("temperature")) {
            TemperatureDate temperature = (TemperatureDate) extras.get("temperature");
            Log.i("GETTIN INTENT", temperature.toString());
            start_position = temperature.temperature.intValue() + start;
        } else {
            Log.i("GETTIN INTENT", "No 'temperature' from Intent's Extras. Must be the first date
we're in this Activity.");
        }

        textView = findViewById(R.id.textview);
        textView.setText("0c");
        //set default view
        seekBar = findViewById(R.id.seekbar);

        stub = findViewById(R.id.viewStub);
        View inflated = stub.inflate();
```

```
        stub.setVisibility(View.INVISIBLE);

        button = findViewById(R.id.button);
        button.setOnClickListener(new Button.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(v.getContext(), TempPicker.class);

                startActivityForResult(i, 0);
            }
        });

        //create event handler for SeekBar
        seekBar.setOnSeekBarChangeListener(new OnSeekBarChangeListener() {
            @Override
            public void onStopTrackingTouch(SeekBar seekBar) {
                Toast.makeText(getBaseContext(), String.format("%sf", String.valueOf(discrete)),
Toast.LENGTH_SHORT).show();
            }

            @Override
            public void onStartTrackingTouch(SeekBar seekBar) {
            }

            @Override
            public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
                // To convert progress passed as discrete (Fahrenheit) value
                temp = progress - start;
                discrete = ts.solve("celsius", "fahrenheit", new BigDecimal(temp)).intValue();
                textView.setText(String.format("%dc", temp));
            }
        });

        seekBar.setProgress(start_position);


    } //end onCreate method
}
```

TemperatureDate.java

```
package me.henryfbp.myapplication;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.Serializable;
import java.math.BigDecimal;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

import me.henryfbp.library.TemperatureSolver;

public class TemperatureDate implements Serializable {

    public static final TemperatureSolver solver = new TemperatureSolver();

    public Double temperature;
    public Date date;

    public TemperatureDate(Date date, Double temperature) {
        this.date = date;
        this.temperature = temperature;
    }

    public static List<TemperatureDate> fromJSONList(JSONArray j) throws JSONException {
        List<TemperatureDate> ret = new ArrayList<>();

        for (Integer i = 0; i < j.length(); i++) {
            JSONObject o = j.getJSONObject(i);

            Date date = new Date((long) o.getInt("dt") * 1000);
            Double temperature = ((double) o.getJSONObject("main").getInt("temp"));
            temperature = solver.solve("kelvin", "celsius", new
BigDecimal(temperature)).doubleValue();

            ret.add(new TemperatureDate(date, temperature));
        }

        return ret;
    }

    @Override
    public String toString() {
        return String.format("%s: %.2fc", new SimpleDateFormat("EEE, MMM d h:mm a").format(date),
temperature);
    }
}
```

TemperaturePicker.java

```
package me.henryfbp.myapplication;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;

import com.android.volley.Cache;
import com.android.volley.Network;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.BasicNetwork;
import com.android.volley.toolbox.DiskBasedCache;
import com.android.volley.toolbox.HurlStack;
import com.android.volley.toolbox.StringRequest;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;

import me.henryfbp.library.TemperatureJSON;

public class TempPicker extends Activity {

    ListView listView;

    ArrayAdapter listAdapter;

    List<TemperatureDate> wkTemps = new ArrayList<>();

    @Override
    public void onCreate(Bundle savedInstanceState) {

        RequestQueue mRequestQueue;

        // Instantiate the cache
        Cache cache = new DiskBasedCache(getCacheDir(), 1024 * 1024); // 1MB cap

        // Set up the network to use HttpURLConnection as the HTTP client.
        Network network = new BasicNetwork(new HurlStack());

        // Instantiate the RequestQueue with the cache and network.
        mRequestQueue = new RequestQueue(cache, network);

        // Start the queue
        mRequestQueue.start();

        String url = TemperatureJSON.form_week_request("Chicago");

        Log.i("JSON BRO", url);

        StringRequest stringRequest = new StringRequest(url, new Response.Listener<String>() {
            @Override
```

```
            public void onResponse(String response) {
                Log.i("JSON BRO", response);

                try {
                    JSONObject jsonObject = new JSONObject(response);

                    JSONArray array = jsonObject.getJSONArray("list");

                    List<TemperatureDate> temps = TemperatureDate.fromJSONList(array);

                    wkTemps.addAll(temps);

                    synchronized (listAdapter) {
                        listAdapter.notifyDataSetChanged();
                    }


                } catch (JSONException e) {
                    e.printStackTrace();
                }

            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                Log.i("JSON BRO", error.toString());
                error.printStackTrace();
            }
        });

        mRequestQueue.add(stringRequest); //add the request


        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_temp_picker);

        listAdapter = new ArrayAdapter(this, R.layout.simplerow, wkTemps); // create arrayAdapter
        listView = findViewById(R.id.listView);

        listView.setAdapter(listAdapter); // Populate listView with arrayAdapter's content

        listView.setOnItemClickListener(new AdapterView.OnItemClickListener() { //Someone clicks a
single item
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                TemperatureDate item = (TemperatureDate) parent.getItemAtPosition(position);

                Log.i("U KLIK ME?", item.toString());

                Intent i = new Intent(getApplicationContext(), MainActivity.class);

                i.putExtra("temperature", item); // pass temperature

                startActivity(i); //start previous activity
            }
        });
    }


}
```

Library

TemperatureJSON.java

```
package me.henryfbp.library;

import com.google.common.collect.ImmutableMap;

import java.util.Map;

public class TemperatureJSON {

    /*
    OpenWeatherMap API Key.

    I am aware embedding API keys is insecure, and to that, I say:
    You wouldn't download a cloud, would you?
    */
    private static final String API_KEY = "b2d4abd7657a8072e535576cfd13b3e6"; // ...would you?

    private static final String URL = "https://samples.openweathermap.org/data/2.5/forecast";

    public static String form_week_request(String location) {
        return form_request(ImmutableMap.of("q", location));
    }

    public static String form_request(Map<String, String> args) {

        String s = URL + "?appid=" + API_KEY;

        s = HLib.apply_json_params(s, args);

        return s;
    }

}
```

HLib.java

```java
package me.henryfbp.library;

import android.graphics.Color;

import java.util.Map;
import java.util.Random;

public class HLib {


    public static void println(String arg)
    {
        System.out.println(arg);
    }

    public static void print(String args)
    {
        System.out.print(args);
    }

    public static Color randomColor() {
        Random r = new Random();
        return Color.valueOf(r.nextFloat(), r.nextFloat(), r.nextFloat());
    }


    /*
     * Adapted from https://stackoverflow.com/a/17544748/4262535.
     *
     * Mixes two colors together.
     */
    public static Color mixColors(Color c1, Color c2, Float percent) {

        if (percent < 0f) {
            throw new IllegalArgumentException(percent.toString() + " < 0!");
        }
        if (percent > 1f) {
            throw new IllegalArgumentException(percent.toString() + " > 1!");
        }

        float inv_percent = 1.0f - percent;

        float r = (c1.red() * percent +
                c2.red() * inv_percent);

        float g = (c1.green() * percent +
                c2.green() * inv_percent);

        float b = (c1.blue() * percent +
                c2.blue() * inv_percent);

        return Color.valueOf(r, g, b);
    }

    public static Color mixColors(int c1, int c2, float percent) {
        return mixColors(Color.valueOf(c1), Color.valueOf(c2), percent);
    }

    public static String apply_json_params(String str, Map<String, String> args){

        StringBuilder s = new StringBuilder(str);
```

```
        for (Map.Entry e : args.entrySet()) {
            String k = (String) e.getKey();
            String v = (String) e.getValue();

            s.append("&").append(k).append("=").append(v);
        }

        return s.toString();

    }


}
```