Mobile Computing

Working with JASON

Introduction

JSON stands for JavaScript Object Notation. JSON is the best alternative to the XML which is the most typical data exchange format used in Android. JSON objects can be defined easier than an XML document.

Android provides four different classes to manipulate JSON data. These classes are **JSONArray, JSONObject, JSONStringer and JSONTokenizer.**

JSONArray

A JSONArray is an ordered sequence of values. Its external text form is a string wrapped in square brackets with commas separating the values. The internal form is an object having get and opt methods for accessing the values by index, and put methods for adding or replacing values. The values can be any of these types: Boolean, JSONArray, JSONObject, Number, String, or the JSONObject.NULL object.

JSONObject

A JSONObject is an unordered collection of name/value pairs. Its external form is a string wrapped in curly braces with colons between the names and values, and commas between the values and names. The internal form is an object having get and opt methods for accessing the values by name, and put methods for adding or replacing values by name. The values can be any of these types: Boolean, JSONArray, JSONObject, Number, String, or the JSONObject.NULL object.

JSONStringer

JSONStringer provides a quick and convenient way of producing JSON text. The texts produced strictly conform to JSON syntax rules. No whitespace is added, so the results are ready for transmission or storage. Each instance of JSONStringer can produce one JSON text.

JSONTokenizer

A JSONTokener takes a source string and extracts characters and tokens from it. It is used by the JSONObject and JSONArray constructors to parse JSON source strings.

Creating JSON Object

Sample JSON Object

Array ([): In a JSON file, square bracket ([) represents a JSON array

Objects ({): In a JSON file, curly bracket ({) represents a JSON object

Key: A JSON object contains a key that is just a string. Pairs of key/value make up a JSON object

Value: Each key has a value that could be string, integer or double etc.

```
{
"sys":
   {
      "country":"Sri Lanka",
      "sunrise":1381107600,
      "sunset":1381149000
   },
"weather":[
   {
      "id":615,
      "main":"Smoke",
      "description":"smoke",
      "icon":"50n"
   }
],
"main":
   {
      "temp":300.15,
      "pressure":1000,
   }
}
```

Parsing JSON

```
String in;
… //JSON String should store in the variable “in”
JSONObject reader = new JSONObject(in);

JSONObject sys  = reader.getJSONObject("sys");
country = sys.getString("country");

JSONObject main  = reader.getJSONObject("main");
temperature = main.getString("temp");
```

Any JSON object contains the following methods.

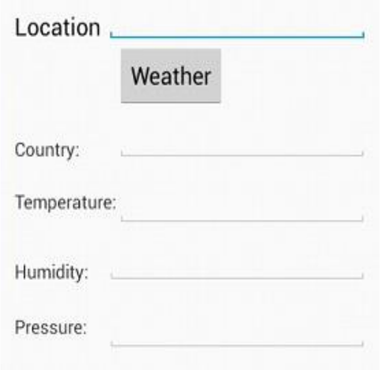
get(String name): Returns the value but in the form of Object type
getBoolean(String name): Returns the boolean value specified by the key
getDouble(String name): Returns the double value specified by the key

getInt(String name): Returns the integer value specified by the key
getLong(String name): Returns the long value specified by the key
length(): Returns the number of name/value mappings in this object..
names(): Returns an array containing the string names in this object.

Example

The following application will fetch the weather information from open weather API.

First create the following Layout.



Add the following code to the MainActivity java file. The following method should be called when the button “weather” is clicked

API: 92512790523a054a101f15bd5f349039

```
private String url1 =
"http://api.openweathermap.org/data/2.5/weather?q=Colombo&APPID=92512790523a054a101f15
bd5f349039";
private EditText location,country,temperature,humidity,pressure;
private HandleJSON obj;

@Override
protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      location = (EditText)findViewById(R.id.editText1);
      country = (EditText)findViewById(R.id.editText2);
      temperature = (EditText)findViewById(R.id.editText3);
      humidity = (EditText)findViewById(R.id.editText4);
      pressure = (EditText)findViewById(R.id.editText5);
   }

public void open(View view){
      String url = location.getText().toString();
      String finalUrl = url1 + url;
      country.setText(finalUrl);
      obj = new HandleJSON(finalUrl);
      obj.fetchJSON();

      while(obj.parsingComplete); //wait until the data arrived and passing complete
      country.setText(obj.getCountry());
      temperature.setText(obj.getTemperature());
      humidity.setText(obj.getHumidity());
      pressure.setText(obj.getPressure());

   }
```

In order to handle the JSON object create the following class. It processes a GET request from open weather API and process the response to parse the data to JSON.

```
public class HandleJSON {
   private String country = "county";
   private String temperature = "temperature";
   private String humidity = "humidity";
   private String pressure = "pressure";
   private String urlString = null;

   public volatile boolean parsingComplete = true;
   public HandleJSON(String url){
      this.urlString = url;
   }
   public String getCountry(){
      return country;
   }
   public String getTemperature(){
      return temperature;
   }
   public String getHumidity(){
      return humidity;
   }
   public String getPressure(){
      return pressure;
   }

   @SuppressLint("NewApi")
   public void readAndParseJSON(String in) {
      try {
         JSONObject reader = new JSONObject(in);

         JSONObject sys  = reader.getJSONObject("sys");
         country = sys.getString("country");

         JSONObject main  = reader.getJSONObject("main");
         temperature = main.getString("temp");

         pressure = main.getString("pressure");
         humidity = main.getString("humidity");



        } catch (Exception e) {
           // TODO Auto-generated catch block
           e.printStackTrace();
        }
        parsingComplete = false;


   }
   public void fetchJSON(){
      Thread thread = new Thread(new Runnable(){
         @Override
         public void run() {
         try {
            URL url = new URL(urlString);
```

```
              HttpURLConnection conn = (HttpURLConnection) url.openConnection();
              conn.setReadTimeout(10000 /* milliseconds */);
              conn.setConnectTimeout(15000 /* milliseconds */);
              conn.setRequestMethod("GET");
              conn.setDoInput(true);
              // Starts the query
              conn.connect();
           InputStream stream = conn.getInputStream();

       String data = convertStreamToString(stream);

       readAndParseJSON(data);
         stream.close();

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

       thread.start();
   }
   static String convertStreamToString(java.io.InputStream is) {
      java.util.Scanner s = new java.util.Scanner(is).useDelimiter("\\A");
      return s.hasNext() ? s.next() : "";
   }
}
```

Permission to The Application

```
<uses-permission android:name="android.permission.INTERNET"/>
```