Android Application Development

Reading JSON from the Web

Dr. Frank Coyle
(drC)

JSON Basics

- **JSON** is an **acronym** for **JavaScript Object Notation**, a lightweight text-based open standard designed for **human-readable** data interchange.
- Derived from the **JavaScript** scripting language for representing simple **data structures** and **associative arrays**, called objects.
- Despite its relationship to JavaScript, it is language-independent, with parsers available for most languages.
- The official **Internet media type** for JSON is application/json. The JSON filename extension is .json.
- The JSON format is often used for **serializing** and transmitting structured data over a network connection.
- It is primarily used to transmit data between a server and web application, serving as an alternative to **XML**.
JSON Data types

- **Number** *(double precision floating-point format)*
- **String** *(double-quoted Unicode with backslash escaping)*
- **Boolean** *(true or false)*
- **Array** *(an ordered sequence of values, comma-separated and enclosed in square brackets. The values don't need to have the same type.)*
- **Object** *(an unordered collection of key:value pairs, comma-separated and enclosed in curly braces; the key must be a string)*
- **null**

### JSON Example

```
{
    "firstName": "John",
    "lastName": "Smith",
    "age": 35,
    "address": {
        "streetAddress": "21 2nd Street",
        "city": "New York",
        "state": "NY",
        "postalCode": "10001"
    },
    "phoneNumber": [
        {
            "type": "home",
            "number": "111 222 3333"
        },
        {
            "type": "Fax",
            "number": "555-666-7777"
        }
    ]
}
```
• Since JSON is a subset of JavaScript it is possible (but not recommended) to parse the JSON text into an object by invoking JavaScript's `eval()` function.

• For example, if the above JSON data is contained within a JavaScript string variable `contact`, one could use it to create the JavaScript object `p` like so:

```javascript
var p = eval("" + contact + ");
```

---

### Use in Ajax

• JavaScript code using an `XMLHttpRequest` to request an object in JSON format from the server.

• The server-side programming is omitted; it has to be set up to respond to requests at url with a JSON-formatted string.

```javascript
var my_JSON_object = {};
var http_request = new XMLHttpRequest();
http_request.open("GET", url, true);
http_request.onreadystatechange = function () {
  var done = 4, ok = 200;
  if (http_request.readyState == done && http_request.status == ok) {
    my_JSON_object = JSON.parse(http_request.responseText);
  }
};
http_request.send(null);
```

A new function, `JSON.parse()`, was developed as a safer alternative to `eval`. It is specifically intended to process JSON data and not JavaScript.
JSON Twitter Feeds using ListActivity

```java
public class HomeActivity extends ListActivity {

    /** Called when the activity is first created. */
    @SuppressWarnings({"rawtypes", "unchecked" })

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

        setListAdapter(new ArrayAdapter(this,
                android.R.layout.simple_list_item_1, 
                this.fetchTwitterPublicTimeline()));
    }
}
```

To see JSON twitter as XML see:
http://twitter.com/statuses/public_timeline.xml

Getting JSON Data

```java
public ArrayList<String> fetchTwitterPublicTimeline() {

    ArrayList<String> listItems = new ArrayList<String>();

    try {
        URL twitter = new URL("http://twitter.com/statuses/public_timeline.json");
        URLConnection tc = twitter.openConnection();
        BufferedReader in = new BufferedReader(new InputStreamReader(tc.getInputStream()));

        String line;
        while ((line = in.readLine()) != null) {
            JSONArray ja = new JSONArray(line);

            for (int i = 0; i < ja.length(); i++) {
                JSONObject jo = (JSONObject) ja.get(i);
                listItems.add(jo.getString("text"));
            }
        }
    }
    return listItems;
}
```

Here we grab a "text" attribute/property of single JSON object.
Summary

• When executing time-intensive code, do not execute it in main UI thread (default)
• Execute in a background thread
• Android provides AsyncTask class to help with thread creation and communication with UI thread
• Create subclass of AsyncTask and overwrite:
  – doInBackground(…)
  – onPostExecute(..)