

CSCI 428/628 - Advanced Web Programming - Assignment: Reading XML from a Server

Overview

In this assignment you will write an app to read a XML file from a remote server and display the file contents in an easily readable and labeled manner. Much of this code will be re-used from the JSON assignment, but obviously the file format and parsing will be different. The output will appear essentially identical.

The XML file follows the format shown here:

```
<?xml version="1.0" encoding="UTF-8"?>
<client region = "Midwest">
```

```
  <person
    name = "Mary Jones"
    profession = "Programmer"
    dob = "April 2, 1970">
    <child
      name = "Tom">
    </child>
    <child
      name = "Sue">
    </child>
    <child
      name = "Mary">
    </child>
  </person>
```

```
  <person
    name = "Dave Young"
    profession = "SysAdmin"
    dob = "June 12, 1960">
  </person>
```

```
</client>
```

The XML file can be found at "http://www.seasite.niu.edu/cs680Android/XML/Client_list_xml.txt" and will be served by that site's web server, so it can be read via an http Get command. (It can be displayed using a web browser if you are curious to see the exact whole file.) Since the Get command will result in a data stream that will be read by your program's code, and since these read calls will block if the next chunk of data from the server is delayed, your code to retrieve the XML data should execute in a separate Thread of execution.

Again, there could be 0 - many children for each client. There will be exactly 3 named attributes for each client, but the order and name of the attributes is not fixed. You should include the attribute names, but you do not have to put them in any particular order.

Platform Implementation Notes

Android

Reading the file from the server will be identical to reading the JSON file, but the data will be stored in a String and will be returned as an String typecast as an Object. Use an XmlPullParser to extract data names and values from the file and display the data as you did in the JSON assignment.

As before, your code should write informative error messages to LogCat in the event that there is a parsing error on the JSON file or other error condition.

iOS

- Use Grand Central Dispatch similar to what you did in assignment #1.
- Connect to the server, download the data and parse the XML-formatted data in the background thread.
- Use NSURL to create an instance and set it to the URL string.
- Use NSXMLParser to send a request to the server, download and parse the data.
- If error occurs, display "Failed to parse" message.
- Let the main UI thread display the parsed data in two views, the same way you did in assignment #2 (JSON parser) using UITableView and DetailView

Windows Phone

(to be added)