Android
Internet Feeders

Victor Matos
Cleveland State University

Notes are based on:
Android Developers
The image illustrates Year 2007 Internet Density. Colored dots represent number of IP addresses per dot-area.

Communicating Via Internet

The image illustrates 2007 Internet density. Colored dots represent number of IP addresses per dot-area.

### 2007 - Breakdown by Geographic Area

<table>
<thead>
<tr>
<th>Geographic area</th>
<th>Number of IP addresses</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Africa</td>
<td>40241664</td>
<td>1.519%</td>
</tr>
<tr>
<td>Antarctica</td>
<td>15620</td>
<td>0.001%</td>
</tr>
<tr>
<td>Asia</td>
<td>371297015</td>
<td>14.015%</td>
</tr>
<tr>
<td>Caribbean</td>
<td>1681866 0.063%</td>
<td>0.097%</td>
</tr>
<tr>
<td>Central America</td>
<td>2557340 0.097%</td>
<td>21.510%</td>
</tr>
<tr>
<td>Europe</td>
<td>569838903</td>
<td>55.932%</td>
</tr>
<tr>
<td>Middle East</td>
<td>12011131</td>
<td>0.453%</td>
</tr>
<tr>
<td>North America</td>
<td>1481754661</td>
<td>2.885%</td>
</tr>
<tr>
<td>Oceania</td>
<td>76417711</td>
<td>3.525%</td>
</tr>
<tr>
<td>South America</td>
<td>93409304</td>
<td>1.519%</td>
</tr>
</tbody>
</table>

Source: [www.ipligence.com](http://www.ipligence.com)
USA & Worldwide Growth of Cellular Telephony

Study on Information & Communication Technology (ICT).
World Bank Organization 2009

CONCLUSION
1. Countries that have taken steps to create a competitive market environment for ICT generally have a larger share of people using ICT services than those that have not.
2. One important outcome of competition is that it lowers prices for ICT services.
3. But while prices are falling rapidly ICT services are still unaffordable for many people in low-income countries.
4. As a result a large share of the population in these countries has yet to realize the potential of ICT for economic and social development.
5. **Among ICT services voice communications particularly mobile telephony has led the way.**
6. Access to new services such as broadband remains limited in developing countries.
7. Yet the expansion of broadband networks plays a catalytic role in the development of trade and e-government.
8. In some developing countries trade in ICT goods and services has sparked export-led growth and job creation.
9. **ICT applications are also transforming how governments deliver public services to citizens and businesses.**

Source:
Key Trends in ICT Development by David A. Cieslikowski Naomi J. Halewood Kaoru Kimura and Christine Zhen-Wei Qiang
Background: RSS

RSS (most commonly translated as Really Simple Syndication but sometimes Rich Site Summary) is a family of web feed formats used to publish frequently updated works—such as blog entries, news headlines, weather reports, audio and video—in a standardized format.

An RSS document (which is called a web feed or channel) includes full or summarized text (plus metadata), publishing dates and authorship.

Notes

• Content distributors expose web feeds allowing users to subscribe to them.

• Making a collection of web feeds accessible in one spot is known as aggregation which is performed by an Internet aggregator.
Background: **Web Feeds**

Web feeds allow (client) software programs to check for summary updates published on a website.

- First version of RSS was created by Netscape around 1999.

- A typical *news feed (or channel)* contains entries which may be:
  - headlines,
  - full-text articles excerpts,
  - summaries,
  - Thumbnails, and/or
  - links to content on a website along with various metadata

- The *Atom Syndication Format* and RSS2.0 are common XML standards used to organize, create and update web feeds (these formats have been adopted by Google, Yahoo!, Apple/iTunes, CNN, NY Times, ...).
Background: **Web Feeds**

Common child elements of Channel and Item

<table>
<thead>
<tr>
<th>Elements</th>
<th>Description</th>
<th>Type</th>
<th># allowed</th>
</tr>
</thead>
<tbody>
<tr>
<td>LastMod</td>
<td>Last modified date for this web page</td>
<td>ISO 8601:1988 Date</td>
<td>0 or 1</td>
</tr>
<tr>
<td>Title</td>
<td>Title</td>
<td>String</td>
<td>0 or 1</td>
</tr>
<tr>
<td>Abstract</td>
<td>Short description summarizing the article (200 characters or less recommended)</td>
<td>String</td>
<td>0 or 1</td>
</tr>
<tr>
<td>Author</td>
<td>Author</td>
<td>String</td>
<td>Any</td>
</tr>
<tr>
<td>Publisher</td>
<td>Publisher</td>
<td>String</td>
<td>Any</td>
</tr>
<tr>
<td>Copyright</td>
<td>Copyright</td>
<td>String</td>
<td>0 or 1</td>
</tr>
<tr>
<td>PublicationDate</td>
<td>Publication Date</td>
<td>String</td>
<td>0 or 1</td>
</tr>
<tr>
<td>Logo</td>
<td>Visual Logo for channel</td>
<td>Logo element</td>
<td>Any</td>
</tr>
<tr>
<td>Keywords</td>
<td>Comma delimited keywords that match this channel</td>
<td>String</td>
<td>Any</td>
</tr>
<tr>
<td>Category</td>
<td>A category to which this web page belongs in. The string value is a URI to a CategoryDef element.</td>
<td>Category element</td>
<td>Any</td>
</tr>
<tr>
<td>Ratings</td>
<td>Rating of the channel by one or more ratings services. (String found in PICS label meta tag.)</td>
<td>String</td>
<td>Any</td>
</tr>
<tr>
<td>Schedule</td>
<td>Schedule for keeping channel up to date</td>
<td>Schedule element</td>
<td>0 or 1</td>
</tr>
<tr>
<td>UserSchedule</td>
<td>Reference to a client/user specified schedule</td>
<td>UserSchedule element</td>
<td>0 or 1</td>
</tr>
</tbody>
</table>

Reference: [http://www.w3.org/TR/NOTE-CDFsubmit.html](http://www.w3.org/TR/NOTE-CDFsubmit.html)
Background: **Web Feeds**

**Elements of <item>**

A channel may contain any number of <item>s. An item may represent a "story" -- much like a story in a newspaper or magazine.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>title</td>
<td>The title of the item.</td>
</tr>
<tr>
<td>link</td>
<td>The URL of the item.</td>
</tr>
<tr>
<td>description</td>
<td>The item synopsis.</td>
</tr>
<tr>
<td>author</td>
<td>Email address of the author of the item.</td>
</tr>
<tr>
<td>category</td>
<td>Includes the item in one or more categories.</td>
</tr>
<tr>
<td>comments</td>
<td>URL of a page for comments relating to the item.</td>
</tr>
<tr>
<td>enclosure</td>
<td>Describes a media object that is attached to the item.</td>
</tr>
<tr>
<td>guid</td>
<td>A string that uniquely identifies the item.</td>
</tr>
<tr>
<td>pubDate</td>
<td>Indicates when the item was published.</td>
</tr>
<tr>
<td>source</td>
<td>The RSS channel that the item came from.</td>
</tr>
</tbody>
</table>

Reference: [http://www.w3.org/TR/NOTE-CDFsubmit.html](http://www.w3.org/TR/NOTE-CDFsubmit.html)
Background: Web Feeds

Example:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
  <channel>
    <title>RSS Title</title>
    <description>This is an example of an RSS feed</description>
    <link>http://www.someexamplerssdomain.com/main.html</link>
    <lastBuildDate>Mon, 06 Sep 2010 00:01:00 +0000</lastBuildDate>
    <pubDate>Mon, 06 Sep 2009 16:45:00 +0000</pubDate>
    
    <item>
      <title>Example entry</title>
      <description>Here is some text containing an interesting description of the thing to be described.</description>
      <link>http://www.wikipedia.org/</link>
      <guid>unique string per item</guid>
      <pubDate>Mon, 06 Sep 2009 16:45:00 +0000</pubDate>
    </item>
  </channel>
</rss>
```

Source: http://en.wikipedia.org/wiki/RSS
Internet Feeders

**RSS Sources**

**US weather:**
http://www.weather.gov/view/national.php?map=on

**The Weather Channel**
http://rss.weather.com/weather/rss/local/44114?cm_ven=LWO&cm_cat=rss&par=LWO_rss

**World weather**
http://www.rssweather.com/dir

**News**
http://www.npr.org/rss/
http://www.cnn.com/services/rss/
http://news.bbc.co.uk/2/hi/help/3223484.stm

**Money Exchange**

**Entertainment**
http://www.nbclosangeles.com/rss/
http://www.movies.com/rss/

**RSS Aggregator**
http://www.rss-network.com/

**Corporate**
http://home3.americanexpress.com/corp/rss/
http://www.aa.com/i18n/urls/rss.jsp
How do RSS feeds look like when using a browser?

http://www.npr.org/rss/
XML Version of the NPR RSS Feed (1/3 just a fragment!)

<?xml version="1.0"?>
<?xml-stylesheet title="XSL_formatting" type="text/xsl" href="/include/xsl/rss.xsl"?>

xmlns:itunes="http://www.itunes.com/dtds/podcast-1.0.dtd"
xmlns:content="http://purl.org/rss/1.0/modules/content/" version="2.0">

<channel>

<title>NPR Topics: Science</title>

<link>http://www.npr.org/templates/story/story.php?storyId=1007&amp;ft=1&amp;f=1007</link>

<description>The latest health and science news. Updates on medicine, healthy living, nutrition, drugs, diet, and advances in science and technology. Subscribe to the Health & Science podcast.</description>

<copyright>Copyright 2009 NPR - For Personal Use Only</copyright>
<generator>NPR API RSS Generator 0.93</generator>
<lastBuildDate>Sat, 17 Oct 2009 14:38:00 -0400</lastBuildDate>

<image>
  <url>http://media.npr.org/images/npr_news_123x20.gif</url>
  <title>Science</title>
  <link>http://www.npr.org/templates/story/story.php?storyId=1007&amp;ft=1&amp;f=1007</link>
</image>

</channel>
</rss>
Internet Feeders

XML Version of the NPR RSS Feed (2/3 just a fragment!)

```xml
<item>
  <title>California Develops Earthquake Early-Warning System</title>
  <description>
    Lives may be saved with an earthquake early-warning system — the kind that's already in place in Japan and Mexico. But here in the U.S., such a system is still several years away.
  </description>
  <pubDate>Sat, 17 Oct 2009 14:38:00 -0400</pubDate>
  <link>http://www.npr.org/templates/story/story.php?storyId=113877510&amp;ft=1&amp;f=1007</link>
  <guid>http://www.npr.org/templates/story/story.php?storyId=113877510&amp;ft=1&amp;f=1007</guid>
  <content:encoded><![CDATA[
<p>Lives may be saved with an earthquake early-warning system — the kind that's already in place in Japan and Mexico. But here in the U.S., such a system is still several years away.</p>
</item>
```

Many <ITEM>s were intentionally removed to fit page size.
Internet Feeders

XML Version of the NPR RSS Feed (3/3 just a fragment!)

```xml
<item>
  <title>Mapping The Boundaries Of The Solar System</title>
  <description>NASA launched the Interstellar Boundary Explorer (IBEX) spacecraft last year to investigate the edges of the heliosphere; the insulating bubble the sun creates around the solar system. IBEX principal investigator David McComas reports on the first surprising results.</description>
  <pubDate>Fri, 16 Oct 2009 13:19:00 -0400</pubDate>
  <link>http://www.npr.org/templates/story/story.php?storyId=113870291&amp;ft=1&amp;f=1007</link>
  <guid>http://www.npr.org/templates/story/story.php?storyId=113870291&amp;ft=1&amp;f=1007</guid>
  <content:encoded><![CDATA[<p>NASA launched the Interstellar Boundary Explorer (IBEX) spacecraft last year to investigate the edges of the heliosphere; the insulating bubble the sun creates around the solar system. IBEX principal investigator David McComas reports on the first surprising results.</p>]]></content:encoded>
</item>

</channel>
</rss>

Many <ITEM>s were intentionally removed to fit page size

<!-- Burned on demand at 2009-10-17 10:49:43-->
<!-- LIVE -->
<!-- Burned 10/17/2009 10:49:43.407-->
Internet Feeders

Representing the RSS XML as a Tree

The Android API includes a `DocumentBuilderFactory` class to create this type of tree from an input stream.
Document Object Model (DOM).

The **Document Object Model (DOM)** is an application programming interface API for valid *HTML* and well-formed *XML* documents.

With the Document Object Model, programmers can build documents, navigate their structure, and add, modify, or delete elements and content.

Android includes support for DOM managers.

```java
DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
DocumentBuilder db = dbf.newDocumentBuilder();
Document dom = db.parse(someHttpInputStream);
```
Android includes the Apache *HttpComponents* API.
The *HttpClient* component (included in the API) facilitates building HTTP-aware client applications (such as web browsers and web services) using very small footprint.

**HTTP Methods include:**

- **GET**
  Requests a representation of the specified resource.
- **POST**
  Submits data to be processed to the identified resource.
- **HEAD**
  Same as GET but no message body.
- **PUT**
  Uploads a representation of the specified resource.
- **DELETE**
  Deletes the specified resource.
- **TRACE**
  Echoes back the received request.
- **OPTIONS**
  Returns the HTTP methods that the server supports for specified URL.
- **CONNECT**
  Converts the request connection to a transparent TCP/IP tunnel, usually to facilitate SSL-encrypted communication (HTTPS) through an unencrypted HTTP proxy.

**Status Codes:**
Ok, Bad Request, Not Found, ...
Putting all things together – An Example Application

In this project we will develop an application to expose the material typically broadcasted by National Public Radio (NPR) in Android phones.
1. A little research indicates that NPR supports a number of web feeds among them the following:

<table>
<thead>
<tr>
<th>Topic</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arts &amp; Entertainment</td>
<td><a href="http://www.npr.org/rss/rss.php?id=1008">http://www.npr.org/rss/rss.php?id=1008</a></td>
</tr>
</tbody>
</table>
NPR Project – Action Plan.

2. We will display the main topics in a ListView widget from which the user will make a selection. We use the associated web-feed to access the current contents of the selected category.

Choose “Health & Science”
NPR Project – Action Plan.

3. The most current headlines of the selected category are displayed. The User can scroll the list and click on a particular subject.

Observe that individual lines correspond to the XML `<item>` entries discussed earlier.

In the example we are interested in the “Health & Science” subject. We want to know about the efforts of mapping the Solar System.
NPR Project – Action Plan.

4. A brief summary of the chosen topic is displayed inside a DialogBox. The user is given the option of closing the window or obtaining more information. We want additional information, so we click the “More” button.
5. The link held in the XML `<item>` currently displayed is given to a WebKit based Activity. We use the Internet to access the page containing the detailed subject.

We may opt for a non-mobile page version which may include a number of additional components such as audio, pictures, video, extensive text, etc.
Acquiring & Processing the NPR web-feed


2. The Java fragment in the next page uses the **HttpComponents** API.

3. The goal of this code is to grab the feed, and create a document holding the tree parsed from the XML document.

4. The tree will be traversed looking for `<item>` elements similar to those discussed earlier.

5. Each item will be dissected and the pieces arranged into a manageable object form. Remember an `<item>` includes among other components `<title>`, `<description>`, `<link>`, `<datepublication>`, etc.
Main Programming Idea: Acquiring & Processing the NPR web-feed

// connecting to the RSS provider
URL url = new URL(urlAddress);
URLConnection connection = url.openConnection();

HttpURLConnection httpConnection = (HttpURLConnection) connection;

int responseCode = httpConnection.getResponseCode();

if (responseCode == HttpURLConnection.HTTP_OK) {
    // get RSS-XML and make a document
    // holding the parse tree constructed from the input
}
Main Programming Idea: Acquiring & Processing the NPR web-feed

// setting & opening the connection using urlAddress which is
// a value such as: http://www.npr.org/rss/rss.php?id=1001

URL url = new URL(urlAddress);
URLConnection connection;
connection = url.openConnection();

HttpURLConnection httpConnection =
    (HttpURLConnection) connection;

int responseCode = httpConnection.getResponseCode();

if (responseCode == HttpURLConnection.HTTP_OK) {
    // get RSS-XML and make a document
    // holding the parse tree constructed from the input
}
// USE httpConnection to get XML and make a document holding the parse tree constructed from the input
InputStream in = httpConnection.getInputStream();
// define a document builder to act on incoming stream
DocumentBuilderFactory dbf =
    DocumentBuilderFactory.newInstance();
DocumentBuilder db = dbf.newDocumentBuilder();
// make XML parse tree for incoming RSS stream
Document dom = db.parse(in);
// define access nodes in the parse tree
Element docEle = dom.getDocumentElement();

// look for individual news ("items" in this case)
// put items in a NodeList collection (nl)
NodeList nl = docEle.getElementsByTagName("item");
if ((nl != null) && (nl.getLength() > 0)) {
    for (int i = 0; i < nl.getLength(); i++) {
        dissectNode(nl, i);
    } // for
} // if
NPR Project – Implementation

main.xml
(shows the main topics)

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:id="@+id/widget28"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:background="#ff000033"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <ImageView
        android:layout_width="70px"
        android:layout_height="30px"
        android:layout_x="2px"
        android:layout_y="2px"
        android:background="@drawable/logo_npr"/>
    <ListView
        android:id="@+id/myListView"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"/>
</LinearLayout>
```
NPR Project – Implementation

my_simple_list_item_1.xml (shows individual rows of the ListView widget)

```xml
<?xml version="1.0" encoding="utf-8"?>

<TextView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@android:id/text1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:gravity="center_vertical"
    android:paddingLeft="10dp"
    android:textSize="18sp"
    android:minHeight="40sp"
/>
```
Activity1: AndroNPR.java

```java
// display a list of main subjects for the user to choose a topic
package matos.internet;

import java.text.DateFormatSymbols;
import java.util.Calendar;
import java.util.Locale;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;

public class AndroNPR extends Activity {
    ArrayAdapter<String> aa;
    ListView myListView;
    Context context;
    SingleNewsItem selectedNewsItem;
```
Activity 1: AndroNPR.java

String [] myUrlAddress = {
    "http://www.npr.org/rss/rss.php?id=1057"
};

String [] myUrlCaption = {
    "Top Stories",
    "U.S. News",
    "World News",
    "Business",
    "Health & Science",
    "Arts & Entertainment",
    "Politics & Society",
    "People & Places",
    "Opinion"
};

String [] myUrlAddress2 = new String[myUrlCaption.length];

String [] myUrlCaption2 = new String[myUrlCaption.length];
Activity1: AndroNPR.java cont...

```java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    context = getApplicationContext();
    this.setTitle("NPR Headline News v0.0 " + niceDate());

    myListView = (ListView) this.findViewById(R.id.myListView);
    myListView.setOnItemClickListener(new OnItemClickListener() {
        public void onItemClick(AdapterView<?> _av, View _v, int _index, long _id) {
            String urlAddress = myUrlAddress[_index];
            String urlCaption = myUrlCaption[_index];
            //create an Intent to talk to Activity2
            Intent NprNewsDetailsIntent = new Intent(AndroNPR.this, NprNewsDetails.class);
            //prepare a Bundle and add the data pieces to be sent
            Bundle myData = new Bundle();
            myData.putString("urlAddress", urlAddress);
            myData.putString("urlCaption", urlCaption);
            NprNewsDetailsIntent.putExtras(myData);
            startActivity(NprNewsDetailsIntent);
        }
    });
```
Activity1: AndroNPR.java cont...

```java
//bind main category list (Top News, ...) to the the listview
//show listID and get ready for user to click on a category
int layoutID = R.layout.my_simple_list_item_1;
String aa = new ArrayAdapter<String>(this, layoutID, myUrlCaption);
myListView.setAdapter(aa);
```
Activity2: NPRNewsDetails.java

```java
package matos.internet;
import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLConnection;
import java.util.ArrayList;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import org.w3c.dom.DOMException;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.DialogInterface.OnClickListener;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;
import android.widget.AdapterView.OnItemClickListener;
```
public class NprNewsDetails extends Activity {

    // a main category subject has already been selected by the user
    // (data <"urlCaption", "urlAddress"> comes in a bundle sent
    // by main, access web-feed and show corresponding headlines
    ArrayList<SingleNewsItem> newsList = new ArrayList<SingleNewsItem>();
    ArrayAdapter<String> aa;
    ListView myListView;
    String urlAddress = "";
    String urlCaption = "";
    SingleNewsItem selectedNewsItem;
    Context context = getApplication();

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        myListView = (ListView) this.findViewById(R.id.myListView);

        // create a local Intent handler (needed to process input parameters)
        Intent myLocalIntent = getIntent();

        // grab the data bundle with all the pieces sent to us
        // it contains 1. url-address and 2. caption-text
        Bundle myBundle = myLocalIntent.getExtras();
        urlAddress = myBundle.getString("urlAddress");
        urlCaption = myBundle.getString("urlCaption");
Activity2: NPRNewsDetails.java  cont...

//top caption for this screen
String todayStr = AndroNPR.niceDate();
this.setTitle("NPR - " + urlCaption + " \t" + todayStr);

//clicking a line shows more about selected news item
myListView = (ListView) this.findViewById(R.id.myListView);
myListView.setOnItemClickListener(new OnItemClickListener() {
    public void onItemClick(AdapterView<?> _av, View _v,
                            int _index, long _id) {
        selectedNewsItem = newsList.get(_index);
        showNiceDialogBox(selectedNewsItem, context);
    }
});
//onCreate
@Override
protected void onResume() {
    super.onResume();
    try {
        URL url = new URL(urlAddress);
        URLConnection connection;
        connection = url.openConnection();

        HttpURLConnection httpConnection = (HttpURLConnection) connection;
        int responseCode = httpConnection.getResponseCode();
    }
}
Activity2: NPRNewsDetails.java cont...

```java
if (responseCode == HttpURLConnection.HTTP_OK) {
    InputStream in = httpConnection.getInputStream();
    DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
    DocumentBuilder db = dbf.newDocumentBuilder();
    Document dom = db.parse(in);
    Element docEle = dom.getDocumentElement();
    // NodeList nl = docEle.getElementsByTagName("entry");
    NodeList nl = docEle.getElementsByTagName("item");
    if ((nl != null) && (nl.getLength() > 0)) {
        for (int i = 0; i < nl.getLength(); i++) {
            dissectNode(nl, i);
        }
    }
    int layoutID = R.layout.my_simple_list_item_1;
    ArrayAdapter<SingleNewsItem> aaNews =
        new ArrayAdapter<SingleNewsItem>(this, layoutID, newsList);
    myListView.setAdapter(aaNews);
} catch (MalformedURLException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
    Toast.makeText(context, "Trouble!!!", 1).show();
} catch (ParserConfigurationException e) {
    e.printStackTrace();
} catch (SAXException e) {
    e.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
} // onResumee
```
Activity2: NPRNewsDetails.java  cont...

```java
public void dissectNode(NodeList nl, int i) {
    try {
        Element entry = (Element) nl.item(i);
        Element title = (Element) entry.getElementsByTagName("title").item(0);
        Element description = (Element) entry.getElementsByTagName("description").item(0);
        Element pubDate = (Element) entry.getElementsByTagName("pubDate").item(0);
        Element link = (Element) entry.getElementsByTagName("link").item(0);

        String titleValue = title.getFirstChild().getNodeValue();
        String descriptionValue = description.getFirstChild().getNodeValue();
        String dateValue = pubDate.getFirstChild().getNodeValue();
        String linkValue = link.getFirstChild().getNodeValue();

        SingleNewsItem singleItem = new SingleNewsItem(
            dateValue, titleValue, descriptionValue, linkValue);
        newsList.add(singleItem);
    }
    catch (DOMException e) {
        e.printStackTrace();
    }
}
```

//dissectNode
Activity2: NPRNewsDetails.java cont...

```java
public void showNiceDialogBox(SingleNewsItem selectedNewsItem, Context context) {
    //assemble a nice looking dialog box
    try {
        final Uri myLink = Uri.parse(selectedNewsItem.getLink());
        AlertDialog.Builder myBuilder = new AlertDialog.Builder(this);
        myBuilder
            .setIcon(R.drawable.logo_npr)
            .setTitle(urlCaption)
            .setMessage(selectedNewsItem.getTitle() + "\n\n" + selectedNewsItem.getDescription() + "\n"
            .setPositiveButton("Close", null)
            .setNegativeButton("More", new OnClickListener() {
                public void onClick(DialogInterface dialog, int whichOne) {
                    // use native web browsing
                    Intent webIntent = new Intent(Intent.ACTION_VIEW, myLink);
                    startActivity(webIntent);
                }
            });
        myBuilder
            .setNegativeButton(null)
            .show();
    } catch (Exception e) {
        e.printStackTrace();
    }
    } //showNiceDialogBox

//NprNewsDetails
```
Class: SingleNewsItem.java

```java
package matos.internet;

public class SingleNewsItem {
    private String pubDate;
    private String title;
    private String description;
    private String link;

    public String getPubDate() { return pubDate; }
    public String getTitle() { return title; }
    public String getDescription() { return description; }
    public String getLink() { return link; }

    public SingleNewsItem( String _pubDate, String _title, String _description, String _link) {
        pubDate = _pubDate;
        description = _description;
        title = _title;
        link = _link;
    }

    @Override
    public String toString() {
        return title;
    }
}
```
manifest

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="matos.internet"
android:versionCode="1"
android:versionName="1.0.0">
  <application android:icon="@drawable/star_big_on" android:label="@string/app_name">
    <activity android:name=".AndroNPR"
      android:label="@string/app_name">
      <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
      </intent-filter>
    </activity>
    <activity android:name=".NprNewsDetails"/>
  </application>
  <uses-permission android:name="android.permission.INTERNET"/>
  <uses-sdk android:minSdkVersion="4" />
</manifest>
```
Internet

Questions