University of Louisiana at Lafayette

The Center for Advanced Computer Studies (CACS)

Android News Reader

By: Sajjad Pourmohammad (sxp7579)

Table of Contents

1. Motivation 2
2. Project overview 2
   2.1 Steps 3
      2.1.1 Establishing connection 3
      2.1.2 Downloading data from internet 6
3. RSS Handler 7
4. Future works 8

1. Motivation

Internet has provided a unique opportunity to connect all the people together around the world. Almost all the Android devices nowadays support wifi/cellular access to the internet. So, developing softwares able to communicate with the other machines (especially servers) is an undeniable need for all smart devices.

News, events and all the information that can be important or interesting for the reader play a significant role in today's lifestyle. Since android devices are always connected to the internet, they can help in finding, downloading and even informing the user about the news.

2. Project overview

In this project, an application is developed for collecting all the news around the world from different websites. The application has a very user friendly environment which makes it possible for the user to choose the type of the news from the category provided. Figure 1 shows the main activity of the application running on an Android tablet.

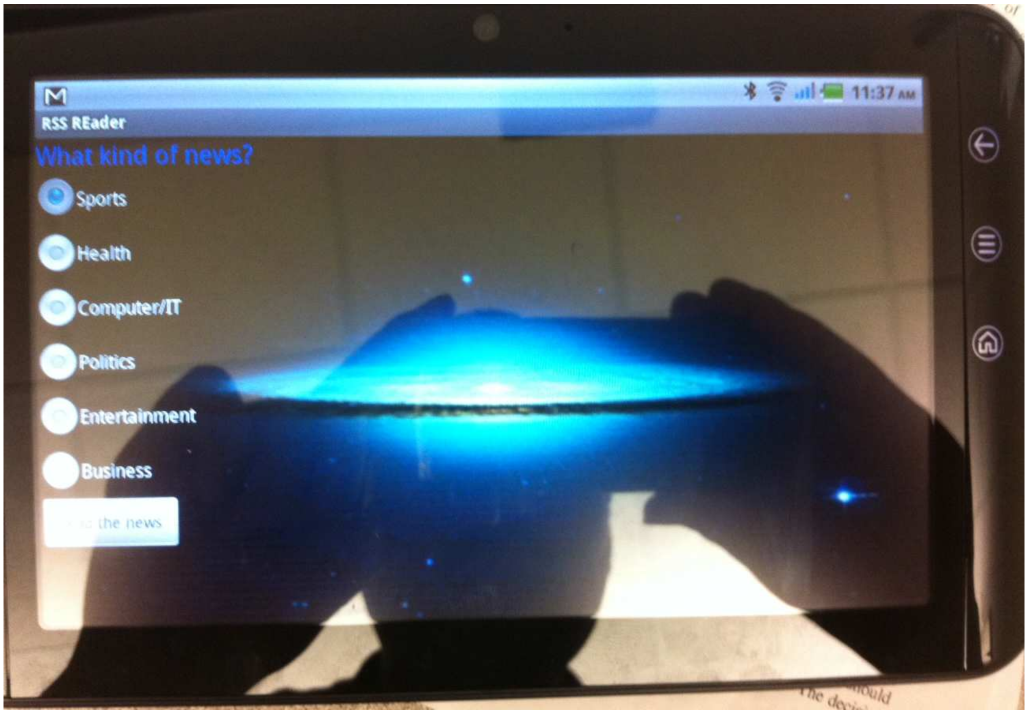

Figure 1. Application running on Android device

In this project, we are going to develop an application to list all the news which may contain valuable information for the reader. The project can be helpful in learning new features provided for the android devices.

2.1 Steps

During the project the following concepts are implemented:

- Establishing a connection to Internet
- Downloading data from internet
- Working with services which are background activities
- Multi-thread programming, since there might be a lot of parallel activities required
- Processing downloaded data

In the next section more details about the project steps are provided.

2.1.1 Establishing connection

There are a lot of protocols to connect to internet. However we should decide about the appropriate one based on our needs. Here are the design steps I went through to find the best option for connection type and the protocol used in the project:

What kind of data is going to be transferred in this project?

Since the news that we are looking for are textual content with possible mixture of links and styles, we should think of a protocol to transfer textual content over the internet. The best way to do this is using HTTP protocol. HTTP is the most popular protocol used in the web servers nowadays and using TCP protocol at transport layer provides a reliable service to the client.

What kind of textual data is preferred?

For our project the best choice is XML. XML is commonly used as a data format on the Internet. If you want to access data from the Internet, chances are that the data will be in the form of XML. If you want to send data to a Web service, you might also need to send XML.

XML's set of tools allows developers to create web pages - and much more. XML allows developers to set standards defining the information that should appear in a document, and in what sequence. XML, in combination with other standards, makes it possible to define the content of a document separately from its formatting, making it easy to reuse that content in other applications or for other presentation environments. Most important, XML provides a basic syntax that can be used to share information between different kinds of computers, different applications, and different organizations without needing to pass through many layers of conversion. XML provides a simple format that is flexible enough to accommodate wildly diverse needs. Even developers performing tasks on different types of applications with different interfaces and different data structures can share XML formats and tools for parsing those formats into data structures that applications can use. XML offers its users many advantages, including[1] :

- Simplicity
- Extensibility
- Interoperability
- Openness
- A core of experienced professionals

One of the greatest strengths of the Android platform is that it leverages the Java programming language. The Android SDK does not quite offer everything available to your standard Java Runtime Environment (JRE,) but it supports a very significant fraction of it. The Java platform has supported many different ways to work with XML for quite some time, and most of Java's XML-related APIs are fully supported on Android. For example, Java's Simple API for XML (SAX) and the Document Object Model (DOM) are both available on Android.

Notice: in order to be able to establish a connection to internet the application should have the internet access permission. So, we should add the following line to Android manifest file:

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

[1] http://www.simonstl.com/articles/whyxml.htm

To establish a HTTP connection we send a request to the server and based on the received reply we will be informed about the connection status. The following function tries to establish a HTTP connection to a server. If it is successful it will return an input stream:

```
private InputStream OpenHttpConnection(String urlString)
throws IOException {
InputStream in = null;
int response = -1;
URL url = new URL(urlString);
URLConnection conn = url.openConnection();
if (!(conn instanceof HttpURLConnection))
throw new IOException(“NotanHTTPconnection”);
Try {
HttpURLConnection httpConn = (HttpURLConnection) conn;
httpConn.setAllowUserInteraction(false);
httpConn.setInstanceFollowRedirects(true);
httpConn.setRequestMethod(“GET”);
httpConn.connect();
response = httpConn.getResponseCode();
if (response == HttpURLConnection.HTTP_OK) {
in = httpConn.getInputStream();}}
catch (Exception ex) {
throw new IOException(“Errorconnecting”);
}
return in;

}
```

2.1.2 Downloading data from internet

After establishing a HTTP connection, the next step is to download the data. Downloading data is just reading textual data from the input stream that we created. The following code shows how this is done:

```
private String DownloadText(String URL) {

int BUFFER_SIZE = 2000;

InputStream in = null;

try {

in = OpenHttpConnection(URL);

} catch (IOException e1) {

Toast.makeText(this, e1.getLocalizedMessage(),

Toast.LENGTH_LONG).show();

e1.printStackTrace();

return “”;}

InputStreamReader isr = new InputStreamReader(in);

int charRead;

String str = “”;

char[] inputBuffer = new char[BUFFER_SIZE];

try {

while ((charRead = isr.read(inputBuffer))>0) {

//---convert the chars to a String---
String readString =

String.copyValueOf(inputBuffer, 0, charRead);

str += readString;

inputBuffer = new char[BUFFER_SIZE]; }

in.close();

} catch (IOException e) {

Toast.makeText(this, e.getLocalizedMessage(),

Toast.LENGTH_LONG).show();
```

3. RSS Handler

As explained before the most challenging part of the project is handling the received RSS file. XML files are a set of <Tags> than contain all the useful information. The following piece of code shows how EndElement methods handle the tags:

```
public void endElement(String uri, String localName, String qName) throws
SAXException {if (localName.equalsIgnoreCase("title")) {

Log.d("LOGGING RSS XML", "Setting article title: " + chars.toString());

currentArticle.setTitle(chars.toString());}

else if (localName.equalsIgnoreCase("description")){

Log.d("LOGGING RSS XML", "Setting article description: " + chars.toString());

currentArticle.setDescription(chars.toString());}

else if (localName.equalsIgnoreCase("pubDate")){

Log.d("LOGGING RSS XML", "Setting article published date: " + chars.toString());

currentArticle.setPubDate(chars.toString());}
```

```
else if (localName.equalsIgnoreCase("encoded")){

Log.d("LOGGING RSS XML", "Setting article content: " + chars.toString());

currentArticle.setEncodedContent(chars.toString());}

else if (localName.equalsIgnoreCase("item")){

                }

else if (localName.equalsIgnoreCase("link")){

try {

        Log.d("LOGGING RSS XML", "Setting article link url: " + chars.toString());

        currentArticle.setUrl(new URL(chars.toString()));

        } catch (MalformedURLException e) {

                Log.e("RSA Error", e.getMessage());}

                }
```

4. Future works

The project was useful in making the basics of transferring data over internet clear. Some future works can be added to this project, which are listed here:

- Some sort of narrator can be added to the project to read the headlines. There are some text to speech engines that can be utilized to achieve this purpose.
- Making the news reader more customizable by providing a textbox to the user; so the user can enter a keyword and search the news for that particular keyword.