Example 10:

Write a mobile application that makes use of RSS feed.

RSS:

RSS stands for Really Simple Syndication. RSS is an easy way to share your website updates and content with your users so that users might not have to visit your site daily for any kind of updates.

RSS Elements:

An RSS document such as above has the following elements.

- **channel**

  This element is used to describe the RSS feed

- **title**

  Defines the title of the channel

- **link**

  Defines the hyper link to the channel

- **description**

  Describes the channel

Parsing RSS:

Parsing an RSS document is more like parsing XML. So now lets see how to parse an XML document.

For this, We will create **XMLPullParser** object , but in order to create that we will first create **XmlPullParserFactory** object and then call its **newPullParser()** method to create **XMLPullParser**.

Its syntax is given below −

```
private XmlPullParserFactory xmlFactoryObject = XmlPullParserFactory.newInstance();
private XmlPullParser myparser = xmlFactoryObject.newPullParser();
```

The next step involves specifying the file for **XmlPullParser** that contains XML. It could be a file or could be a Stream. In our case it is a stream.

Its syntax is given below –

```
myparser.setInput(stream, null);
```

The last step is to parse the XML. An XML file consist of events , Name , Text , **AttributesValue** e.t.c. So **XMLPullParser** has a separate function for parsing each of the components of XML file

Its syntax is given below –

```
int event = myParser.getEventType();
while (event != XmlPullParser.END_DOCUMENT)  {
  String name=myParser.getName();

  switch (event){
    case XmlPullParser.START_TAG:
    break;

    case XmlPullParser.END_TAG:
    if(name.equals("temperature")){
      temperature = myParser.getAttributeValue(null,"value");
    }
    break;
  }
  event = myParser.next();
}
```

The method **getEventType** returns the type of event that happens. e.g: Document start, tag start e.t.c. The method **getName** returns the name of the tag and since we are only interested in temperature, so we just check in conditional statement that if we got a temperature tag, we call the method **getAttributeValue** to return us the value of temperature tag.

Apart from the these methods, there are other methods provided by this class for better parsing XML files. These methods are listed below −

- **getAttributeCount()**

  This method just Returns the number of attributes of the current start tag.

- **getAttributeName(int index)**

  This method returns the name of the attribute specified by the index value.

- **getColumnNumber()**

  This method returns the Returns the current column number, starting from 0.

- **getDepth()**

  This method returns Returns the current depth of the element.

- **getLineNumber()**

  Returns the current line number, starting from 1.

- **getNamespace()**

  This method returns the name space URI of the current element.

- **getPrefix()**

  This method returns the prefix of the current element.

- **getName()**

  This method returns the name of the tag.

- **getText()**

  This method returns the text for that particular element.

- **isWhitespace()**

  This method checks whether the current TEXT event contains only white space characters.