Android File I/O Part 2

XML

Agenda

- Android 資源目錄
- XML
- DOM
- SAX

Android的資源目錄

- /res/xml
- /res/raw
    - Activity : getFileContent()
    - Activity : getResources().openRawResource(R.xxx)
- /assets (可以自訂目錄結構)

assets

- 也可以放在<project>/assets/*
- AssetManager assets = getAssets();
- assets.open(….)

XML

- 標籤語言
- 大小寫有別
- 標簽需成對
- 結構不可以交錯
- 主要有兩種解析方式：DOM vs. SAX

DOM

- Document Object Model是一種平台
  - 動態存取元件
  - 動態更新元件
- DOM有三種
  - Core DOM
  - HTML DOM
  - XML DOM
- XML DOM
  - 一種標準的平台，定義使用者如何取得，修改，新增，刪除XML裡面的元件

DOM常用函數

- 取得一個DOM的parser factory
  - DocumentBuilderFactory factory = DocumentBuilderFactory.*newInstance();*
- 建立一個 Parser
  - DocumentBuilder builder = factory.newDocumentBuilder();
- 從Input Stream中建立DOM物件
  - Document document = builder.parse(inStream);

DOM常用函數

- 取得根節點物件
  - Element root = document.getDocumentElement();
- 依照名字取得一階子節點
  - NodeList nodes = root.getElementsByTagName(”節點名");
  - NodeList可以用一般的迴圈來存取
- 將Node轉換成Element
  - Element bookElement = (Element) nodes.item(i);

DOM常用函數

- 取得Node的內容
  - <x>內容</x>
  - element.getTextContent()
- 取得Element的屬性
  - element.getAttribute()

Android中存取 XML(使用DOM)

- 將XML檔放置在res/raw
- 取得這個XML檔的inputstream
  - getResources().openRawResource(id)
- 產生對應的中間物件
  - 使用Eclipse的property生成
- 使用DOM Parser來解析DOM產生對應的中間物件
- 應用在應用程式之中

XML File

<NewDataSet>

<Table>

<Country>Taiwan</Country>

<City>Kangshan Tw-Afb</City>

</Table>

<Table>

<Country>Taiwan</Country>

<City>Pingtung North Air Force Base</City>

</Table>

.....

</NewDataSet>

DOM Parse Code

```
StringBuilder sb = new StringBuilder();
try {
        DocumentBuilder builder = DocumentBuilderFactory.newInstance().newDocumentBuilder();
        Document doc = builder.parse(getResources().openRawResource(R.raw.cities2));
        Element root = doc.getDocumentElement();
        NodeList list = root.getElementsByTagName("Table");
        for(int i = 0; i < list.getLength();i++){
                Element ele = (Element) list.item(i);
                String country = ele.getElementsByTagName("Country").item(0).getTextContent();
                                .....
        }
} catch (Exception e) {
}
return sb.toString();
```

XML File - 2

…

<GetCitiesByCountryResult><![CDATA[<NewDataSet>

<Table>

<Country>Taiwan</Country>

<City>Kangshan Tw-Afb</City>

</Table>

<Table>

<Country>Taiwan</Country>

<City>Pingtung North Air Force Base</City>

</Table>

…….

</NewDataSet>**]]></GetCitiesByCountryResult>**

…

DOM Parse Code

```
StringBuilder sb = new StringBuilder();
try {
        DocumentBuilder builder = DocumentBuilderFactory.newInstance().newDocumentBuilder();
        Document doc = builder.parse(getResources().openRawResource(R.raw.cities2));
        Element root = doc.getDocumentElement();
        Element data = (Element) root.getElementsByTagName("GetCitiesByCountryResult").item(0);
        String str = data.getTextContent();
        Document nRoot = getDomElement(str);
        NodeList list = nRoot.getElementsByTagName("Table");
        for(int i = 0; i < list.getLength();i++){.....
        }
} catch (Exception e) {
}
return sb.toString();
```

DOM Parse Code

```
protected Document getDomElement(String xml) {
        Document doc = null;
        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
        try {
                DocumentBuilder db = dbf.newDocumentBuilder();
                InputSource is = new InputSource();
                is.setCharacterStream(new StringReader(xml));
                doc = db.parse(is);
        } catch (Exception e) {
                ….
        }
        return doc;
}
```

SAX

- Simple API for XML
- 最早只有Java的實作
- 是Java中存取XML的標準
- 在Android的標準實作中已經包含
- 也是許多web server的XML支援的標準
- SAX是以事件觸發為基礎
  - 依序處理文件並產生對應的事件
- 不支援文件的隨意存取
- 開發需自己實作物件處理器

使用SAX

- SAX所處理的資料必須是完善定義的
  - 遇到特殊字元必須先過濾
- 新增一個 SAX Parser的factory
  - SAXParserFactory factory=SAXParserFactory.*newInstance();*
- *新增一個Parser*
  - SAXParser saxParse=factory.newSAXParser();
- 新增一個文件解析器
  - 必須是DefaultHandler的子類別
- 開始parse文件
  - saxParse.parse(inputStream, handler);

使用SAX

- **public void startDocument() throws SAXException**
  - **開始處理文件**
- **public void endDocument() throws SAXException**
  - **結束文件處理**
- **public void startElement(...) throws SAXException**
  - **開始一個節點的處理**
  - **Attributes是所有屬性的列表，可以用迴圈取出**
- **public void characters(char[] ch, int start, int length)**
  - 處理節點中的內容
- **public void endElement(...) throws SAXException**
  - 結束一個節點的處理

XML File

<CurrentWeather>

<Location>Sungshan / Taipei, Taiwan (RCSS) 25-04N 121-33E</Location>

<Time>Sep 07, 2013 - 08:00 PM EDT / 2013.09.08 0000 UTC</Time>

<Wind> from the E (100 degrees) at 12 MPH (10 KT):0</Wind>

<Visibility> greater than 7 mile(s):0</Visibility>

<SkyConditions> mostly clear</SkyConditions>

<Temperature> 82 F (28 C)</Temperature>

<DewPoint> 69 F (21 C)</DewPoint>

<RelativeHumidity> 65%</RelativeHumidity>

<Pressure> 29.97 in. Hg (1015 hPa)</Pressure>

<Status>Success</Status>

</CurrentWeather>

SAX Parse Code

```
try {
        SAXParser builder = SAXParserFactory.newInstance().newSAXParser();
        builder.parse(getResources().openRawResource(R.raw.taipeiweather2), new DefaultHandler(){
                public void startDocument() throws SAXException {....}
                public void endDocument() throws SAXException {....}
                public void startElement(String uri, String localName,String qName, Attributes attributes) throws SAXException
                {....}
                public void endElement(String uri, String localName,String qName) throws SAXException {....}
                public void characters(char[] ch, int start, int length) throws SAXException {....}
        });
} catch (Exception e) {
        ...
}
```