

Bài Tập 01 – Đọc XML

Đề bài: - Viết chương trình đọc báo của trang 24h đơn giản.

Giao diện:



Khi chọn vào một tin thì hiển thị ra nội dung chi tiết:



Gợi ý thực hiện:

- Lấy dữ liệu thông qua đường link: <http://www.24h.com.vn/upload/rss/tintuctrongngay.rss>
- Dùng DOM Parser để đọc dữ liệu XML.
- Lấy dữ liệu hiển thị lên ListView.
- Lấy đường link hình tải hình về và hiển thị lên ListView.

Hướng dẫn chi tiết:

Bước 1: Thiết kế giao diện file “activity_main.xml” như sau:

```

<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="${relativePackage}.${activityClass}" >
<ListView
    android:id="@+id/listView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:visibility="gone">
</ListView>
<ProgressBar
    android:id="@+id/progressBar"
    style="?android:attr/progressBarStyleLarge"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true" />
</RelativeLayout>

```

Bước 2: Thiết kế item cho ListView trong file "item.xml":

```

<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="10dp" >
<FrameLayout
    android:id="@+id/frameLayout"
    android:layout_width="60dp"
    android:layout_height="60dp"
    android:layout_alignParentLeft="true"
    android:layout_centerVertical="true" >
<ImageView
    android:id="@+id/imageView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:visibility="gone"
    android:contentDescription="@null" />
<ProgressBar
    android:id="@+id/progressBar"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
</FrameLayout>
<LinearLayout
    android:id="@+id/linearLayout1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerVertical="true"
    android:layout_marginLeft="10dp"
    android:layout_toRightOf="@+id/frameLayout"
    android:orientation="vertical" >
<TextView
    android:id="@+id/textViewTitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge" />
<TextView
    android:id="@+id/textViewDescription"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceSmall" />
</LinearLayout>
</RelativeLayout>

```

Bước 3: Sử dụng WebView để hiển thị nội dung tạo trong file “my_web_view.xml”:

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
<WebView
    android:id="@+id/webView"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
</LinearLayout>
```

Bước 4: Tạo đối tượng “Channel” để lưu dữ liệu gồm các thuộc tính: *title*, *link*, *description*, *image*.

Bước 5: Viết xử lý trong “MainActivity.java”:

```
ArrayList<Channel> channels = new ArrayList<>();
public static HashMap<String, Bitmap> map = new HashMap<>();
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    new GetData().execute();
}
```

Chú thích:

- “channels” dùng để lưu lại danh sách dữ liệu nhận được.
- “map” dùng để lưu lại hình đã tải về được.

```
private class GetData extends AsyncTask<String, Void, Void> {
@Override
protected Void doInBackground(String... params) {
    try {
        URL url = new URL
            ("http://www.24h.com.vn/upload/" +
            "rss/tintuctrongngay.rss");
        DocumentBuilderFactory fac = DocumentBuilderFactory
            .newInstance();
        DocumentBuilder builder = fac.newDocumentBuilder();
        Document doc = builder.parse(url.openConnection()
            .getInputStream());
        Element root = doc.getDocumentElement();
        NodeList list = root.getElementsByTagName("item");
        for (int i = 0; i < list.getLength(); i++) {
            Node node = list.item(i);
            if (node instanceof Element) {
                Channel channel = new Channel();
                Element employee = (Element) node;
                NodeList listTitle= employee
                    .getElementsByTagName("title");
                channel.setTitle(listTitle.item(0)
                    .getTextContent());
                NodeList listDescription= employee
                    .getElementsByTagName("description");
                channel.setDescription(listDescription.item(0)
                    .getTextContent());
                NodeList listLink= employee
                    .getElementsByTagName("link");
                channel.setLink(listLink.item(0)
                    .getTextContent());
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}
```

```

        NodeList listSummaryImg= employee
            .getElementsByTagName("summaryImg");
        channel.setImage(listSummaryImg.item(0)
            .getTextContent());
        channels.add(channel);
    }
}
} catch (MalformedURLException e) { e.printStackTrace();
} catch (IOException e) { e.printStackTrace();
} catch (ParserConfigurationException e){
    e.printStackTrace();
} catch (SAXException e) { e.printStackTrace();}
return null;
}
@Override
protected void onPostExecute(Void result) {
    ListView listView = (ListView) findViewById(R.id.listView);
    listView.setVisibility(View.VISIBLE);
    listView.setAdapter (new MyAdapter(channels,
    MainActivity.this));
    listView.setOnItemClickListener(new OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view,
    int position, long id) {
        Intent intent = new Intent(getApplicationContext(),
        WebViewActivity.class);
        intent.putExtra("link",channels.get(position).getLink());
        startActivity(intent);
    }
    });
    (findViewById(R.id.progressBar)).setVisibility(View.GONE);
    super.onPostExecute(result);
}
}

```

Giải thích:

- Element: đại diện cho một thẻ trong XML.
- NodeList: đại diện cho một thẻ có chứa nhiều thẻ con.
- Document: tập tin tài liệu được khởi tạo từ dữ liệu XML thông qua DocumentBuilder.
- DocumentBuilder: đối tượng hỗ trợ chuyển đổi dữ liệu XML thành cấu trúc tập tin XML cho việc đọc ghi dữ liệu.
- DocumentBuilderFactory: khởi tạo đối tượng DocumentBuilder.

Bước 6: Tạo đối tượng ViewHolder:

```

public class ViewHolder {
    TextView textViewTitle, textViewDescription;
    ImageView imageView;
    ProgressBar progressBar;
}

```

Bước 7: Tạo lớp tải hình ảnh theo url:

```

private class DownloadImage extends AsyncTask<ViewHolder,Void, Bitmap>
{
String summaryImg;
ViewHolder viewHolder;
public DownloadImage(String summaryImg) {
this.summaryImg = summaryImg;
}
@Override
protected Bitmap doInBackground(ViewHolder... params) {
try {
viewHolder = params[0];
URLConnection connexion = new URL(summaryImg).openConnection();
connexion.connect();
BufferedInputStream buffinput = new BufferedInputStream(
connexion.getInputStream());
ByteArrayOutputStream dataStream = new
ByteArrayOutputStream(1024);
int current = 0;
while ((current = buffinput.read()) != -1) {
dataStream.write((byte) current);
}
Bitmap bitmap = Bitmap.createBitmap(
BitmapFactory.decodeByteArray( dataStream.toByteArray(),
0, dataStream.toByteArray().length));
MainActivity.map.put(summaryImg, bitmap);
return bitmap;
} catch (MalformedURLException e) {
e.printStackTrace();
} catch (IOException e) {
e.printStackTrace();
}
}
return null;
}
@Override
protected void onPostExecute(Bitmap result) {
if (result != null) {
viewHolder.progressBar.setVisibility(View.GONE);
viewHolder.imageView.setVisibility(View.VISIBLE);
viewHolder.imageView.setImageBitmap(result);
}
super.onPostExecute(result);
}
}
}

```

Bước 8: Viết code Adapter cho ListView:


```

public View getView(int position, View convertView,
ViewGroup parent) {
ViewHolder viewHolder = null;
if (convertView == null) {
convertView = inflater.inflate(R.layout.item, null);
viewHolder = new ViewHolder();
viewHolder.imageView = (ImageView) convertView
.findViewById(R.id.imageView);
viewHolder.textViewTitle = (TextView) convertView
.findViewById(R.id.textViewTitle);
viewHolder.textViewTitle.setSingleLine();
viewHolder.textViewDescription = (TextView) convertView
.findViewById(R.id.textViewDescription);
viewHolder.textViewDescription.setSingleLine();
viewHolder.progressBar = (ProgressBar) convertView
.findViewById(R.id.progressBar);
convertView.setTag(viewHolder);
} else {
viewHolder = (ViewHolder) convertView.getTag();}
viewHolder.textViewTitle.setText(channels.get(position)
.getTitle() + "");
viewHolder.textViewDescription.setText(channels.get(position)
.getDescription() + "");
viewHolder.progressBar.setVisibility(View.VISIBLE);
viewHolder.imageView.setVisibility(View.GONE);
if (MainActivity.map.get(channels.get(position).getImage())
== null) {
new DownloadImage(channels.get(position).getImage())
.execute(viewHolder);
} else {
viewHolder.progressBar.setVisibility(View.GONE);
viewHolder.imageView.setVisibility(View.VISIBLE);
viewHolder.imageView.setImageBitmap(MainActivity.map
.get(channels.get(position).getImage()));
}
return convertView;
}

```

Bước 9: Tạo WebView để hiển thị nội dung khi nhấp vào vào item ListView:

```

public class WebViewActivity extends Activity{
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.my_web_view);
Intent intent = getIntent();
String link = intent.getStringExtra("link");
WebView webView = (WebView) findViewById(R.id.webView);
webView.getSettings().setLoadsImagesAutomatically(true);
webView.getSettings().setJavaScriptEnabled(true);
webView.setScrollBarStyle(View.SCROLLBARS_INSIDE_OVERLAY);
new GetHTML(webView).execute(link);
}
}

```

Trong đó: Lớp GetHTML được viết như sau:

```

public class GetHTML extends AsyncTask<String, Void, String>{
    WebView webView;
    public GetHTML(WebView webView) {
        this.webView = webView;
    }
    @Override
    protected String doInBackground(String... params) {
        try {
            URL url = new URL(params[0]);
            URLConnection urlConnection = url.openConnection();
            HttpURLConnection httpURLConnection = (HttpURLConnection)
            urlConnection;
            int responseCode = httpURLConnection.getResponseCode();
            if (responseCode == HttpURLConnection.HTTP_OK) {
                InputStream inputStream = httpURLConnection.
                getInputStream();
                BufferedReader myReader = new BufferedReader(
                new InputStreamReader(inputStream));
                String aDataRow = "";
                String aBuffer = "";
                while ((aDataRow = myReader.readLine()) != null) {
                    aBuffer += aDataRow + "\n";
                }
                Log.e("TQKy", aBuffer);
                myReader.close();
                inputStream.close();
                return aBuffer;
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
        return null;
    }
    @Override
    protected void onPostExecute(String result) {
        if (result != null) {
            webView.loadDataWithBaseURL("x-data://base", result,
            "text/html", "UTF-8", null);
        }
        super.onPostExecute(result);
    }
}

```