

Laborator 8 Android – Retea

La recomandare aveti sugesita sa create o clasa pentru imagini si link-uri. De aceea creem o clasa noua cu cele 3 atribute:

```
public class ImagineDomeniu { no usages
    private String textAfisat; 3 usages
    private Bitmap imagine; 3 usages
    private String link; 3 usages
```

Adaugam activitatea in care treubie sa avem ListView-ul in care vom afisa imaginile.

```
<ListView
    android:id="@+id/imagini"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

Pentru acest ListView trebuie sa ne creem un Adapter personalizat. Astfel incepem cu adaugarea layout-ului pentru un item in care sa afisam o imagine, si un text (textul descriptiv). Pentru imagine folositi ImageView.

```
<ImageView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/imagineIV"/>
```

Apoi adaugam clasa pentru Adapter in care trebuie sa avem implementata metoda getView().

```
public class ImagineAdapter extends BaseAdapter {
```

Adapterul personalizat trebuie realizat asemanator cu cel din laboratorul precedent.

In activitatea in care avem ListView-ul in care dorim sa afisam lista de imagini in metoda onCreate ne creem o lista cu cele 5 obiecte.

Pentru crearea acestor obiecte vom avea nevoie de cele 5 imagini sau cele 5 link-uri catre aceste imagini de unde sa le descarcam.

```
List<String> linkuriImagini=new ArrayList<>();  
linkuriImagini.add("");  
linkuriImagini.add("");  
linkuriImagini.add("");  
linkuriImagini.add("");  
linkuriImagini.add("");
```

Aceste imagini trebuie sa le descarcam. Pentru descarcare putem folosi AsyncTask sau Executors.

AsyncTask este deprecat, asadar vom folosi Executors.

Prin-tr-un executor vom porni un fir parallel de executie. Pentru intoarcerea la firul principal vom folosi un Handler.

```
Executor executor= Executors.newSingleThreadExecutor();
Handler handler= new Handler(Looper.myLooper());
```

Deoarece aducerea acestor imagini o sa dureze foarte multe este recomandata crearea unui fir de executie secundar, prin intermediul acestui executor.

```
executor.execute(new Runnable() {
    @Override
    public void run() {
        for(String link:linkuriImagini) {
            HttpURLConnection con = null;
            try {
                URL url=new URL(link);
                con= (HttpURLConnection) url.openConnection();
                InputStream is= con.getInputStream();
                imagini.add(BitmapFactory.decodeStream(is));
            }
        }
    }
});
```

Acea lista de imagini o putem crea local sau ca atribut al activitatii in care sunteti.

Pentru intoarcerea la firul principal am spus ca folosim acel handler.

```
handler.post(new Runnable() {
    @Override
    public void run() {
        lista=new ArrayList<>();
        lista.add(new ImagineDomeniu(textAfisat: "La multi ani", imagini.get(0), link: "https://www.google.com"));
        lista.add(new ImagineDomeniu(textAfisat: "Happy anniversary", imagini.get(1), link: "https://www.facebook.com"));
    }
});
```

Pe firul principal ne creem lista de obiecte de tipul clasei pe care am realizat-o mai devreme, in care vom preciza texte, imaginile descarcate si link-uri catre paginile pe care vom dori sa le deschidem in acel WebView.

Cu aceasta lista ne creem adapterul pe care il vom seta listview-ului

```
ListView lv=findViewById(R.id.imagini);
ImagineAdapter adapter=new ImagineAdapter(getApplicationContext(),R.layout.imagine_layout,lista);
lv.setAdapter(adapter);
```

In acest mod afisam lista de imagini si texte in acel listView.

Acum trebuie sa ne abonam la evenimentul de ItemClick pe ListView sis a deschidem acea activitate care contine acel WebView. Acestei activitati trebuie sa ii trimitem link-ul pe care trebuie sa il deschidem in acel WebView.

```
ListView lv=findViewById(R.id.imagini);
lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
        Intent it=new Intent(getApplicationContext(),WebViewActivity.class);
        it.putExtra(name: "link",lista.get(i).getLink());
        startActivity(it);
    }
});
```

Lista – este atributul in care avem toatele obiectele, dar de la obiect luam doar link-ul pe care sa il trimitem catre noua activitate.

ListView-ul din aceasta activitate are id-ul imagini.

In noua activitate trebuie

- sa obtinem link-ul trimis;
- sa identificam webView-ul din layout;
- sa ii setam url-ul pe care sa il deschida.

```
Intent it=getIntent();
String link=it.getStringExtra(name: "link");

WebView wv=findViewById(R.id.webView);
wv.loadUrl(link);
```

Apoi testam ☺.