

DERS İÇERİĞİ:

- İnternet Erişimi
- Process, Threads ve AsyncTask

İnternet Erişimi

Uygulamanın internete erişmesi gerekiyorsa **manifest** dosyasında **<manifest></manifest>** tag'ları arasına bu izni aşağıdaki gibi belirtmek gerekmektedir.

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

Uygulama

Amaç: (Bu kısmı uygulamanın sonunda siz dolduracaksınız!!!)

.....

.....

.....

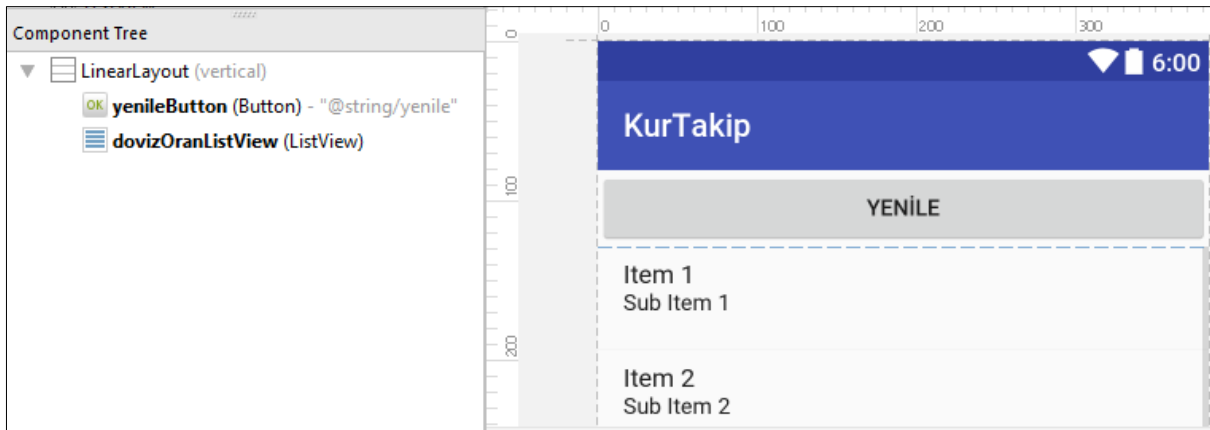
.....

.....

1. Start New Android Project, Appliation Name: **KurTakip**, Min. SDK: API15, Empty Activity , Finish.

```
<resources>
  <string name="menu_settings">Settings</string>
  <string name="title_activity_main">Döviz Takip</string>
  <string name="yenile">Yenile</string>
  <string name="doviz_takip_url">http://www.ecb.int/stats/eurofxref/eurofxref-daily.xml</string>
</resources>
```

2. strings.xml dosyasının içeriğine aşağıdaki kodları ekleyin.
3. Aşağıdaki gibi bir tasarım oluşturun.



4. MainActivity'yi içerisine aşağıdaki gibi bir liste ve bir adapter oluşturun. Ayrıca Yenile butonunun içeriğini yazın.

```
public class MainActivity extends AppCompatActivity {

    private List<String> dovizOranList;
    private ArrayAdapter<String> adapter;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        dovizOranList = getDovizOranList();
        adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, dovizOranList);

        ListView dovizOranLV = (ListView) findViewById(R.id.dovizoranlistview);
        dovizOranLV.setAdapter(adapter);

        Button yenileButton = (Button) findViewById(R.id.yenilebutton);

        yenileButton.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                dovizOranList = getDovizOranList();
                adapter.notifyDataSetChanged();
            }
        });
    }
}
```

5. MainActivity içerisine belirtilen url ile başarılı bir bağlantı gerçekleşmesi halinde döviz oranları indirme işlemini başlatarak olan **getDovizOranList()** metodunu aşağıdaki gibi yazın.

```
private List<String> getDovizOranList() {

    HttpURLConnection uC = null;
    try {
        StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);

        URL url = new URL(getResources().getString(R.string.doviz_takip_url));
        uC = (HttpURLConnection) url.openConnection();
        int sonuokodu = uC.getResponseCode();

        if (sonuokodu == HttpURLConnection.HTTP_OK) {
            BufferedInputStream stream = new BufferedInputStream(uC.getInputStream());
            return getDovizOranListFromInputStream(stream);
        }
    } catch (Exception e) {
        Log.d("DovizTakip", "HTTP bağlantısı kurulurken hata oluştu", e);
    } finally {
        if(uC != null)
            uC.disconnect();
    }
    return new ArrayList<String>();
}
```

6. Yukarıdaki kodda sarı ile işaretlenmiş satırların hangi işlevi yerine getirdiğini arkadaşınızla tartışarak aşağıya yazınız.

.....

.....

.....

7. **MainActivity** içerisine döviz oranlarını stream ile çeken ve parse eden kodu yazın.

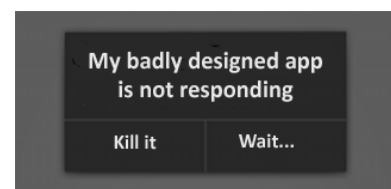
```
private List<String> getDovizOranListFromInputStream(BufferedInputStream stream) {  
  
    List<String> dovizOranList = new ArrayList<String>();  
    if(stream == null)  
        return dovizOranList;  
    try {  
  
        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();  
        DocumentBuilder docBuilder = dbf.newDocumentBuilder();  
        Document d = docBuilder.parse (stream);  
        Element fCube = (Element) d.getElementsByTagName("Cube").item(0);  
        Element sCube = (Element) fCube.getElementsByTagName("Cube").item(0);  
        NodeList dovizOranNL = sCube.getElementsByTagName("Cube");  
  
        for (int i = 0; i < dovizOranNL.getLength(); i++) {  
            Element dovizOranE = (Element) dovizOranNL.item(i);  
            String paraBirimi = dovizOranE.getAttribute("currency");  
            String euroyaOrani = dovizOranE.getAttribute("rate");  
            dovizOranList.add(paraBirimi + " / € = " + euroyaOrani);  
        }  
  
    } catch (Exception e) {  
        Log.d("DovizTakip", "XML parse edilirken hata oluştu", e);  
    }  
    return dovizOranList;  
}
```

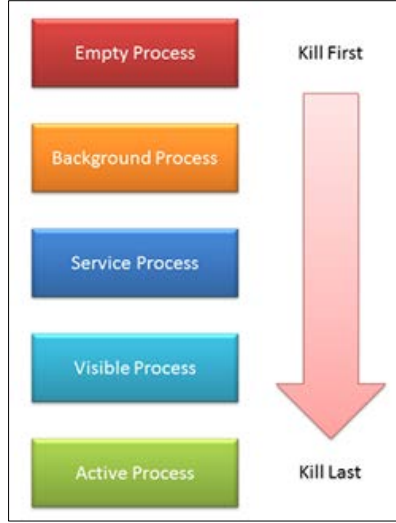
Process, Threads ve AsyncTask

Android uygulaması çalıştığında, sistem tarafından bir **process** başlatılır. Uygulama bu process'e bağlı tek bir **thread** üzerinde yürütülür. Buna **main thread** veya **UI(User Interface) thread** adı verilir. Uygulamanın bir arayüzü olmasa bile bütün component'ler bu thread üzerinden çalıştırılır.



Burada önemli olan, arayüz elemanlarının da diğer bileşenler gibi **main thread** üzerinde çalışıyor olmasıdır. Yapılacak herhangi bir **veritabanı sorgulama** veya **network** işlemi, **arayüzü bloke ederek** kullanıcının işlem yapmasına engel olur. Eğer işleminiz 5 saniye içerisinde tamamlanmazsa, sistem uygulamanızı **ANR(Application Not Responding)** mesajıyla kapatır.





Process yaşam döngüsü yukarıdaki gibidir

Yukarıdaki uygulamada **main thread** içerisinde **getDovizOranList** metodunda bir network işlemi yaptık. Network işlemini main thread içerisinde yapabilmek için kodda bir istisna gerçekleştirdik. Bu istisna sizce hangi satır/satırlarda gerçekleştirilmektedir?

Cevabınızı Buraya Yazın:

.....

.....

.....

Bütün bileşenlerin tek bir thread üzerinden çalışıyor olması, yeni bir thread oluşturmaya engel değildir. Hatta **main thread**'i mümkün olduğunca **meşgul etmemek önerilir**. Ancak şunu unutmamak gerekir ki arayüzü sadece main thread içerisinde güncelleyebilirsiniz.

AsyncTask, yukarıda bahsedilen zaman alacak işlemleri yardımcı bir thread üzerinde, arka planda yapmayı sağlayan, tavsiye edilen bir yöntemdir. Arka plan işlemi devam ederken ya da tamamlandığında arayüzü güncellemeyi de sağlamaktadır.

Uygulama

Amaç: (Bu kısmı uygulamanın sonunda siz dolduracaksınız!!!)

.....

.....

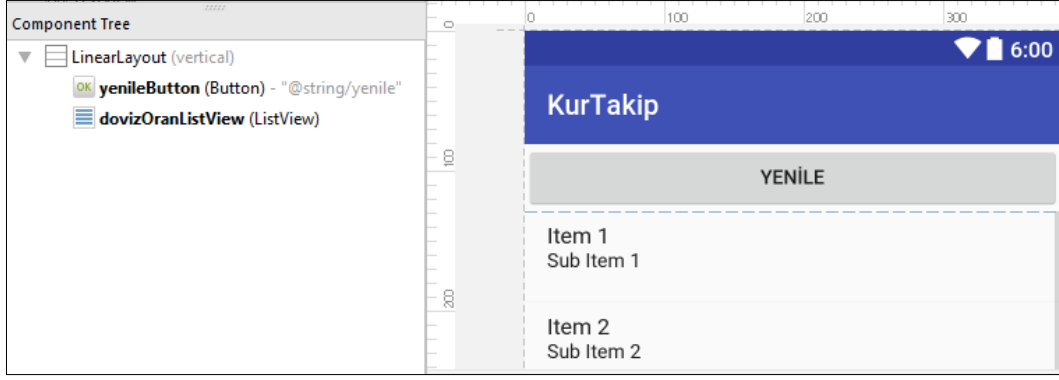
.....

.....

1. Start New Android Project, Appliation Name: **KurTakip2**, Min. SDK: API25, Empty Activity , Finish.
2. **strings.xml** dosyasının içeriğini aşağıdaki kodu ekleyin

```
<resources>
  <string name="doviz_takip_url">http://www.ecb.int/stats/eurofxref/eurofxref-daily.xml</string>
</resources>
```

3. Aşağıdaki gibi bir tasarım oluşturun.



4. Manifest dosyasında internet erişim iznini vermeyi unutmayın!

5. **AsyncTask** sınıfından türeyen **DovizTakipAsyncTask** adlı bir sınıf oluşturun ve içine aşağıdaki kodları ekleyin.

```
public class DovizTakipAsyncTask extends AsyncTask<String, String, List<String>> {

    private static final String TAG = "DovizTakipAsyncTask";
    private Context context;
    private ListView dolv;
    private ProgressDialog pd;
    private ArrayAdapter<String> adapter;

    public DovizTakipAsyncTask(Context context) {
        super();
        this.context = context;
        dolv = (ListView) ((Activity)context).findViewById(R.id.dovizoranlistview);
    }

    protected void onPreExecute() {
        pd = ProgressDialog.show(context, "Lütfen Bekleyin...", "İşlem Yürütülüyor...", true);
        for (int i=0; i<1000; i++){
            for (int j=0; j<1000000; j++){
            }
        }
    }

    protected List<String> doInBackground(String... params) {
        for (int i=0; i<1000; i++){
            for (int j=0; j<1000000; j++){
            }
        }
        return getDovizOranList(params[0]);
    }

}
```

6. onPreExecute() ve doInBackground() metodlarının yaptığı işi arkadaşınızla tartışarak aşağıya yazınız.

.....

.....

.....

.....

.....

.....

7. **DovizTakipAsyncTask** sınıfına `getDovizOranList` metodunu ekleyin.

```
private List<String> getDovizOranList(String dovizTakipUrl) {

    HttpURLConnection uC = null;
    try {
        publishProgress("HTTP bağlantısı kuruluyor...");
        URL url = new URL(dovizTakipUrl);
        uC = (HttpURLConnection) url.openConnection();

        int sonucKodu = uC.getResponseCode();
        if (sonucKodu == HttpURLConnection.HTTP_OK) {
            BufferedInputStream bs = new BufferedInputStream(uC.getInputStream());
            publishProgress("Döviz oranları okunuyor...");
            List<String> dovizOranList = getDovizOranListFromInputStream(bs);
            publishProgress("Liste güncelleniyor...");
            return dovizOranList;
        }
    } catch (Exception e) {
        Log.d(TAG, "HTTP bağlantısı kurulurken hata oluştu", e);
    } finally {
        if (uC != null)
            uC.disconnect();
    }
    return new ArrayList<String>();
}
```

8. **DovizTakipAsyncTask** sınıfına `getDovizOranListFromInputStream` metodunu ekleyin.

```
private List<String> getDovizOranListFromInputStream(BufferedInputStream stream) {

    List<String> doranlist = new ArrayList<String>();
    if (stream == null)
        return doranlist;
    try {
        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
        DocumentBuilder docBuilder = dbf.newDocumentBuilder();
        Document document = docBuilder.parse(stream);
        Element fCube = (Element) document.getElementsByTagName("Cube").item(0);
        Element sCube = (Element) fCube.getElementsByTagName("Cube").item(0);
        NodeList nl = sCube.getElementsByTagName("Cube");
        int nll = nl.getLength();

        for (int i = 0; i < 5; i++) {

            Random random = new Random();
            int rasgeleSayi = random.nextInt(nll);
            Element doe = (Element) nl.item(rasgeleSayi);
            String paraBirimi = doe.getAttribute("currency");
            String euroyaOrani = doe.getAttribute("rate");
            doranlist.add(paraBirimi + " / € = " + euroyaOrani);
        }
    } catch (Exception e) {
        Log.d(TAG, "XML parse edilirken hata oluştu", e);
    }
    return doranlist;
}
```

9. 7. Ve 8. Adımlarda eklediğiniz metodların ilk uygulamadakilerden farkını arkadaşınızla tartışıp aşağıya yazınız.

.....

.....

.....

.....

.....

10. DovizTakipAsyncTask sınıfına aşağıdaki metotları ekleyin.

```
protected void onProgressUpdate(String... progress) {
    pd.setMessage(progress[0]);
    for (int i=0;i<1000;i++){
        for (int j=0; j<1000000; j++){
        }
    }
}
protected void onPostExecute(List<String> result)
{
    adapter = new ArrayAdapter<String>(context, android.R.layout.simple_list_item_1, result);
    dolv.setAdapter(adapter);
    pd.cancel();
}
```

11. Yukarıdaki metodların yaptığı işi arkadaşınızla tartışıp aşağıya yazınız.

.....

.....

.....

.....

.....

12. MainActivity sınıfının içeriğine aşağıdaki kodları ekleyin.

```
private DovizTakipAsyncTask task;

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    task = new DovizTakipAsyncTask(this);
    task.execute(getResources().getString(R.string.doviz_takip_url));

    Button yenileButton = (Button) findViewById(R.id.yenileButton);
    yenileButton.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            task = new DovizTakipAsyncTask(MainActivity.this);
            task.execute(getResources().getString(R.string.doviz_takip_url));
        }
    });
}
```



Sarı ile işaretlenmiş satırdaki **DovizTakipAsyncTask** nesnesini yeniden **execute** etmek yerine, pembe ile işaretli satırda tekrar oluşturup ikinci kez execute ettik. Çünkü bir **AsyncTask yalnızca bir defa execute edilebilir**. Aynı nesnenin **execute** metodu ikinci kez çağrılırsa hataya sebep olur. Bu yüzden ihtiyaç duyulan her sefer yeni bir nesne oluşturulmalıdır.