Http request
GET, POST, PUT

Nikola Ralev, Ph.D.
nralev@setelis.com
Setelis Labs LLC

Permissions

Request permission to access network

AndroidManifest.xml

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

Using HttpClient

```
HttpClient httpclient = new DefaultHttpClient();
HttpResponse response = httpclient.execute(new HttpGet(url));
StatusLine statusLine = response.getStatusLine();

if(statusLine.getStatusCode() == HttpStatus.SC_OK){
       ByteArrayOutputStream out = new ByteArrayOutputStream();
       response.getEntity().writeTo(out);
       out.close();
      return out.toString();

   } else{
       //Closes the connection.
       response.getEntity().getContent().close();
       throw new IOException(statusLine.getReasonPhrase());
   }
```

Using HttpURLConnection

```
URL URL = new URL(url);
   HttpURLConnection urlConnection = (HttpURLConnection) URL.openConnection();
   try {
     InputStream in = new BufferedInputStream(urlConnection.getInputStream());
     return in.toString();
   }
    finally {
     urlConnection.disconnect();
   }
```

AsyncTask

- **android.os.NetworkOnMainThreadException** is thrown

```
class AsyncGetTask extends AsyncTask<String, Void, String> {

    protected String doInBackground(String... urls) {
        try {
               //Execute request
            return resposne;
        } catch (Exception e) {
            return null;
        }
    }

    protected void onPostExecute(String parsedData) {

    }
}
```

Post request

```
HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();
  try {
    urlConnection.setDoOutput(true);
    urlConnection.setChunkedStreamingMode(0);

    OutputStream out = new BufferedOutputStream(urlConnection.getOutputStream());
    writeStream(out);

    InputStream in = new BufferedInputStream(urlConnection.getInputStream());
    readStream(in);
   finally {
    urlConnection.disconnect();
  }
}
```

Parse JSON

- Using JSONObject
- Using jackson
- Other

Parse with JSONObject

```
"address_components" : [
         { "long_name" : "Sofia",
          "short_name" :"Sofia"
         }
         ]}
```

```
JSONObject jObject = new JSONObject(json);

JSONArray jArray = jObject.getJSONArray("address_components");
JSONObject arrayObject = (JSONObject) jArray.get(0);

JSONObject long_name = arrayObject.getString("long_name");
JSONObject short_name = arrayObject.getString("short_name");
```

Parse with Jackson-RS

- Map POJOs (Plain Old Java Objects) back and forth from JSON
- Download lib and add to project

  http://jackson.codehaus.org/
- Sample json/xml url

  http://maps.googleapis.com/maps/api/geocode/json?address=Sofia&sensor=false

Class representing the json

```
"address_components" : [
         { "long_name" : "Sofia",
          "short_name" :"Sofia"
         }
         ]}
```

```
public class AddressComponent {
  private String long_name;
  private String short_name;

  public String getLong_name() {
    return long_name;
  }

  public void setLong_name(String long_name) {
    this.long_name = long_name;
  }

  public String getShort_name() {
    return short_name;
  }

  public void setShort_name(String short_name) {
    this.short_name = short_name;
  }
}
```

```
ObjectMapper mapper = new ObjectMapper();
return mapper.readValue(jsonStream,
Results.class);
```

Parsing XML

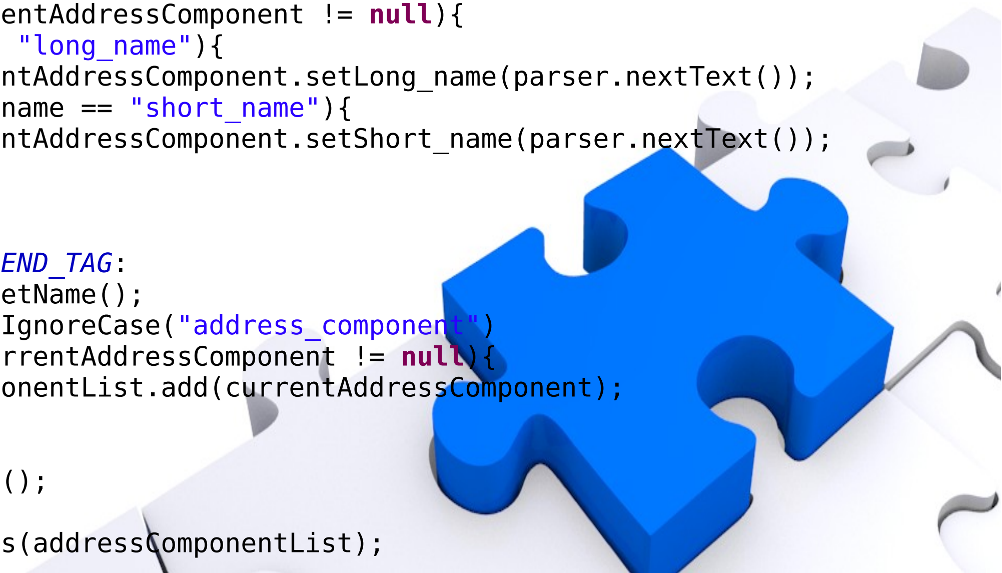
- **XMLPullParser**
  - recommended by Android
- **DOM parser**
  - creates a complete memory model of the XML
  - create or parse XML
  - Uses a lot of memory
- **SAXParsers**
  - Only parse XML.

Parsing with XMLPullParse

```
<result>
<address_component>
  <long_name>Sofia, Bulgaria</long_name>
  <short_name>Sofia</short_name>
</address_component>
<address_component>
  <long_name>Plovdiv, Bulgaria</long_name>
  <short_name>Plovdiv</short_name>
</address_component>
<result>
```

```
private void parseXml(InputStream xmlStream) throws XmlPullParserException,IOException{
XmlPullParser parser = Xml.newPullParser();
parser.setFeature(XmlPullParser.FEATURE_PROCESS_NAMESPACES, false);
parser.setInput(xmlStream,null);

Result result;
List<AddressComponent> addressComponentList = new ArrayList<AddressComponent>();
AddressComponent currentAddressComponent =null;
int eventType = parser.getEventType();
        while (eventType != XmlPullParser.END_DOCUMENT){
            String name = null;
            switch (eventType){
                case XmlPullParser.START_DOCUMENT:
                  result = new Result();
                    break;
                case XmlPullParser.START_TAG:
                    name = parser.getName();
                    if (name == "address_component"){
                        currentAddressComponent = new AddressComponent();
                    } else if (currentAddressComponent != null){
                        if (name == "long_name"){
                              currentAddressComponent.setLong_name(parser.nextText());
                        } else if (name == "short_name"){
                              currentAddressComponent.setShort_name(parser.nextText());
                        }
                    }
                    break;
                case XmlPullParser.END_TAG:
                    name = parser.getName();
                    if (name.equalsIgnoreCase("address_component")
                              && currentAddressComponent != null){
                        addressComponentList.add(currentAddressComponent);
                    }
            }
            eventType = parser.next();
        }
        result.setAddress_components(addressComponentList);
}
```



Any questions ?