MODULE 13

WEB API

Françoise Dubisy

TABLE OF CONTENT

Françoise Dubisy

Internet Permission

▸ To access mobile services

○ Add Internet permission into AndroidManifest.xml

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

Françoise Dubisy

Access to Web Api

▸ Use Async Task to access Web Api

  ⇨ Do not block the main thread during loading!

▸ Async Task in Activity

  ◦ doInBackground : to load data

  ◦ onPostExecute : to display data on the screen

Françoise Dubisy

Access to Web Api

▸ E.g,

```
import org.json.*;
import java.io.*;
import java.net.*;

public class PersonDAO
{
  public ArrayList<Person> getAllPersons() throws Exception
  {   URL url = new URL("http://api.androidhive.info/volley/person_array.json");
      URLConnection connection = url.openConnection();
      BufferedReader br = new BufferedReader(new InputStreamReader(connection.getInputStream()));
      StringBuilder sb = new StringBuilder();
      String stringJSON = "", line;
      while ((line = br.readLine()) != null)
         { sb.append(line); }
      br.close();
      stringJSON = sb.toString();
      return jsonToPersons(stringJSON);
  }
}
```

Connection to URL

Reading
of the string

Françoise Dubisy

Access to Web Api

```java
private ArrayList<Person> jsonToPersons(String stringJSON) throws Exception
{   ArrayList<Person> persons = new ArrayList<>();
    Person person;
    JSONArray jsonArray  = new JSONArray(stringJSON);
    for (int i = 0; i < jsonArray.length(); i++)
        { JSONObject jsonPerson = jsonArray.getJSONObject(i);
          person = new Person(jsonPerson.getString("name"),jsonPerson.getString("email"));
          persons.add(person);
        }
    return persons;
}
```

Françoise Dubisy

Access to Web Api

```java
public class MainActivity extends AppCompatActivity {
  private ListView personList;

  @Override
  protected void onCreate(Bundle savedInstanceState)
  { ...
     new LoadPerson().execute();
  }
  private class LoadPerson extends AsyncTask<String,Void,ArrayList<Person>>
  {       @Override
          protected ArrayList<Person> doInBackground(String... params)
          {   PersonDAO personDAO = new PersonDAO();
          ArrayList<Person> persons = new ArrayList<>();
          try { persons = personDAO.getAllPersons(); }
          catch (Exception e) { ... }
          return persons;
           }

          @Override
          protected void onPostExecute(ArrayList<Person> persons)
          { ...
           personList.setAdapter(adapter);
           }
  }
}
```
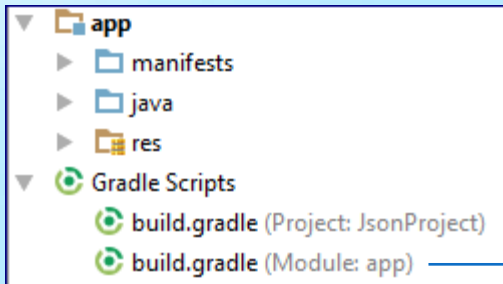
Display the list of persons
on the screen

Volley

- HTTP library to make networking for Android apps

- Volley is not suitable for large download or streaming operations

- Do not use Volley because of coupling of layers !

  - Access to data and modification of the view in the same class!!!

Françoise Dubisy

Gson

▸ Library for converting Java objects to and from JSON

▸ Using reflection

◦ Define a Java class that has the same attribute (instance variable) names as the Json keys

◦ Pass to Gson the Json object string and this Java class

⇨ Gson fills automatically the instance variables of the object

Françoise Dubisy

Gson

- https://github.com/google/gson

- Add dependency to build.gradle


```
dependencies {
        compile 'com.google.code.gson:gson:2.8.0'
        ...
}
```

Françoise Dubisy

Gson

- ▸ E.g,

  - ◦ without Gson

```
JSONObject jsonPerson = jsonArray.getJSONObject(i);
person = new Person(jsonPerson.getString("name"),jsonPerson.getString("email"));
```

  - ◦ With Gson

```
import com.google.gson.*;
```

```
JSONObject jsonPerson = jsonArray.getJSONObject(i);
Gson object = new GsonBuilder().create();
person = object.fromJson(jsonPerson.toString(), Person.class);
```

Françoise Dubisy