|  |  |
| --- | --- |
|  | **UNIVERSIDAD DON BOSCO**  **FACULTAD DE INGENIERÍA – ESCUELA DE COMPUTACIÓN** |
|  | TALLER ANDROID  **TEMA VI – Networking**  Uso de Volley, Gson y WebServices |

**I Objetivos**

* Utilizar Volley para realizar peticiones a endpoints.
* Serializar contenido json a objetos de Java
* Comprender el funcionamiento de las peticiones a webservices

**II Introducción**

Volley es una librería desarrollada por Google para optimizar el envío de peticiones Http desde las aplicaciones Android hacia servidores externos.

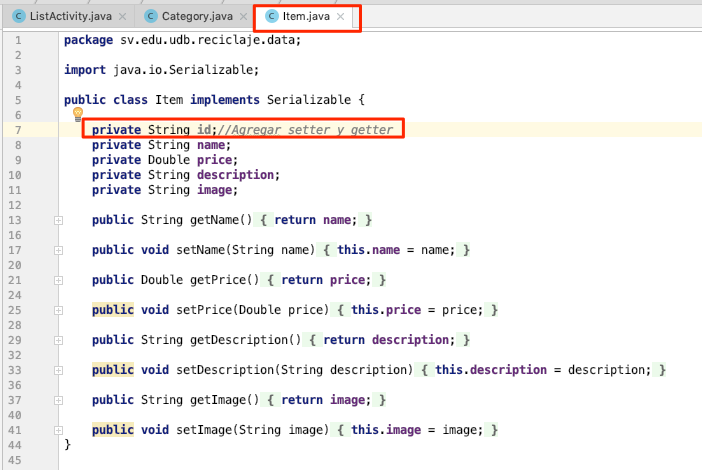
Este componente actúa como una interfaz de alto nivel, liberando al programador de la administración de hilos y procesos tediosos de parsing, para permitir publicar fácilmente resultados en el hilo principal.

**III Desarrollo**

1. Agregar librerías:

|  |
| --- |
| */\*Librería de networking Volley\*/* implementation **'com.android.volley:volley:1.1.1'** */\*Librería de serialización json\*/* implementation **'com.google.code.gson:gson:2.8.5'** |

1. **Crear una nueva actividad denominada OfferActivity**
2. Modificaremos **Item.java** debido a que no cuenta con un atributo id todavía:



1. Crear la clase CategoryDAO.java. Contendrá el siguiente código:

|  |
| --- |
| **package** sv.edu.udb.reciclaje.data;  **import** android.content.Context; **import** android.util.Log;  **import** com.android.volley.Request; **import** com.android.volley.RequestQueue; **import** com.android.volley.Response; **import** com.android.volley.VolleyError; **import** com.android.volley.toolbox.JsonArrayRequest; **import** com.android.volley.toolbox.Volley; **import** com.google.gson.Gson;  **import** org.json.JSONArray;  **import** java.util.ArrayList; **import** java.util.Arrays; **import** java.util.List;  **public class** CategoryDAO {   **public static final** String ***API\_URI*** = **"http://udb.ddns.net:8080/taller/reciclaje/categories"**;    Context **context**;  RequestQueue **requestQueue**;   CategoryListener **categoryListener**;   **public** CategoryDAO(Context context){  **this**.**context** = context;  **requestQueue** = Volley.*newRequestQueue*(**this**.**context**);  **categoryListener** = (CategoryListener) context;  }   **public void** loadCategoryList(){  JsonArrayRequest request = **new** JsonArrayRequest(  Request.Method.***GET***,  ***API\_URI***,  **null**, **new** Response.Listener<JSONArray>() {  @Override  **public void** onResponse(JSONArray response) {  Log.*d*(**"UDBLog"**,response.toString());  List<Category> categoryList = load(response);  **categoryListener**.onCategoryResponse(categoryList);  }  }, **new** Response.ErrorListener() {  @Override  **public void** onErrorResponse(VolleyError error) {   }  });   **requestQueue**.add(request);  }   **public** List<Category> load(JSONArray response){  Gson gson = **new** Gson();   Category [] categoryList = gson.fromJson(response.toString(),Category[].**class**);   **return new** ArrayList<>(Arrays.*asList*(categoryList));   }   **public interface** CategoryListener{  **public void** onCategoryResponse(List<Category> categoryList);  }  } |

1. Limpie el ListActivity (dejarlo en blanco). Ahora vamos a reescribirlo:

ListActivity.java

|  |
| --- |
| **package** sv.edu.udb.reciclaje;  **import** android.content.Intent; **import** android.support.v7.app.ActionBar; **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.util.Log; **import** android.view.MenuItem; **import** android.view.View; **import** android.widget.AdapterView; **import** android.widget.ArrayAdapter; **import** android.widget.ListView; **import** android.widget.Spinner;  **import** org.json.JSONArray;  **import** java.util.ArrayList; **import** java.util.List;  **import** sv.edu.udb.reciclaje.data.Category; **import** sv.edu.udb.reciclaje.data.CategoryDAO; **import** sv.edu.udb.reciclaje.data.Data; **import** sv.edu.udb.reciclaje.data.Item;  **public class** ListActivity **extends** AppCompatActivity **implements** CategoryDAO.CategoryListener {   List<String> **categoryStringList** = **new** ArrayList<>();  List<String> **itemStringList** = **new** ArrayList<>();   Spinner **categorySpinner**;  ListView **itemListView**;   *//Agregado* List<Category> **categoryList** = **new** ArrayList<>();   ArrayAdapter<String> **arrayAdapter**;   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_list***);    ActionBar actionBar = getSupportActionBar();  actionBar.setDisplayHomeAsUpEnabled(**true**);   **categorySpinner** = findViewById(R.id.***categoryspinner***);  **itemListView** = findViewById(R.id.***itemlistview***);  **arrayAdapter** = **new** ArrayAdapter<String>(**this**,android.R.layout.***simple\_spinner\_dropdown\_item***,**categoryStringList**);  **categorySpinner**.setAdapter(**arrayAdapter**);   **categorySpinner**.setOnItemSelectedListener(**new** AdapterView.OnItemSelectedListener() {  @Override  **public void** onItemSelected(AdapterView<?> adapterView, View view, **int** i, **long** l) {  loadListView();  }   @Override  **public void** onNothingSelected(AdapterView<?> adapterView) {   }  });   }   @Override  **protected void** onResume() {  **super**.onResume();   CategoryDAO categoryDAO= **new** CategoryDAO(**this**);  categoryDAO.loadCategoryList();  }    **void** loadSpinner(){   **categoryStringList**.clear();   **for**(Category category:**this**.**categoryList**){  **categoryStringList**.add(category.getName());  }   **arrayAdapter**.notifyDataSetChanged();    }  **void** loadListView(){  **final int** index = **categorySpinner**.getSelectedItemPosition();   **itemStringList**.clear();   **for**(Item selected:**this**.**categoryList**.get(index).getItemList()){  **itemStringList**.add(selected.getName());  }   ArrayAdapter<String> listViewAdapter = **new** ArrayAdapter<String>(**this**,android.R.layout.***simple\_list\_item\_1***,**itemStringList**);   **itemListView**.setAdapter(listViewAdapter);   **itemListView**.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  @Override  **public void** onItemClick(AdapterView<?> adapterView, View view, **int** i, **long** l) {  *//Modificado* Item item = **categoryList**.get(index).getItemList().get(i);  Log.*d*(**"UDBLog"**, item.getDescription() );  Log.*d*(**"UDBLog"**, item.getName() );  Log.*d*(**"UDBLog"**, item.getPrice() + **""** );   Intent intent = **new** Intent(ListActivity.**this**,DetailActivity.**class**);  intent.putExtra(**"item"**,item);  startActivity(intent);   }  });   **itemListView**.deferNotifyDataSetChanged();  }   @Override  **public boolean** onOptionsItemSelected(MenuItem item) {  **switch** (item.getItemId()) {  **case** android.R.id.***home***:  finish();  **return true**;  }   **return super**.onOptionsItemSelected(item);  }   @Override  **public void** onCategoryResponse(List<Category> categoryList) {  **this**.**categoryList** = categoryList;  loadSpinner();  } } |

1. Editar activity\_detail.xml

|  |  |
| --- | --- |
|  | *<?***xml version="1.0" encoding="utf-8"***?>* <**android.support.v7.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".DetailActivity"  app:cardElevation="10dp"  app:cardCornerRadius="10dp"  android:layout\_margin="5dp"  app:contentPadding="5dp"** >   <**RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/itemname"  android:textSize="40sp"  android:background="#94200C"  android:textColor="#fff"** />  <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/itemdescription"  android:layout\_below="@+id/itemname"  android:textSize="20sp"** />  <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Valor: $"  android:id="@+id/labelvalue"  android:textSize="20sp"  android:layout\_below="@+id/itemdescription"** />  <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:id="@+id/itemvalue"  android:layout\_below="@+id/itemdescription"  android:layout\_toRightOf="@+id/labelvalue"  android:textSize="20sp"** />  <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text=" por libra"  android:id="@+id/labelpound"  android:textSize="20sp"  android:layout\_below="@+id/itemdescription"  android:layout\_toRightOf="@+id/itemvalue"** />  <**ImageView  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:id="@+id/itemimage"  android:layout\_below="@+id/itemvalue"  android:layout\_margin="10dp"** />  </**RelativeLayout**>   <**android.support.design.widget.FloatingActionButton  android:id="@+id/fab"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_gravity="bottom|end"  android:layout\_margin="20dp"  android:layout\_alignParentBottom="true"  android:layout\_alignParentRight="true"  android:src="@android:drawable/ic\_dialog\_email"  android:onClick="add"** />  </**android.support.v7.widget.CardView**> |

1. Nuestro DetailActivity tendrá una función nueva llamada add (Marcado en amarillo):

|  |
| --- |
| **package** sv.edu.udb.reciclaje;  **import** android.content.Intent; **import** android.support.v7.app.ActionBar; **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.view.MenuItem; **import** android.view.View; **import** android.widget.ImageView; **import** android.widget.TextView;  **import** com.squareup.picasso.Picasso;  **import** sv.edu.udb.reciclaje.data.Item;  **public class** DetailActivity **extends** AppCompatActivity {   Item **item**;   TextView **itemName**;  TextView **itemDescription**;  TextView **itemValue**;  ImageView **itemImage**;   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);   ActionBar actionBar = getSupportActionBar();  actionBar.setDisplayHomeAsUpEnabled(**true**);   setContentView(R.layout.***activity\_detail***);     **itemName** = findViewById(R.id.***itemname***);  **itemDescription** = findViewById(R.id.***itemdescription***);  **itemValue** = findViewById(R.id.***itemvalue***);  **itemImage** = findViewById(R.id.***itemimage***);   **item** = (Item) getIntent().getSerializableExtra(**"item"**);   **itemName**.setText(**item**.getName());  **itemValue**.setText(**item**.getPrice() + **""**);  **itemDescription**.setText(**item**.getDescription());   Picasso.*get*().load(**item**.getImage()).into(**itemImage**);   }   @Override  **public boolean** onOptionsItemSelected(MenuItem item) {  **switch** (item.getItemId()) {  **case** android.R.id.***home***:  finish();  **return true**;  }   **return super**.onOptionsItemSelected(item);  }   **public void** add(View view) {  Intent intent = **new** Intent(**this**, OfferActivity.**class**);   intent.putExtra(**"item"**,**item**);   startActivity(intent);   } } |

1. Editar activity\_offer.xml

|  |  |
| --- | --- |
|  | *<?***xml version="1.0" encoding="utf-8"***?>* <**android.support.v7.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  app:cardElevation="10dp"  app:cardCornerRadius="10dp"  android:layout\_margin="5dp"  app:contentPadding="5dp"** > <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".OfferActivity"  android:orientation="vertical"** >   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/recycleritem"  android:textAlignment="center"  android:textSize="20dp"** />   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/price"  android:textAlignment="center"  android:textSize="20dp"** />   <**EditText  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/quantity"  android:inputType="number"  android:hint="Ingrese cantidad"** />  <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Centro de reciclaje"  android:textAlignment="center"** />    <**Spinner  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/recyclercenter"** />   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/email"  android:textAlignment="center"** />   <**Button  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="enviar"  android:onClick="presend"  android:id="@+id/btnsend"  android:background="#C20E0E"  android:textColor="#fff"** /> </**LinearLayout**> </**android.support.v7.widget.CardView**> |

1. Modificamos nuestro OfferActivity.java

|  |
| --- |
| **package** sv.edu.udb.reciclaje;  **import** android.arch.persistence.room.Room; **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.view.View; **import** android.widget.AdapterView; **import** android.widget.ArrayAdapter; **import** android.widget.Button; **import** android.widget.EditText; **import** android.widget.Spinner; **import** android.widget.TextView; **import** android.widget.Toast;  **import** com.android.volley.Request; **import** com.android.volley.RequestQueue; **import** com.android.volley.Response; **import** com.android.volley.VolleyError; **import** com.android.volley.toolbox.JsonObjectRequest; **import** com.android.volley.toolbox.Volley; **import** com.mobsandgeeks.saripaar.ValidationError; **import** com.mobsandgeeks.saripaar.Validator; **import** com.mobsandgeeks.saripaar.annotation.NotEmpty;  **import** org.json.JSONObject;  **import** java.util.ArrayList; **import** java.util.HashMap; **import** java.util.List; **import** java.util.Map;  **import** sv.edu.udb.reciclaje.data.AppDatabase; **import** sv.edu.udb.reciclaje.data.Item; **import** sv.edu.udb.reciclaje.data.RecyclingCenter; **import** sv.edu.udb.reciclaje.data.RecyclingCenterDAO;  **public class** OfferActivity **extends** AppCompatActivity **implements** Validator.ValidationListener {    **public static** Spinner *recyclerCenterSpinner*;   List<String> **centerList** = **new** ArrayList<>();   List<RecyclingCenter> **recyclingCenterList**;   Item **item**;   TextView **recyclerItem**;   TextView **price**;  Button **btnsend**;  TextView **email**;   @NotEmpty  EditText **quantity**;   Validator **validator**;   RecyclingCenter **recyclingCenter**;      @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_offer***);   *recyclerCenterSpinner* = findViewById(R.id.***recyclercenter***);  **recyclerItem** = findViewById(R.id.***recycleritem***);  **price** = findViewById(R.id.***price***);  **quantity** = findViewById(R.id.***quantity***);  **email** = findViewById(R.id.***email***);  **btnsend** = findViewById(R.id.***btnsend***);   **item** =(Item) getIntent().getSerializableExtra(**"item"**);  **recyclerItem**.setText(**item**.getName());  **price**.setText(**item**.getPrice().toString());    **validator** = **new** Validator(**this**);  **validator**.setValidationListener(**this**);   AppDatabase database = Room.*databaseBuilder*(**this**, AppDatabase.**class**, **"mydb"**)  .allowMainThreadQueries()  .build();   RecyclingCenterDAO recyclingCenterDAO = database.getRecyclingCenterDAO();   **recyclingCenterList**=recyclingCenterDAO.getRecyclingCenterList();   loadSpinner();     *recyclerCenterSpinner*.setOnItemSelectedListener(**new** AdapterView.OnItemSelectedListener() {  @Override  **public void** onItemSelected(AdapterView<?> adapterView, View view, **int** i, **long** l) {  loadEmail(i);  }   @Override  **public void** onNothingSelected(AdapterView<?> adapterView) {   }  });  }     **void** loadSpinner(){  **centerList**.clear();   **for**(RecyclingCenter tmp:**recyclingCenterList**){  **centerList**.add(tmp.getName());  }    ArrayAdapter<String> arrayAdapter;   arrayAdapter = **new** ArrayAdapter<String>(**this**,android.R.layout.***simple\_spinner\_dropdown\_item***,**centerList**);   *recyclerCenterSpinner*.setAdapter(arrayAdapter);   loadEmail(0);*//Cargamos el primer sitio para el email al inicio de la ventana* }    **void** loadEmail(**int** index){  RecyclingCenter recyclingCenter = **recyclingCenterList**.get(index);  **email**.setText(recyclingCenter.getEmail());  **this**.**recyclingCenter** = recyclingCenter;  }   **public void** presend(View view) {  **validator**.validate();  }   @Override  **public void** onValidationSucceeded() {  sendEmail();  }   **private void** sendEmail() {   **btnsend**.setEnabled(**false**);   String url = **"http://udb.ddns.net:8080//taller/reciclaje/email"**;  String apikey = **"3erfgbhnjko0kmjn"**;*//conserve segura esta llave* **double** valueQuantity = Double.*parseDouble*(**quantity**.getText().toString());   **double** price = **item**.getPrice() \* valueQuantity;   String message = **"Notifico envío de "** + **item**.getName() + **" con precio de $"** + price + **" "**;   Map<String, String> params = **new** HashMap();  params.put(**"email"**, **recyclingCenter**.getEmail());  params.put(**"subject"**, **"COTIZACION"**);  params.put(**"content"**,message);  params.put(**"apikey"**,apikey);   JSONObject parameters = **new** JSONObject(params);    RequestQueue queue = Volley.*newRequestQueue*(**this**);  JsonObjectRequest jsonObjectRequest = **new** JsonObjectRequest(Request.Method.***POST***, url, parameters, **new** Response.Listener<JSONObject>() {  @Override  **public void** onResponse(JSONObject response) {  OfferActivity.**this**.finish();  }  }, **new** Response.ErrorListener() {  @Override  **public void** onErrorResponse(VolleyError error) {  **btnsend**.setEnabled(**true**);  Toast.*makeText*(OfferActivity.**this**,**"Error "** + error.getMessage(),Toast.***LENGTH\_LONG***).show();  }  });   queue.add(jsonObjectRequest);  }   @Override  **public void** onValidationFailed(List<ValidationError> errors) {  **for** (ValidationError error : errors) {  View view = error.getView();  String message = error.getCollatedErrorMessage(**this**);   *// Display error messages ;)* **if** (view **instanceof** EditText) {  ((EditText) view).setError(message);  } **else** {  Toast.*makeText*(**this**, message, Toast.***LENGTH\_LONG***).show();  }  }  } } |

1. Ahora es requerido que usted aplique los siguientes cambios:

* Bloquear la rotación de pantalla de las actividades.
* Generar la navegación para las actividades que no la tengan.
* Agregar elementos en CardView donde no se encuentre.

**IV Ejercicio complementario**

|  |
| --- |
| Proyecto: **Sistema de cotizaciones móvil.** |
| Localmente vía **~~GreenDAO~~ Room**, almacenará su base de datos de clientes. Debe guardar nombre, dirección, teléfono, email.  Listará categorías de productos, al seleccionarlo desplegará los productos con sus precios. Esta información será accedida vía webservice.  El usuario vía GreenDAO podrá guardar la información de la cotización. Deberá guardar el nombre de usuario de aplicación, el correo(destino), los IDs de los item en la cotización individualmente, cantidad, precio sugerido (webservice) precio ofertado (manual) y el total.  Finalmente, la información será enviada en formato json vía petición POST hacia el webservice, el cual enviará un correo detallado hacia el cliente.  La aplicación pedirá usuario y password (Esto estará habilitado al final del taller). |

Se ha iniciado un webservice con listado de categorías y productos. La documentación puede accederla desde:

<http://udb.ddns.net:8080/taller>

Los endpoints que utilizará son:

/taller/proyecto/categories  
/taller/proyecto/products

/taller/reciclaje/email

Consuma el endpoint que usted considere adecuado para que al seleccionar uno o varios productos, pueda estimar el total del costo. Al realizar la cotización (un producto o varios productos), debe enviar un correo al cliente. Recuerde el funcionamiento del endpoint de email:

|  |
| --- |
| Use POST Request (postman, form, android client, etc) Required post fields: email, subject, content, apikey Request to admin the apikey.  **apikey: 3erfgbhnjko0kmjn** |

Tips a recordar:

**apikey:** 3erfgbhnjko0kmjn

**Documentación:** <http://udb.ddns.net:8080/taller>