|  |  |
| --- | --- |
|  | **UNIVERSIDAD DON BOSCO**  **FACULTAD DE ESTUDIOS TECNOLÓGICOS**  **COORDINACIÓN DE COMPUTACIÓN Y MÓVILES** |
| **Ciclo II** | **Desarrollo de Avanzado de Software para móviles** Guía de Laboratorio No. 2 Submit de Datos con Retrofit |

* **I Objetivos**
* Hacer envío de ficheros por medio de Retrofit.
* Acceder a webservices y utilizar sus respectivos endpoints.
* **II Desarrollo**

Para esta guía usted hará uso del proyecto creado previamente en la guía anterior. Usted modificará y añadirá funcionalidad a la aplicación.

Verifique que cuenta con las librerías anteriores necesarias y ahora incluirá Matisse que le dará la habilidad de seleccionar imágenes desde la biblioteca del dispositivo.

|  |
| --- |
| compile **'com.github.bumptech.glide:glide:3.6.1'** compile **'com.squareup.retrofit2:retrofit:2.3.0'** compile **'com.squareup.retrofit2:converter-gson:2.3.0'** *//La version de design depende de su compile version* compile **'com.android.support:design:27.1.1'**  *//image picker* compile **'com.zhihu.android:matisse:0.5.1'** |

La aplicación ahora contará con permisos de lectura y escritura de medios externos:

|  |
| --- |
| <**uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"**/> <**uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"**/> |

Editar EndPointInterface.java

|  |
| --- |
| **package** sv.edu.udb.retrofitexample;  **import** java.util.List;   **import** okhttp3.MultipartBody; **import** okhttp3.RequestBody; **import** okhttp3.ResponseBody; **import** retrofit2.Call; **import** retrofit2.http.Field; **import** retrofit2.http.FormUrlEncoded; **import** retrofit2.http.GET; **import** retrofit2.http.Multipart; **import** retrofit2.http.POST; **import** retrofit2.http.Part; **import** retrofit2.http.Path;   **public interface** EndPointInterface {   String ***base\_url*** = **"http://34.211.243.185:8080/"**;   *//Agregamos la ruta de imágenes en el webservice* String ***image\_base\_url*** = **"http://34.211.243.185:8080/images/"**;   @GET(**"autores"**)  Call<List<Autor>> getAutores();   @GET(**"autores/{id}"**)  Call<Autor> getAutor(@Path(**"id"**) String id);   *//Mantener especial cuidado en utilizar o no utilizar  //los slash "/" para acceder a los recursos  //el método post requiere que no se utilicen* @FormUrlEncoded  @POST(**"autores"**)  Call<ResponseBody> addAutor(  @Field(**"nombre"**) String nombre);   *//Mantener especial cuidado en utilizar o no utilizar  //los slash "/" para acceder a los recursos  //el método post requiere que no se utilicen* @FormUrlEncoded  @POST(**"chistes"**)  Call<ResponseBody> addChiste(  @Field(**"idautor"**) **int** idautor,  @Field(**"nombre"**) String nombre,  @Field(**"texto"**) String texto);    *//El método de agregar ahora incorporando  //subida de imágenes.* @Multipart  @POST(**"chistes"**)  Call<ResponseBody> addChiste(  @Part(**"idautor"**) **int** idautor,  @Part(**"nombre"**) String nombre,  @Part(**"texto"**) String texto,  @Part MultipartBody.Part image);  } |

Agregar una nueva actividad denominada AddActivity

Editar AddActivity.java

|  |
| --- |
| **package** sv.edu.udb.guia01;  **import** android.Manifest; **import** android.content.Context; **import** android.content.Intent; **import** android.content.pm.PackageManager;  **import** android.database.Cursor; **import** android.net.Uri; **import** android.provider.MediaStore; **import** android.support.v4.app.ActivityCompat; **import** android.support.v4.content.ContextCompat; **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.util.Log; **import** android.view.View; **import** android.widget.EditText; **import** android.widget.ImageView; **import** android.widget.ProgressBar; **import** android.widget.RelativeLayout; **import** android.widget.ScrollView; **import** android.widget.Toast;  **import** com.bumptech.glide.Glide; **import** com.zhihu.matisse.Matisse; **import** com.zhihu.matisse.MimeType; **import** com.zhihu.matisse.engine.impl.GlideEngine;  **import** java.io.File; **import** java.io.IOException; **import** java.util.List;  **import** okhttp3.MediaType; **import** okhttp3.MultipartBody; **import** okhttp3.RequestBody; **import** okhttp3.ResponseBody; **import** retrofit2.Call; **import** retrofit2.Callback; **import** retrofit2.Response; **import** retrofit2.Retrofit; **import** retrofit2.converter.gson.GsonConverterFactory;   **public class** AddActivity **extends** AppCompatActivity {   ProgressBar **progressBar**;  ImageView **imageView**;  ScrollView **scrollView**;   **static int** *REQUEST\_CODE\_CHOOSE* = 432;  *//Contenedor de la imagen* File **imageFile**;    Retrofit **retrofit** = **new** Retrofit.Builder()  .baseUrl(EndPointInterface.***base\_url***)  .addConverterFactory(GsonConverterFactory.*create*())  .build();   EndPointInterface **apiService** =  **retrofit**.create(EndPointInterface.**class**);    @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_add***);  **imageView**=(ImageView)findViewById(R.id.***imagenchiste***);  }     **public void** add(View view) {   EditText nombre, texto;  nombre = (EditText) findViewById(R.id.***txtnombre***);  texto = (EditText) findViewById(R.id.***txttexto***);   RelativeLayout relativeLayout = findViewById(R.id.***mainlayout***);  ProgressBar progressBar = findViewById(R.id.***progressbar***);   relativeLayout.setVisibility(View.***GONE***);  progressBar.setVisibility(View.***VISIBLE***);    Chiste chiste = **new** Chiste();  chiste.setId\_autor(1);*//Id de usuario Anonimo.* chiste.setNombre(nombre.getText().toString());  chiste.setTexto(texto.getText().toString());   RequestBody requestBody=**null**;   *//Hay imagen para enviar* **if**(**imageFile** != **null**){  requestBody = RequestBody.*create*(  MediaType.*parse*(getContentResolver().getType(**mSelected**.get(0))),  **imageFile**);   MultipartBody.Part body =  MultipartBody.Part.*createFormData*(**"imagen"**, **imageFile**.getName(), requestBody);   Call<ResponseBody> call = **apiService**.addChiste(  chiste.getId\_autor(),  chiste.getNombre(),  chiste.getTexto(),  body);   call.enqueue(**new** Callback<ResponseBody>() {  @Override  **public void** onResponse(Call<ResponseBody> call, Response<ResponseBody> response) {    **try** {  Log.*d*(**"UDBTEST:info"**,response.body().string().toString());  } **catch** (IOException e) {  e.printStackTrace();  }     AddActivity.**this**.finish();   }   @Override  **public void** onFailure(Call<ResponseBody> call, Throwable t) {  Log.*d*(**"UDBTEST:error reason"**,t.getMessage());  Toast.*makeText*(AddActivity.**this**,**"No fue posible publicar su informacion"**,Toast.***LENGTH\_SHORT***).show();  }  });  }**else**{  Call<ResponseBody> call = **apiService**.addChiste(  chiste.getId\_autor(),  chiste.getNombre(),  chiste.getTexto()  );   call.enqueue(**new** Callback<ResponseBody>() {  @Override  **public void** onResponse(Call<ResponseBody> call, Response<ResponseBody> response) {  **try** {  Log.*d*(**"UDBTEST:info"**,response.body().string().toString());  } **catch** (IOException e) {  e.printStackTrace();  }     AddActivity.**this**.finish();  }   @Override  **public void** onFailure(Call<ResponseBody> call, Throwable t) {  Log.*d*(**"UDBTEST:error reason"**,t.getMessage());  Toast.*makeText*(AddActivity.**this**,**"No fue posible publicar su informacion"**,Toast.***LENGTH\_SHORT***).show();  }  });  }   }   **public void** fileFromGallery(View view){   **if** (ContextCompat.*checkSelfPermission*(**this**,  Manifest.permission.***READ\_EXTERNAL\_STORAGE***)  != PackageManager.***PERMISSION\_GRANTED***) {     *// Should we show an explanation?* **if** (ActivityCompat.*shouldShowRequestPermissionRationale*(**this**,  Manifest.permission.***READ\_EXTERNAL\_STORAGE***)) {   *// Show an explanation to the user \*asynchronously\* -- don't block  // this thread waiting for the user's response! After the user  // sees the explanation, try again to request the permission.* } **else** {   *// No explanation needed, we can request the permission.* ActivityCompat.*requestPermissions*(**this**,  **new** String[]{Manifest.permission.***READ\_EXTERNAL\_STORAGE***},  1);   }  }**else**{  *//Librería matisse* Matisse.*from*(**this**)  .choose(MimeType.*of*(MimeType.***JPEG***))*//SOLO ACEPTARA JPEG* .countable(**true**)  .maxSelectable(1)*//SOLO UN ARCHIVO* .thumbnailScale(0.85f)  .imageEngine(**new** GlideEngine())  .forResult(*REQUEST\_CODE\_CHOOSE*);*//* }   }   @Override  **public void** onRequestPermissionsResult(**int** requestCode,  String permissions[], **int**[] grantResults) {  **switch** (requestCode) {  **case** 1: {   *// If request is cancelled, the result arrays are empty.* **if** (grantResults.**length** > 0  && grantResults[0] == PackageManager.***PERMISSION\_GRANTED***) {   Toast.*makeText*(AddActivity.**this**,**"Permisos otorgados para acceder a fotos"**,Toast.***LENGTH\_SHORT***);   } **else** {   *// permission denied,Disable the  // functionality that depends on this permission.* Toast.*makeText*(AddActivity.**this**, **"Permiso denegado para acceder a fotos"**, Toast.***LENGTH\_SHORT***).show();  }  **return**;  }   *// other 'case' lines to check for other  // permissions this app might request* }  }   List<Uri> **mSelected**;   *//Utilizando la libreria Matisse para el resultado* **protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  **if** (requestCode == *REQUEST\_CODE\_CHOOSE* && resultCode == ***RESULT\_OK***) {  **mSelected** = Matisse.*obtainResult*(data);  *//Obteniendo la imagen* Log.*d*(**"Matisse"**, **"mSelected: "** + **mSelected**);  String path = *getRealPathFromUri*(**this**, **mSelected**.get(0));  **imageFile** = **new** File(path);  Glide.*with*(**this**).load(**mSelected**.get(0)).centerCrop().into(**imageView**);  }  }    **public static** String getRealPathFromUri(Context context, Uri contentUri) {  Cursor cursor = **null**;  **try** {  String[] proj = { MediaStore.Images.Media.***DATA*** };  cursor = context.getContentResolver().query(contentUri, proj, **null**, **null**, **null**);  **int** column\_index = cursor.getColumnIndexOrThrow(MediaStore.Images.Media.***DATA***);  cursor.moveToFirst();  **return** cursor.getString(column\_index);  } **finally** {  **if** (cursor != **null**) {  cursor.close();  }  }  }   } |

Modifique activity\_add.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**ScrollView  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"**>  <**RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>   <**ProgressBar  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:visibility="gone"  android:id="@+id/progressbar"** />    <**RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context="sv.edu.udb.guia01.AddActivity"  android:id="@+id/mainlayout"** >    <**EditText  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:hint="nombre"  android:id="@+id/txtnombre"** />   <**EditText  android:layout\_width="match\_parent"  android:layout\_height="200dp"  android:hint="Texto"  android:id="@+id/txttexto"  android:layout\_below="@+id/txtnombre"** />    <**Button  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Galería"  android:onClick="fileFromGallery"  android:layout\_below="@+id/txttexto"  android:id="@+id/gallery"** />   <**Button  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Agregar"  android:onClick="add"  android:layout\_below="@+id/gallery"  android:id="@+id/add"** />   <**ImageView  android:layout\_width="200dp"  android:layout\_height="200dp"  android:layout\_gravity="center"  android:id="@+id/imagenchiste"  android:layout\_below="@+id/add"  android:layout\_centerHorizontal="true"** />  </**RelativeLayout**> </**RelativeLayout**> </**ScrollView**> |

Modificaremos el MainActivity.java para que ahora cargue la actividad de AddActivity. Busque la sección de código en color amarillo a continuación:

|  |
| --- |
| **package** sv.edu.udb.retrofitexample;  **import** android.content.Intent; **import** android.support.design.widget.FloatingActionButton; **import** android.support.design.widget.Snackbar; **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.util.Log; **import** android.view.View; **import** android.widget.AdapterView; **import** android.widget.ArrayAdapter; **import** android.widget.ListView;  **import** com.google.gson.Gson; **import** com.google.gson.GsonBuilder;  **import** java.util.ArrayList; **import** java.util.List;  **import** retrofit2.Call; **import** retrofit2.Callback; **import** retrofit2.Response; **import** retrofit2.Retrofit; **import** retrofit2.converter.gson.GsonConverterFactory;  **public class** MainActivity **extends** AppCompatActivity {    ListView **listView**;   List<String> **stringListAutores**=**new** ArrayList<>();  List<Autor> **listAutores**;   Gson **gson** = **new** GsonBuilder()  .setDateFormat(**"yyyy-MM-dd"**)  .create();    Retrofit **retrofit** = **new** Retrofit.Builder()  .baseUrl(**"http://34.211.243.185:8080/"**)  *//.addCallAdapterFactory(RxJava2CallAdapterFactory.create())* .addConverterFactory(GsonConverterFactory.*create*(**gson**))  .build();   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);   **listView** = (ListView) findViewById(R.id.***listView***);   EndPointInterface apiService = **retrofit**.create(EndPointInterface.**class**);    Call<List<Autor>> call = apiService.getAutores();   call.enqueue(**new** Callback<List<Autor>>() {  @Override  **public void** onResponse(Call<List<Autor>> call, Response<List<Autor>> response) {  **listAutores** = response.body();  fillData(**listAutores**);  }   @Override  **public void** onFailure(Call<List<Autor>> call, Throwable t) {  Log.*d*(**"UDBFAILURE:Error"**,t.getMessage());  }  });    FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.***fab***);  fab.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View view) {  *//Apertura de actividad de agregar.* Intent intent = **new** Intent(MainActivity.**this**,AddActivity.**class**);  startActivity(intent);  }  });  }   **private void** fillData(List<Autor> autores) {   **for**(Autor autor:autores){  **this**.**stringListAutores**.add(autor.getNombre());  }   ArrayAdapter<String> arrayAdapter = **new** ArrayAdapter<>(**this**,android.R.layout.***simple\_list\_item\_1***,**this**.**stringListAutores**);  **listView**.setAdapter(arrayAdapter);   **listView**.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  @Override  **public void** onItemClick(AdapterView<?> adapterView, View view, **int** i, **long** l) {  Intent intent = **new** Intent(MainActivity.**this**,GridListActivity.**class**);   intent.putExtra(**"idautor"**,String.*valueOf*(**listAutores**.get(i).getId()));   startActivity(intent);   }  });  }   } |

* GridListActivity.java no tiene modificaciones.

SpeakRequest.java no tiene modificaciones.

* Activity\_detail.xml no tiene modificaciones.
* Activity\_grid\_list.xml no tiene modificaciones.
* Activity\_main.xml no tiene modificaciones.

Grid\_item.xml no tiene modificaciones.

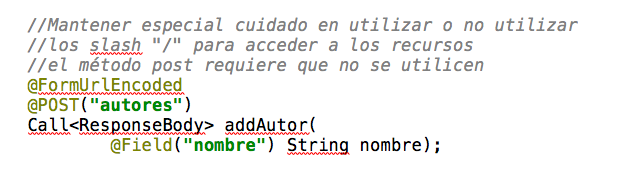
Autor.java no tiene modificaciones.

Chiste.java no tiene modificaciones.

ChistesAdapter no tiene modificaciones.

* **III Ejercicios Complementarios**

Existe un endpoint denominado autor al que debe acceder para registrarse.



Actualmente no se registra un nuevo usuario. Modifique la aplicación para que al iniciarse, solicite al usuario un user name y lo almacene localmente, de esa manera modificar la aplicación para no enviar nuevas publicaciones de tipo “Anonymous” (id 1).

