Web Services (servicios web)

Son servicios que a través de Api’s puede realizar cualquier operación (CRUD) desde un celular, Tablet u otro dispositivo en un servidor Web



Requerimientos

* Tener Instalado Xampp, Visual Studio Code y Android Studio
* Tener conexión de Internet

Pasos

1. Crear una base de datos llamada Empresa en MySql con una tabla llamada Usuario con los siguientes campos:



Create Database Empresa

* Crear la tabla Usuarios dentro de la base de datos Empresa

CREATE TABLE Usuarios(

usr varchar(15) PRIMARY key,

nombre varchar(50) not null,

correo varchar (50) not null,

clave varchar(10) not null,

activo varchar(2) not null default 'si')

Crear al menos tres registros

insert into Usuario (usr,nombre,correo,clave)

values('FreddyMC','Freddy Moscoso','fmc@gmail.com','1234'),

('LuisaFer','Luisa Fernanda','lfmg@gmail.com','1234'),

('DianaP','Diana Patricia','dpga@gmail.com','1234')

* Crear una copia de seguridad por Exportar (crear plantilla y exportar)

1. Crear la carpeta WebServices en C:\Xampp\htdocs

* Ingresar al Visual Studio Code y crear las Api’s correspondientes
* Ingreso.php

<?php

if (isset($\_REQUEST['correo']) && isset($\_REQUEST['clave']))

{

$correo=$\_REQUEST['correo'];

$clave=$\_REQUEST['clave'];

$cnx = mysqli\_connect("localhost","root","","Empresa");

$res=$cnx->query("select \* from Usuarios where correo = '$correo' and clave = '$clave'");

$json = array();

foreach ($res as $row)

{

$json['datos'][]=$row;

}

//pasar los datos del array a JSON con información o vacío

echo json\_encode($json);

}

else

{

echo "El correo y la clave son obligatorios";

}

?>

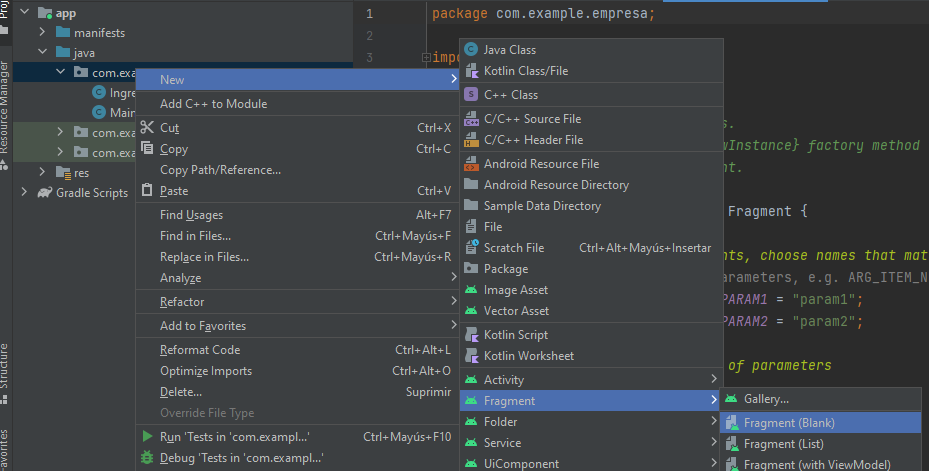
1. Crear una aplicación en Android Studio llamada Empresa con un formulario con un solo layout (con nombre y orientación horizontal), queda así:

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout 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:id="@+id/Escenario"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".MainActivity">  
</LinearLayout>

* Crear una vista (fragment) dentro de la actividad



* Los pasos para crearla son los siguientes



* Dar el nombre de IngresoFragment
* Cambiar en el diseño del fragment el tipo de Layout por LinearLayout y su orientación debe ser vertical
* Agregar el siguiente diseño en fragment\_ingreso.xml

<TextView  
 android:layout\_marginTop="200dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Inicio de Sesion"  
 android:gravity="center\_horizontal" />  
  
<TextView  
 android:layout\_marginTop="10dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Correo Electronico"  
 android:gravity="center\_horizontal" />  
  
<EditText  
 android:id="@+id/etcorreo"  
 android:layout\_width="300dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:hint="p.e. micorreo@gmail.com"  
 android:inputType="textEmailAddress"  
 android:gravity="center\_horizontal" />  
  
<TextView  
 android:layout\_marginTop="10dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Contraseña"  
 android:gravity="center\_horizontal" />  
  
<EditText  
 android:id="@+id/etclave"  
 android:layout\_width="300dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:hint="Contraseña"  
 android:inputType="textPassword"  
 android:gravity="center\_horizontal" />  
  
<Button  
 android:layout\_marginTop="20dp"  
 android:id="@+id/btingresar"  
 android:layout\_height="wrap\_content"  
 android:layout\_width="200dp"  
 android:text="Ingresar"  
 android:layout\_gravity="center\_horizontal" />

<TextView  
 android:layout\_marginTop="10dp"  
 android:id="@+id/tvregistrar"  
 android:layout\_width="200dp"  
 android:layout\_height="wrap\_content"  
 android:text="Haz clic AQUI para Registrarse!"  
 android:textColor="#3F51B5"  
 android:layout\_gravity="center\_horizontal"  
 android:gravity="center\_horizontal" />

* Eliminar el código en IngresoFragment.Java para que solo quede el ultimo método:

Líneas desde la 16 hasta 54 y desde 9 a 13

Queda así:

package com.example.empresa;  
  
import android.os.Bundle;  
import androidx.fragment.app.Fragment;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
  
public class IngresoFragment extends Fragment {  
   
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 // Inflate the layout for this fragment  
 return inflater.inflate(R.layout.*fragment\_ingreso*, container, false);  
 }  
}

* Cambiar el layout del MainActivity por la vista, se ingresa al método OnCreate del MainActivity.Java y se adiciona el siguiente código:

//Ocultar la barra de menu por defecto  
getSupportActionBar().hide();  
//Libreria para el manejo de vistas (fragment)  
FragmentManager fm = getSupportFragmentManager();  
fm.beginTransaction().replace(R.id.*Escenario*,new IngresoFragment()).commit();

* Ejecutar desde el emulador...
* Especificar en GradleScripts (build.gradle(Module.app)) la siguiente dependencia que permita hacer conexiones http: (permite hacer conexiones remotas)

implementation 'com.android.volley:volley:1.2.1'

* Hacer clic en Sync now....(parte superior derecha)
* En el archivo AndroidManifest.xml (está en la carpeta manifest, encima de la carpeta Java), especificar la línea que permite tomar datos de internet:

<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE" />  
  
<application  
 android:allowBackup="true"

Nota: debe colocarse antes de <Aplication

* En la vista fragment (IngresoFragment.java) agregar al encabezado de la clase el siguiente código:

implements Response.Listener<JSONObject>,Response.ErrorListener

debe quedar:

public class IngresoFragment extends Fragment implements Response.Listener<JSONObject>,Response.ErrorListener{  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 // Inflate the layout for this fragment  
 return inflater.inflate(R.layout.*fragment\_ingreso*, container, false);  
 }  
  
 @Override  
 public void onErrorResponse(VolleyError error) {  
  
 }  
  
 @Override  
 public void onResponse(JSONObject response) {  
  
 }  
 }

Nota: La clase a implementar es una clase abstracta y por lo tanto necesita que se implementen los métodos abstractos

(click derecho sobre el encabezado de la clase, generate, implements methods y Aceptar)

....y se genera 2 métodos...onResponse y onErrorResponse ....

* Definir los objetos que se requieren para la conexión

EditText jetcorreo,jetclave;

Button jbtingresar;

TextView jtvregistrar;

RequestQueue rq;

JsonRequest jrq;

* Asociar objetos Xml con objetos Java
* Se debe poner en comentario el return por defecto y cambiarlo por mi propia vista
* // return inflater.inflate(R.layout.fragment\_ingreso, container, false);  
    
   //Asociar objetos  
   View vista = inflater.inflate(R.layout.*fragment\_ingreso*,container,false);  
   jetcorreo = vista.findViewById(R.id.*etcorreo*);  
   jetclave = vista.findViewById(R.id.*etclave*);  
   jbtingresar = vista.findViewById(R.id.*btingresar*);  
   jtvregistrar=vista.findViewById(R.id.*tvregistrar*);
* jbtingresar.setOnClickListener(new View.OnClickListener() {  
   @Override  
   public void onClick(View v) {  
   Iniciar\_sesion();  
   }  
  });  
  rq= Volley.*newRequestQueue*(getContext());
* return vista;
* Donde:
* jbtingresar.setOnClickListener(new View.OnClickListener() {  
   @Override  
   public void onClick(View v) {  
   Iniciar\_sesion();  
   }  
  });
* Asocia el objeto jbtingresar con el evento OnClick y este a su vez lo asocia con el evento Iniciar\_sesion()
* rq= Volley.*newRequestQueue*(getContext());
* Inicializa el objeto rq como una nueva cola de requerimientos

Instrucciones UsuariosActivity

Código Xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout 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=".UsuariosActivity"  
 android:orientation="horizontal">  
  
 <LinearLayout  
 android:layout\_width="409dp"  
 android:layout\_height="729dp"  
 android:orientation="vertical"  
 tools:layout\_editor\_absoluteX="1dp"  
 tools:layout\_editor\_absoluteY="1dp">  
  
 <TextView  
 android:layout\_width="410dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="60dp"  
 android:gravity="center"  
 android:text="Usuarios"  
 android:textSize="40dp" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="70dp"  
 android:layout\_marginTop="40dp">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="20dp"  
 android:layout\_marginTop="20dp"  
 android:text="Usuario"  
 android:textSize="30dp" />  
  
 <EditText  
 android:id="@+id/etusuario"  
 android:layout\_width="240dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="20dp"  
 android:layout\_marginTop="20dp"  
 android:textSize="30dp" />  
  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="70dp">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="20dp"  
 android:layout\_marginTop="20dp"  
 android:text="Nombre"  
 android:textSize="30dp" />  
  
 <EditText  
 android:id="@+id/etnombre"  
 android:layout\_width="240dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="20dp"  
 android:layout\_marginTop="20dp"  
 android:textSize="30dp" />  
  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="70dp">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="20dp"  
 android:layout\_marginTop="20dp"  
 android:text="Correo"  
 android:textSize="30dp" />  
  
 <EditText  
 android:id="@+id/etcorreo"  
 android:layout\_width="260dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="20dp"

android:inputType="textEmailAddress"  
 android:layout\_marginTop="20dp"  
 android:textSize="30dp" />  
  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="70dp">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="20dp"  
 android:layout\_marginTop="20dp"  
 android:text="Clave"  
 android:textSize="30dp" />  
  
 <EditText  
 android:id="@+id/etclave"  
 android:layout\_width="270dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="20dp"

android:inputType="textPassword"  
 android:layout\_marginTop="20dp"  
 android:textSize="30dp" />  
  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="200dp"  
 android:layout\_height="70dp"  
 android:layout\_gravity="center">  
  
 <CheckBox  
 android:id="@+id/cbactivo"  
 android:layout\_width="200dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:enabled="false"  
 android:layout\_weight="1"  
 android:gravity="center"  
 android:text=" Activo"  
 android:textSize="30dp" />  
  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="60dp"  
 android:gravity="center">  
  
  
 <Button  
 android:id="@+id/btnadicionar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Adicionar" />  
  
 <Button  
 android:id="@+id/btnconsultar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="20dp"  
 android:text="Consultar" />  
  
 <Button  
 android:id="@+id/btnmodificar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="20dp"  
 android:text="Modificar" />  
  
  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="50dp"  
 android:layout\_gravity="center"  
 android:gravity="center">  
  
  
 <Button  
 android:id="@+id/btnanular"  
 android:layout\_width="100dp"  
 android:layout\_height="wrap\_content"  
 android:text="Anular" />  
  
 <Button  
 android:id="@+id/btnlimpiar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="20dp"  
 android:text="Limpiar" />  
  
 <Button  
 android:id="@+id/btnregresar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="20dp"  
 android:text="Regresar" />  
  
 </LinearLayout>  
 </LinearLayout>  
</LinearLayout>

Código Java

Objetos EditText’s y Json

EditText jetusuario,jetnombre,jetcorreo,jetclave;  
CheckBox jcbactivo;  
RequestQueue rq;  
JsonRequest jrq;  
String usr,nombre,correo,clave,url;  
byte sw;

Asociar objetos Xml con los objetos Java

// Ocultar barra, asociar objetos, iniciar la cola de consultas  
getSupportActionBar().hide();  
jetusuario=findViewById(R.id.*etusuario*);  
jetnombre=findViewById(R.id.*etnombre*);  
jetcorreo=findViewById(R.id.*etcorreo*);  
jetclave=findViewById(R.id.*etclave*);  
jcbactivo=findViewById(R.id.*cbactivo*);  
rq = Volley.*newRequestQueue*(this);//conexion a internet  
sw=0;

Método para consultar usuarios

Primero la clase debe heredar de Android Studio la librería correspondiente

implements Response.Listener<JSONObject>,Response.ErrorListener

Nota: recuerde, va en el encabezado de la clase y al ser una clase abstracta se le deben implementar los métodos OnResponse y OnErrorResponse

Las instrucciones para capturar la información enviada por Json son las siguientes

public void Consultar\_usuario(View view){  
 usr=jetusuario.getText().toString();  
 if (usr.isEmpty()){  
 Toast.*makeText*(this, "Usuario necesario para consultar", Toast.*LENGTH\_SHORT*).show();  
 jetusuario.requestFocus();  
 }  
 else{  
 url = "http://172.16.59.102:80/Empresa6/consulta.php?usr=" + usr;  
 jrq = new JsonObjectRequest(Request.Method.*GET*, url, null, this, this);  
 rq.add(jrq);  
 }  
  
}  
  
@Override  
public void onErrorResponse(VolleyError error) {  
 Toast.*makeText*(this, "Error consultando usuario", Toast.*LENGTH\_SHORT*).show();  
}  
  
@Override  
public void onResponse(JSONObject response) {  
 sw=1;  
 Toast.*makeText*(this, "Usuario registrado", Toast.*LENGTH\_SHORT*).show();  
 JSONArray jsonArray = response.optJSONArray("datos");  
 JSONObject jsonObject = null;  
 try {  
 jsonObject = jsonArray.getJSONObject(0);//posicion 0 del arreglo....  
 jetnombre.setText(jsonObject.optString("nombre"));  
 jetcorreo.setText(jsonObject.optString("correo"));  
 jetclave.setText(jsonObject.optString("clave"));  
 if (jsonObject.optString("activo").equals("si"))  
 jcbactivo.setChecked(true);  
 else  
 jcbactivo.setChecked(false);  
 }  
 catch (JSONException e)  
 {  
 e.printStackTrace();  
 }  
}

Donde JSONArray es una clase de Android Studio que me recibe la información enviada a través de un arreglo de json (en este caso datos)

Recuerde que el try es una función para manejar la recolección automática de errores y Java y Android Studio la exigen cada vez que se realizan operaciones en una base de datos

Código para adicionar y modificar un registro cliente

public void Adicionar\_usuario(View view){  
 usr=jetusuario.getText().toString();  
 nombre=jetnombre.getText().toString();  
 correo=jetcorreo.getText().toString();  
 clave=jetclave.getText().toString();  
 if (usr.isEmpty() || nombre.isEmpty() || correo.isEmpty() || clave.isEmpty()){  
 Toast.*makeText*(this, "Todos los datos son requeridos", Toast.*LENGTH\_SHORT*).show();  
 jetusuario.requestFocus();  
 }  
 else{  
 if (sw == 0)  
 url = "http://172.16.59.102:80/Empresa6/registrocorreo.php";  
 else {  
 url = "http://172.16.59.102:80/Empresa6/actualiza.php";  
 sw=0;  
 }  
 StringRequest postRequest = new StringRequest(Request.Method.*POST*, url,  
 new Response.Listener<String>()  
 {  
 @Override  
 public void onResponse(String response) {  
 Limpiar\_campos();  
 Toast.*makeText*(getApplicationContext(), "Registro de usuario realizado correctamente!", Toast.*LENGTH\_LONG*).show();  
 }  
 },  
 new Response.ErrorListener()  
 {  
 @Override  
 public void onErrorResponse(VolleyError error) {  
 Toast.*makeText*(getApplicationContext(), "Registro de usuario incorrecto!", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 ) {  
 @Override  
 protected Map<String, String> getParams()  
 {  
 Map<String, String> params = new HashMap<String, String>();  
 params.put("usr",jetusuario.getText().toString().trim());  
 params.put("nombre", jetnombre.getText().toString().trim());  
 params.put("correo",jetcorreo.getText().toString().trim());  
 params.put("clave",jetclave.getText().toString().trim());  
 return params;  
 }  
 };  
 RequestQueue requestQueue = Volley.*newRequestQueue*(this);  
 requestQueue.add(postRequest);  
 }  
}

private void Limpiar\_campos(){  
 jetusuario.setText("");  
 jetnombre.setText("");  
 jetcorreo.setText("");  
 jetclave.setText("");  
 jetusuario.requestFocus();  
}

Código para eliminar un registro

public void Eliminar\_usuario(View view){  
 if (sw == 0)  
 Toast.*makeText*(this, "Debe primero consultar", Toast.*LENGTH\_SHORT*).show();  
 else{  
 usr=jetusuario.getText().toString();  
 if (usr.isEmpty()){  
 Toast.*makeText*(this, "El usuario es requerido", Toast.*LENGTH\_SHORT*).show();  
 jetusuario.requestFocus();  
 }  
 else {  
 sw=0;  
 url = "http://172.16.59.102:80/Empresa6/elimina.php";  
 StringRequest postRequest = new StringRequest(Request.Method.*POST*, url,  
 new Response.Listener<String>() {  
 @Override  
 public void onResponse(String response) {  
 Limpiar\_campos();  
 Toast.*makeText*(getApplicationContext(), "Registro eliminado!", Toast.*LENGTH\_LONG*).show();  
 }  
 },  
 new Response.ErrorListener() {  
 @Override  
 public void onErrorResponse(VolleyError error) {  
 Toast.*makeText*(getApplicationContext(), "Error eliminando registro!", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 ) {  
 @Override  
 protected Map<String, String> getParams() {  
 Map<String, String> params = new HashMap<String, String>();  
 params.put("usr", jetusuario.getText().toString().trim());  
 return params;  
 }  
 };  
 RequestQueue requestQueue = Volley.*newRequestQueue*(this);  
 requestQueue.add(postRequest);  
 }  
 }  
}

Código para anular un registro

public void Anular(View view){  
 if (sw == 0)  
 Toast.*makeText*(this, "Debe primero consultar", Toast.*LENGTH\_SHORT*).show();  
 else{  
 usr=jetusuario.getText().toString();  
 if (usr.isEmpty()){  
 Toast.*makeText*(this, "El usuario es requerido", Toast.*LENGTH\_SHORT*).show();  
 jetusuario.requestFocus();  
 }  
 else {  
 sw=0;  
 url = "http://172.16.59.102:80/Empresa6/anula.php";  
 StringRequest postRequest = new StringRequest(Request.Method.*POST*, url,  
 new Response.Listener<String>() {  
 @Override  
 public void onResponse(String response) {  
 Limpiar\_campos();  
 Toast.*makeText*(getApplicationContext(), "Registro anulado!", Toast.*LENGTH\_LONG*).show();  
 }  
 },  
 new Response.ErrorListener() {  
 @Override  
 public void onErrorResponse(VolleyError error) {  
 Toast.*makeText*(getApplicationContext(), "Error anulando registro!", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 ) {  
 @Override  
 protected Map<String, String> getParams() {  
 Map<String, String> params = new HashMap<String, String>();  
 params.put("usr", jetusuario.getText().toString().trim());  
 return params;  
 }  
 };  
 RequestQueue requestQueue = Volley.*newRequestQueue*(this);  
 requestQueue.add(postRequest);  
 }  
 }  
}

Código Limpiar y regresar

public void Limpiar(View view){  
 Limpiar\_campos();  
}  
  
public void Regresar(View view){  
 Intent intmain=new Intent(this,MainActivity.class);  
 startActivity(intmain);  
}