**Listing Program**

package com.keamanan\_rumah.sistemkeamananrumah;  
  
import android.content.SharedPreferences;  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.os.Handler;  
import android.support.annotation.Nullable;  
import android.support.v4.app.Fragment;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
import android.widget.LinearLayout;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import org.json.JSONArray;  
import org.json.JSONException;  
import org.json.JSONObject;  
  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
import java.util.Timer;  
import java.util.TimerTask;  
  
import static android.content.ContentValues.TAG;  
  
public class FragmentMonitoring extends Fragment {  
  
 LinearLayout llNoNetwork, llNetworkAvailable;  
  
 TextView tvIndoor, tvOutdoor, tvDoorLock,tvStatusSensor, tvInformasiHardware;  
 ImageView ivIndoor, ivOutdoor, ivDoorLock,ivStatusSensor;  
  
 SharedPreferences pref;  
 SharedPreferences.Editor editor;  
  
 boolean loaddata;  
  
 String JSON\_data;  
 String url;  
 String str\_id;  
 String str\_state;  
 String str\_outdoor;  
 String str\_indoor;  
 String str\_ussrf;  
 String str\_magnetic;  
 String str\_datetime;  
 String str\_status\_perangkat;  
 String str\_datetime\_perangkat;  
  
 int ln = 0;  
  
 public static String pref\_id;  
 public static String pref\_username;  
 public static String pref\_nama;  
 public static String pref\_tipe;  
 public static String pref\_api\_key;  
 public static String pref\_secure\_key;  
 public static String pref\_waktu;  
  
 public static String id;  
 public static String api\_daftar;  
 public static String api\_dashboard;  
 public static String api\_profil;  
 public static String api\_update\_profil;  
 public static String api\_update\_password;  
 public static String api\_load\_all\_parent;  
 public static String api\_monitoring;  
 public FragmentMonitoring() {  
 }  
  
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,Bundle savedInstanceState) {  
 View inflaterMonitoring = inflater.inflate(R.layout.fragment\_monitoring, container, false);  
 llNetworkAvailable = (LinearLayout) inflaterMonitoring.findViewById(R.id.llNetworkAvailable);  
 llNoNetwork = (LinearLayout) inflaterMonitoring.findViewById(R.id.llNoNetwork);  
 tvIndoor = (TextView) inflaterMonitoring.findViewById(R.id.tvIndoor);  
 tvOutdoor = (TextView) inflaterMonitoring.findViewById(R.id.tvOutdoor);  
 tvDoorLock = (TextView) inflaterMonitoring.findViewById(R.id.tvDoorLock);  
 tvStatusSensor = (TextView) inflaterMonitoring.findViewById(R.id.tvStatusSensor);  
 tvInformasiHardware = (TextView) inflaterMonitoring.findViewById(R.id.tvInformasiHardware);  
 ivIndoor = (ImageView)inflaterMonitoring.findViewById(R.id.ivIndoor);  
 ivOutdoor = (ImageView)inflaterMonitoring.findViewById(R.id.ivOutdoor);  
 ivDoorLock = (ImageView)inflaterMonitoring.findViewById(R.id.ivDoorLock);  
 ivStatusSensor = (ImageView)inflaterMonitoring.findViewById(R.id.ivStatusSensor);  
 return inflaterMonitoring;  
 }  
  
 @Override  
 public void onViewCreated(final View view, @Nullable Bundle savedInstanceState) {  
 super.onViewCreated(view, savedInstanceState);  
 pref = getActivity().getSharedPreferences("KEAMANAN\_RUMAH", 0);  
 editor = pref.edit();  
  
 pref\_id = pref.getString("ID",null);  
 pref\_username = pref.getString("USERNAME",null);  
 pref\_nama = pref.getString("NAMA",null);  
 pref\_tipe = pref.getString("TIPE",null);  
 pref\_api\_key = pref.getString("API\_KEY",null);  
 pref\_secure\_key = pref.getString("SECURE\_KEY",null);  
 pref\_waktu = pref.getString("WAKTU",null);  
  
 api\_daftar = getResources().getString(R.string.api\_site\_url).concat(getResources().getString(R.string.api\_daftar));  
 api\_dashboard = getResources().getString(R.string.api\_site\_url).concat(getResources().getString(R.string.api\_dashboard));  
 api\_profil = getResources().getString(R.string.api\_site\_url).concat(getResources().getString(R.string.api\_profil)).concat(pref\_id);  
 api\_update\_profil = getResources().getString(R.string.api\_site\_url).concat(getResources().getString(R.string.api\_update\_profil)).concat(pref\_id);  
 api\_update\_password = getResources().getString(R.string.api\_site\_url).concat(getResources().getString(R.string.api\_update\_password)).concat(pref\_id);  
 api\_load\_all\_parent = getResources().getString(R.string.api\_site\_url).concat(getResources().getString(R.string.api\_load\_all\_parent));  
 api\_monitoring = getResources().getString(R.string.api\_site\_url).concat(getResources().getString(R.string.api\_monitoring));  
  
  
 final Handler handler = new Handler();  
 Timer timer = new Timer();  
 TimerTask doAsynchronousTask = new TimerTask() {  
 @Override  
 public void run() {  
 handler.post(new Runnable() {  
 public void run() {  
 try {  
 AsyncMonitoring async= new AsyncMonitoring();  
 async.execute();  
 } catch (Exception e) {  
 }  
 }  
 });  
 }  
 };  
 timer.schedule(doAsynchronousTask, 0, 5000);  
 }  
  
 private class AsyncMonitoring extends AsyncTask<Void, Void, Void> {  
 @Override  
 protected void onPreExecute() {  
 super.onPreExecute();  
 }  
  
 @Override  
 protected Void doInBackground(Void... arg0) {  
 Log.d(TAG, "Do in background");  
 HTTPSvc sh = new HTTPSvc();  
 url = api\_monitoring.concat(pref\_api\_key);  
 JSON\_data = sh.makeServiceCall(url, HTTPSvc.POST);  
 if(JSON\_data!=null){  
 try {  
 JSONObject jsonObj = new JSONObject(JSON\_data);  
 JSONArray response = jsonObj.getJSONArray("response");  
 ln = response.length();  
 if(ln > 0){  
 JSONObject obj\_sensor = response.getJSONObject(0);  
 str\_id = obj\_sensor.getString("id");  
 str\_state = obj\_sensor.getString("state");  
 str\_indoor = obj\_sensor.getString("indoor");  
 str\_outdoor = obj\_sensor.getString("outdoor");  
 str\_ussrf = obj\_sensor.getString("ussrf");  
 str\_magnetic = obj\_sensor.getString("magnetic");  
 str\_datetime = obj\_sensor.getString("datetime");  
 str\_status\_perangkat = obj\_sensor.getString("status\_perangkat");  
 str\_datetime\_perangkat = obj\_sensor.getString("datetime\_perangkat");  
 }  
 } catch (final JSONException e) {  
 Log.e(TAG, e.getMessage());  
 }  
 loaddata=true;  
 }  
 else{  
 loaddata=false;  
 }  
 Log.d(TAG, "JSON data : " + JSON\_data);  
 return null;  
 }  
  
 @Override  
 protected void onPostExecute(Void result) {  
 super.onPostExecute(result);  
 if(loaddata){  
 if(ln > 0){  
 llNetworkAvailable.setVisibility(View.VISIBLE);  
 llNoNetwork.setVisibility(View.GONE);  
 String status\_indoor, status\_outdoor,status\_doorlock;  
 if(str\_indoor.equals("0")){  
 status\_indoor = "AMAN";  
 ivIndoor.setImageResource(R.mipmap.home\_secure);  
 }else{  
 status\_indoor = "Terdeteksi Orang";  
 ivIndoor.setImageResource(R.mipmap.burglar);  
 }  
 if(str\_outdoor.equals("0")){  
 status\_outdoor = "AMAN";  
 ivOutdoor.setImageResource(R.mipmap.home\_secure);  
 }else{  
 status\_outdoor = "Terdeteksi Orang.";  
 ivOutdoor.setImageResource(R.mipmap.burglar);  
 }  
 if(str\_magnetic.equals("0")){  
 status\_doorlock = "AMAN";  
 ivDoorLock.setImageResource(R.mipmap.pintu\_tertutup);  
 }else{  
 status\_doorlock = "Terbuka";  
 ivDoorLock.setImageResource(R.mipmap.pintu\_terbuka);  
 }  
 if(str\_status\_perangkat.equals("1")){  
 tvStatusSensor.setText("Online");  
 tvInformasiHardware.setText("Data terakhir dikirim pada \n" + str\_datetime);  
 }else{  
 tvStatusSensor.setText("Offline");  
 ivIndoor.setImageResource(R.mipmap.no\_data);  
 ivOutdoor.setImageResource(R.mipmap.no\_data);  
 ivDoorLock.setImageResource(R.mipmap.no\_data);  
 tvInformasiHardware.setText("Sensor offline sejak \n" + str\_datetime\_perangkat);  
 status\_indoor = "Sensor Offline";  
 status\_outdoor = "Sensor Offline";  
 status\_doorlock = "Sensor Offline";  
 }  
 tvIndoor.setText(status\_indoor);  
 tvOutdoor.setText(status\_outdoor);  
 tvDoorLock.setText(status\_doorlock);  
 String sekarang = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(new Date());  
 SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");  
 try {  
 Date date1 = simpleDateFormat.parse(str\_datetime);  
 Date date2 = simpleDateFormat.parse(sekarang);  
 long different = date2.getTime() - date1.getTime();  
 long secondsInMilli = 1000;  
 long minutesInMilli = secondsInMilli \* 60;  
 long hoursInMilli = minutesInMilli \* 60;  
 long daysInMilli = hoursInMilli \* 24;  
 long elapsedDays = different / daysInMilli;  
 different = different % daysInMilli;  
 long elapsedHours = different / hoursInMilli;  
 different = different % hoursInMilli;  
 Log.d("diference : ",String.valueOf(different));  
 if(different > 60000 && str\_status\_perangkat.equals("1")){  
 long elapsedMinutes = different / minutesInMilli;  
 different = different % minutesInMilli;  
 long elapsedSeconds = different / secondsInMilli;  
 Log.d("Hari : ",String.valueOf(elapsedDays));  
 Log.d("Jam : ",String.valueOf(elapsedHours));  
 Log.d("Menit : ",String.valueOf(elapsedMinutes));  
 Log.d("Detik : ",String.valueOf(elapsedSeconds));  
 tvInformasiHardware.setText(  
 "Harap periksa kondisi sensor." + "\n" +  
 "Data terakhir dikirim pada \n" + str\_datetime\_perangkat + ".\n" +  
 "Sensor tidak mengirimkan data selama : " + "\n" +  
 String.valueOf(elapsedDays) + " Hari" + "\n" +  
 String.valueOf(elapsedHours) + " Jam" + "\n" +  
 String.valueOf(elapsedMinutes) + " Menit" + "\n" +  
 String.valueOf(elapsedSeconds) + " Detik");  
 }  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 }else{  
 ivIndoor.setImageResource(R.mipmap.no\_data);  
 ivOutdoor.setImageResource(R.mipmap.no\_data);  
 ivDoorLock.setImageResource(R.mipmap.no\_data);  
 tvIndoor.setText("Tidak ada data");  
 tvOutdoor.setText("Tidak ada data");  
 tvDoorLock.setText("Tidak ada data");  
 tvInformasiHardware.setText("Tidak ada data");  
 }  
 }else {  
 llNetworkAvailable.setVisibility(View.GONE);  
 llNoNetwork.setVisibility(View.VISIBLE);  
 }  
 }  
 }  
}