ALERT DIALOG BASIC

# 

# //Alert Dialog

AlertDialog dlg = new AlertDialog.Builder(this)

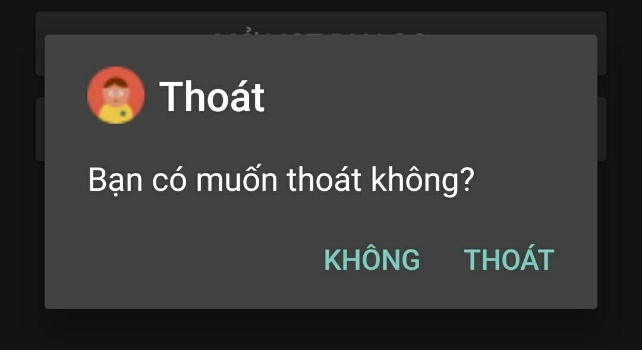
.setTitle("Error Message")

.setIcon(R.drawable.male\_24)

.setMessage("You have error. That is all")

.create();

dlg.show();



# /\*Promt Dialog - hỏi thoát ứng dụng\*/

AlertDialog.Builder builder = new AlertDialog.Builder(this);

builder.setTitle("Thoát"); // tiêu đề

builder.setIcon(R.drawable.male\_24); // ảnh tiêu đề

builder.setMessage("Bạn có muốn thoát không?"); // tin nhắn

// nút khẳng định

builder.setPositiveButton("Thoát", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

finish();

}

});

// nút phủ định

builder.setNegativeButton("Không", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

// hủy

dialog.cancel();

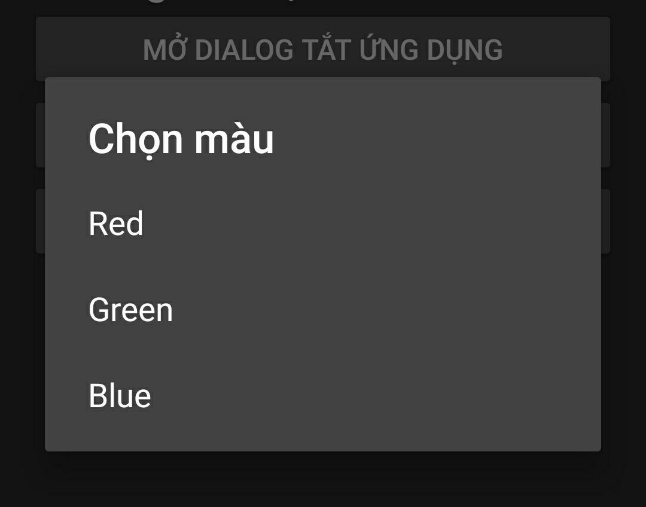
}

});

// hiện builder

Dialog dialog = builder.create();

dialog.show();



# /\*Select List Dialog - Hiện list màu sắc\*/

AlertDialog.Builder builder = new AlertDialog.Builder(this);

builder.setTitle("Chọn màu");

// đặt list array

builder.setItems(R.array.array\_color, new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int which) {

// The 'which' argument contains the index position of the selected item

String[] arrayColor = getResources().getStringArray(R.array.array\_color);

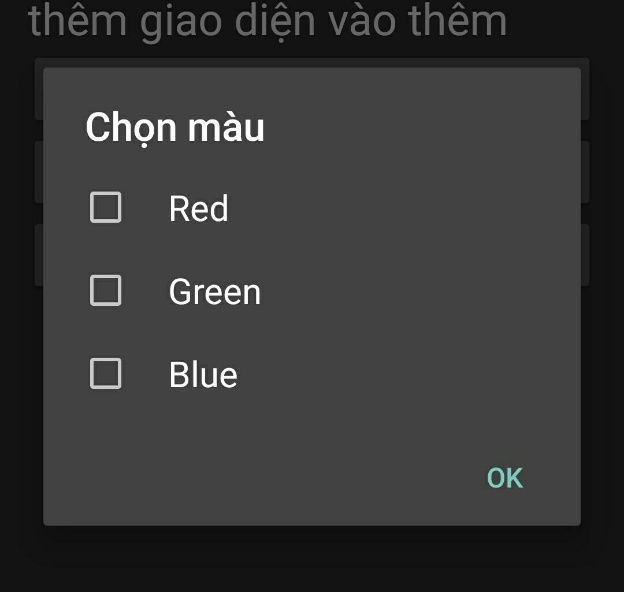
String message = String.format("Vị trí: %s - Giá trị: %s", which, arrayColor[which]);

Toast.makeText(getBaseContext(), message, Toast.LENGTH\_LONG).show();

}

});

builder.create().show();



# /\*Check Box List Dialog - chọn màu sắt\*/

final String[] arrayColor = getResources().getStringArray(R.array.array\_color);

final ArrayList<String> arraySelectedColor = new ArrayList<>();

AlertDialog.Builder builder = new AlertDialog.Builder(this);

builder.setTitle("Chọn màu");

// đặt multiple-choise list array

builder.setMultiChoiceItems(R.array.array\_color, null,

new DialogInterface.OnMultiChoiceClickListener() {

@Override

public void onClick(DialogInterface dialog, int which, boolean isChecked) {

if (isChecked) {

// check item => add to list

arraySelectedColor.add(arrayColor[which]);

} else {

// not check item => remove

arraySelectedColor.remove(arrayColor[which]);

}

}

}

);

// button OK

builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

// hiện các giá trị dã chọn

String messsage = "List: ";

for (String s: arraySelectedColor){

messsage += s + " - ";

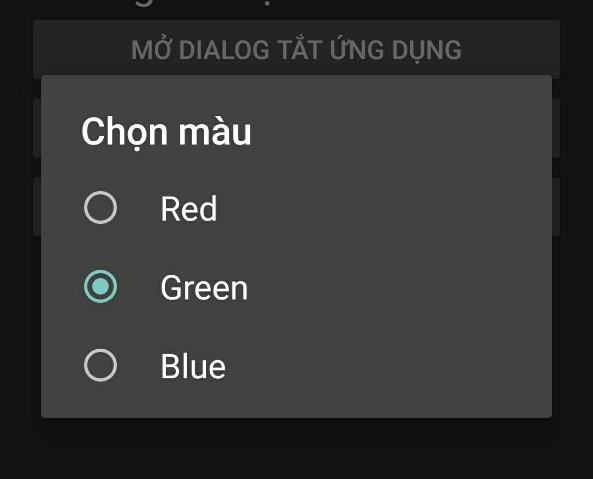
}

Toast.makeText(getBaseContext(), messsage, Toast.LENGTH\_LONG).show();

}

});

builder.create().show();



# //Single Choose, Chọn 1

builder.setSingleChoiceItems(R.array.array\_color, 1,

new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface di, int what){

//...

}

});

# //Custom Layout Dialog- Giao diện Xml

// mở rộng và cài đặt Layout cho Dialog

LayoutInflater inflater = this.getLayoutInflater();

final View dialogView = inflater.inflate(R.layout.custom\_view\_login, null);

AlertDialog.Builder builder = new AlertDialog.Builder(this)

//set XML view

.setView(dialogView)

.setPositiveButton("Đăng nhập", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int id) {

AlertDialog alertDialog = (AlertDialog)dialog;

//set login interface, dùng dialogView cũng được

EditText txtUsername = (EditText) alertDialog.findViewById(R.id.username);

EditText txtPassword = (EditText) alertDialog.findViewById(R.id.password);

String message = String.format("Username: %s - Password: %s", txtUsername.getText().toString(), txtPassword.getText().toString());

Toast.makeText(getBaseContext(), message, Toast.LENGTH\_LONG).show();

}

});

builder.create().show();

//AnalogClock, Chronometor, DatePickerDialog, TimePickerDialog

# //Datepicker Dialog

Calendar calendar = Calendar.getInstance();

DatePickerDialog datePickerDialog = new DatePickerDialog(

this,

new DatePickerDialog.OnDateSetListener() {

@Override

public void onDateSet(DatePicker datePicker, int year, int monthOfYear, int dayOfMonth) {

Toast.makeText(getApplicationContext(),

year+"/"+monthOfYear, Toast.LENGTH\_SHORT).show();

}

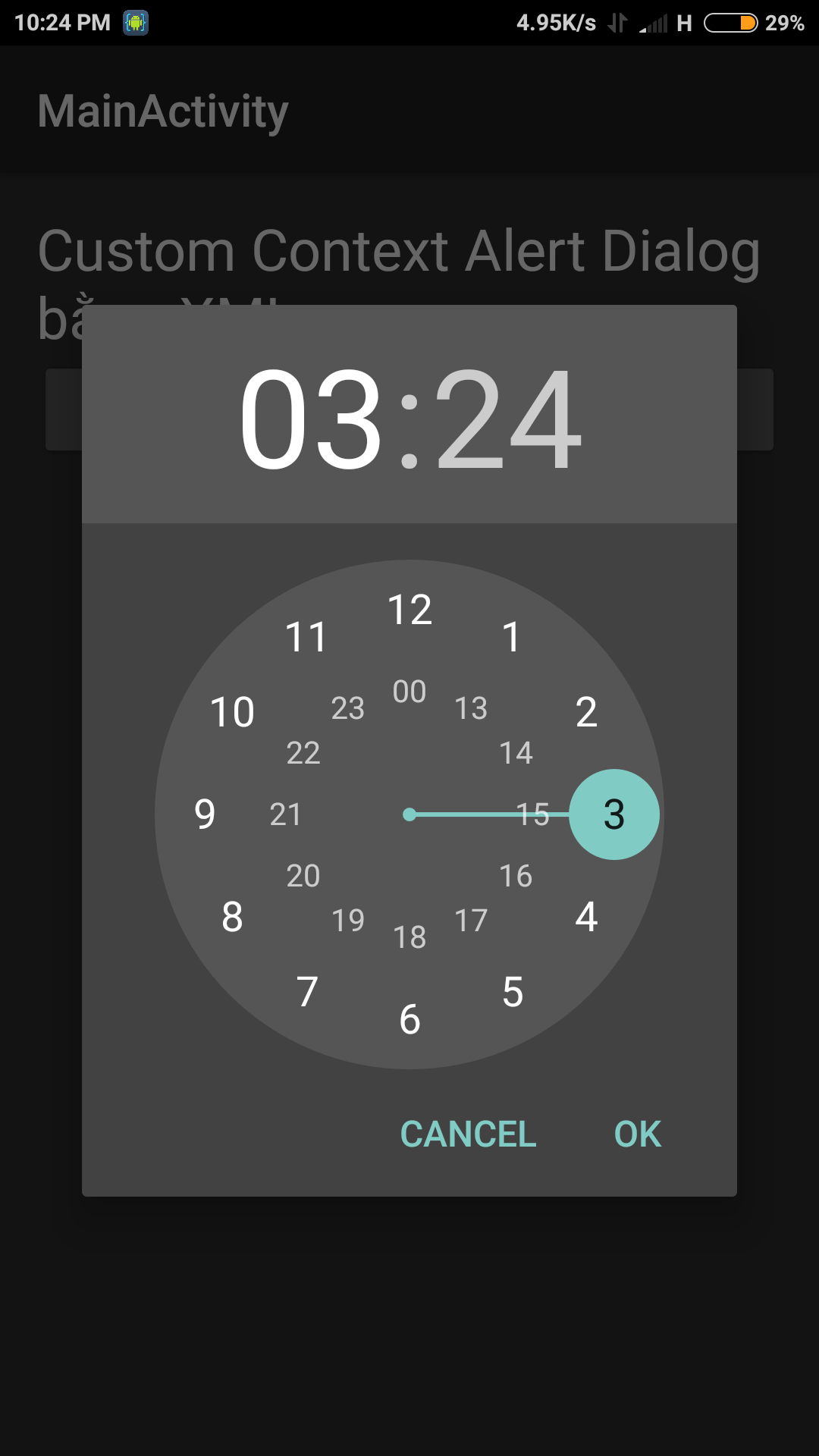
},

calendar.get(Calendar.YEAR),

calendar.get(Calendar.MONTH),

calendar.get(Calendar.DAY\_OF\_MONTH)

);



# //TimePickerDialog

Calendar calendar = Calendar.getInstance();

TimePickerDialog timePickerDialog = new TimePickerDialog(

this,

new TimePickerDialog.OnTimeSetListener() {

@Override

public void onTimeSet(TimePicker timePicker, int hour, int minute) {

Toast.makeText(getApplicationContext(),

hour+":"+minute, Toast.LENGTH\_SHORT).show();

}

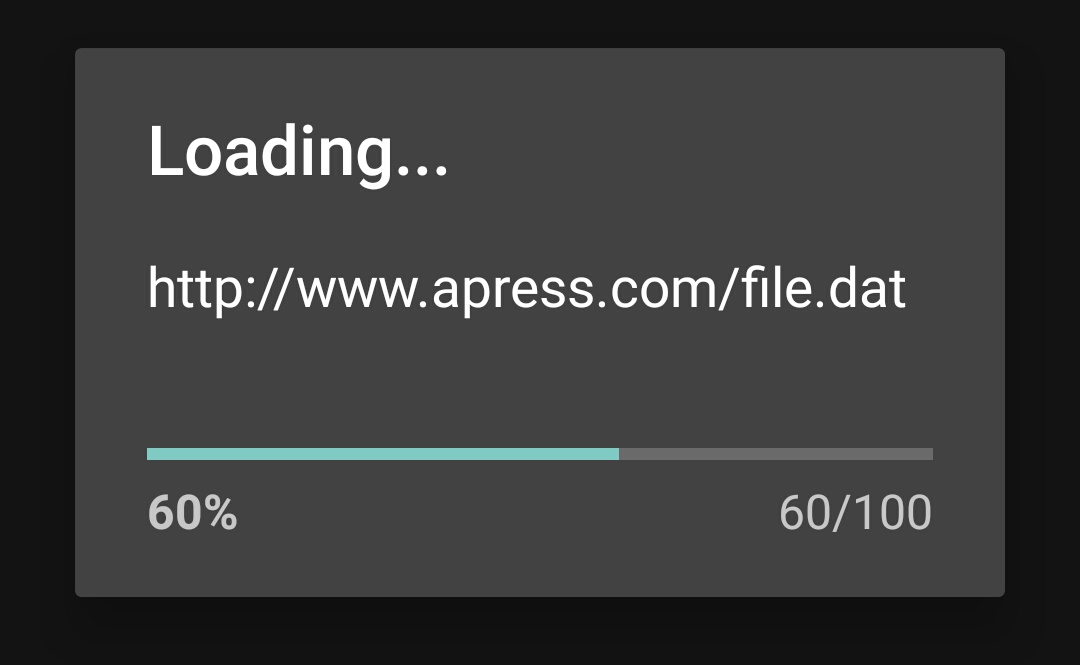
},

calendar.get(Calendar.HOUR),

calendar.get(Calendar.MINUTE),

true

);



# //Process Dialog

ProgressDialog progressDialog = new ProgressDialog(this);

progressDialog.setProgressStyle(

ProgressDialog.STYLE\_HORIZONTAL);

progressDialog.setTitle("Loading...");

progressDialog.setMessage("http://www.apress.com/file.dat");

progressDialog.setMax(100);

progressDialog.incrementProgressBy(60);

//fragment dialog : handles the life-cycle event