Fragment Dialog

handle the necessary life-cycle events directly.

Make dialog Template, dùng được nhiều lần.

# 

# MAKE FRAGMENT DIALOG SIMPLE

DialogFragment dialogFragment = new DialogFragment() {

@Override

public Dialog onCreateDialog(Bundle savedInstanceState) {

AlertDialog alertDialog =

new AlertDialog.Builder(getActivity())

.setTitle("Dialog Title")

.setMessage("Do you want to save?")

.create();

return alertDialog;

}

};

FragmentManager manager = getFragmentManager();

dialogFragment.show(manager, "nghia\_tag");

# 

# MAKE FRAGMENT DIALOG TEMPLATE

**File: fragment\_edit\_name.xml**

<LinearLayout... android:id="@+id/edit\_name">

<TextView... android:id="@+id/lbl\_your\_name"/>

<EditText... android:id="@+id/txt\_your\_name"/>

</LinearLayout>

**File: FragmentEditNameDialog.java**

public class FragmentEditNameDialog extends DialogFragment {

// --- Variables ---

private EditText mEditText;

IDialogListener mListener;

//--- Constructors ---

// khởi tạo Fragment với Title

public static FragmentEditNameDialog newInstance(String title) {

// khởi tạo Listener truyền dữ liệu

FragmentEditNameDialog frag = new FragmentEditNameDialog();

Bundle bundle = new Bundle();

bundle.putString("title", title);

frag.setArguments(bundle);

return frag;

}

// khởi tạo listener - do không cho tạo Constructor

public void setListener(IDialogListener listener){

mListener = listener;

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {

return super.onCreateView(inflater, container, savedInstanceState);

}

@Override

public void onActivityCreated(Bundle savedInstanceState) {

super.onActivityCreated(savedInstanceState);

}

@Override

public Dialog onCreateDialog(Bundle savedInstanceState) {

// khởi tạo Fragment View Dialog

// find Views

View view = getActivity().getLayoutInflater().inflate(R.layout.fragment\_edit\_name, null);

mEditText = (EditText) view.findViewById(R.id.txt\_your\_name);

// Khởi tạo giao diện Dialog

AlertDialog dialog = new AlertDialog.Builder(getActivity())

.setTitle(getArguments().getString("title", "Enter Name"))

.setView(view)

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

@Override

public void onClick(DialogInterface dialog, int which) {

// send success

mListener.positiveButton\_onClick(mEditText.getText().toString());

}

})

.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

dialog.dismiss();

}

}).create();

return dialog;

}

/\*\*

\* Interface trung gian

\*/

public interface IDialogListener {

public void positiveButton\_onClick(String value);

}

}

**Sử dụng trong Activity:**

// hiển thị Dialog, mListener nhận giá trị trả về

FragmentManager manager = getFragmentManager();

FragmentEditNameDialog editNameDialog = FragmentEditNameDialog.newInstance("Tiêu đề khỉ khô");

editNameDialog.setListener(mListener);

editNameDialog.show(manager, "tag");

# Select Time Dialog Template

**File: TimePickerFragment.java**

public class TimePickerFragment extends DialogFragment {

IListener mListener;

// khởi tạo dữ liệu ban đầu

public static TimePickerFragment newInstance(int hourOfDay, int minute){

TimePickerFragment fragment = new TimePickerFragment();

Bundle bundle = new Bundle();

bundle.putInt("hour\_of\_day", hourOfDay);

bundle.putInt("minute", minute);

fragment.setArguments(bundle);

return fragment;

}

// khởi tạo IListener

public void setListener(IListener listener){

this.mListener = listener;

}

@Override

public Dialog onCreateDialog(Bundle savedInstanceState) {

// lấy giá trị mặc định truyền vào,

// không có => giờ mặc định là hiện tại

final Calendar calendar = Calendar.getInstance();

Bundle bundle = this.getArguments();

int hour = bundle.getInt("hour\_of\_day", calendar.get(Calendar.HOUR\_OF\_DAY));

int minute = bundle.getInt("minute", calendar.get(Calendar.MINUTE));

TimePickerDialog.OnTimeSetListener listener = new TimePickerDialog.OnTimeSetListener() {

@Override

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

// Làm gì đó khi chọn xong thời gian

// trả về listener

mListener.onTimeSet(timePicker, hourOfDay, minute);

}

};

// Tạo khởi tạo TimePickerDialog trả về

TimePickerDialog timeDialog = new TimePickerDialog(

getActivity(),

listener,

hour, minute,

DateFormat.is24HourFormat(getActivity()));

return timeDialog;

}

interface IListener {

/\*\* Đặt thời gian \*/

public void onTimeSet(TimePicker timePicker, int hourOfDay, int minute);

}

}

**Xử dụng trong Activity:**

int mHourOfDay = 0;

int mMinute = 0;

public void showTimePickerDialog(View view) {

final Button button = (Button)view;

TimePickerFragment timePicker = TimePickerFragment.newInstance(mHourOfDay, mMinute);

timePicker.setListener(new TimePickerFragment.IListener() {

@Override

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

mHourOfDay = hourOfDay;

mMinute = minute;

button.setText(hourOfDay + ":"+ minute);

}

});

timePicker.show(getFragmentManager(), "tag");

}