noteViewModel.getIdUpdate().observe(this, new Observer<Integer>() {  
 @Override  
 public void onChanged(Integer integer) {  
 Log.*d*("BBB", "Id update " + integer);  
 }  
});  
  
noteViewModel.getIdInsert().observe(this, new Observer<Long>() {  
 @Override  
 public void onChanged(Long aLong) {  
 Log.*d*("BBB", "Id insert " + aLong);  
 }  
});

//observe size, chỉ thực hiện 1 lần, chỉ thực hiện lại nếu listNoteSize change  
// noteViewModel.getListNoteSize().observe(this, new Observer<Long>() {  
// @Override  
// public void onChanged(Long aLong) {  
// mListNoteSize = aLong;  
// mTotalPage =  
// (int) (mListNoteSize / mNotePerPage)  
// + (mListNoteSize % mNotePerPage == 0 ? 0 : 1);  
// // call data  
// noteViewModel.queryGetListNote(Long.MAX\_VALUE, mNotePerPage);  
// }  
// });

// //Insert thủ công  
// noteViewModel.insertNote(new NoteEntity("Work 5",  
// "Do 5",  
// new SimpleDateFormat("dd MMM, yyyy - HH:mm").format(new Date())));  
  
// //Insert data  
// mBtnInsert.setOnClickListener(new View.OnClickListener() {  
// @Override  
// public void onClick(View view) {  
// noteViewModel.insertNote(new NoteEntity(mEditTitle.getText().toString(),  
// mEditDescription.getText().toString(),  
// new SimpleDateFormat("dd MMM, yyyy - HH:mm").format(new Date())));  
// }  
// });  
//  
// //Delete data  
// mBtnDelete.setOnClickListener(new View.OnClickListener() {  
// @Override  
// public void onClick(View v) {  
// noteViewModel.deletedNote(mNoteEntities.get(mNoteEntities.size() - 1));  
// }  
// });  
//  
// //Update data  
// mBtnUpdate.setOnClickListener(new View.OnClickListener() {  
// @Override  
// public void onClick(View v) {  
// NoteEntity noteEntity = mNoteEntities.get(mNoteEntities.size() - 1);  
// noteEntity.setDescription("Do something 3.1");  
// noteEntity.setTitle("Work 3.1");  
// noteEntity.setTimestamp(new SimpleDateFormat("dd MMM, yyyy - HH:mm").format(new Date()));  
// noteViewModel.updateNote(noteEntity);  
// }  
// });

// private void loadNextPage(){  
// if (mCurrentPage > 1){  
// mNoteAdapter.removeLoading();  
// }  
// mNoteEntities.addAll(getMock());  
// mNoteAdapter.notifyDataSetChanged();  
// mLoading = false;  
//  
// addFooter();  
// }

//Insert Note  
 mInsertFAB.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 Intent intent = new Intent(MainActivity.this, InsertEditNoteActivity.class);  
 startActivity(intent);  
// finish();  
 }  
 });  
  
 //Update note (click vào item)  
 mNoteAdapter.setOnClickItemListener(new NoteClickInterface() {  
 @Override  
 public void onNoteClick(NoteEntity noteEntity) {  
 Intent intent = new Intent(MainActivity.this, InsertEditNoteActivity.class);  
 intent.putExtra("noteType", "Edit");  
 intent.putExtra("noteTitle", noteEntity.getTitle());  
 intent.putExtra("noteDescription", noteEntity.getDescription());  
 intent.putExtra("noteId", noteEntity.getId());  
 startActivity(intent);  
 }  
 });  
  
 //Delete  
// mNoteAdapter.setOnDeleteItemListener(new NoteClickDeleteInterface() {  
// @Override  
// public void onDeleteIconClick(NoteEntity noteEntity) {  
// Toast.makeText(  
// MainActivity.this,  
// "Xóa thẻ " + noteEntity.getTitle(),  
// Toast.LENGTH\_SHORT)  
// .show();  
////  
// noteViewModel.deletedNote(noteEntity);  
// mNoteEntities.remove(noteEntity);  
// mNoteAdapter.notifyItemRemoved(noteEntity);  
// changeType = ChangeType.DELETE;  
// }  
// });  
 //Delete  
 mNoteAdapter.setOnDeleteItemListener(new NoteClickDeleteInterface() {  
 @Override  
 public void onDeleteIconClick(int position, NoteEntity noteEntity) {  
 Toast.*makeText*(  
 MainActivity.this,  
 "Xóa thẻ " + noteEntity.getTitle(),  
 Toast.*LENGTH\_SHORT*)  
 .show();  
 //Delete item trong bảng  
 noteViewModel.deletedNote(noteEntity);  
 //Cập nhật RecyclerView trực tiếp, không thông qua giá trị bảng ở noteViewModel.getListNote().observe  
 mNoteEntities.remove(noteEntity);  
 mNoteAdapter.notifyItemRemoved(position);  
 changeType = ChangeType.*DELETE*;  
 }  
 });

private List<NoteEntity> getMock() {  
 return new ArrayList<>(Arrays.*asList*(  
 new NoteEntity("W6", "D6", "T6"),  
 new NoteEntity("W7", "D7", "T7"),  
 new NoteEntity("W8", "D8", "T8"),  
 new NoteEntity("W9", "D9", "T9"),  
 new NoteEntity("W10", "D10", "T10")  
 ));  
}

@Override  
protected void onStart() {  
 super.onStart();  
 Log.*d*("BBB", "onStart: ");  
}  
  
@Override  
protected void onResume() {  
 super.onResume();  
 Log.*d*("BBB", "onResume: ");  
}  
  
@Override  
protected void onPause() {  
 super.onPause();  
 Log.*d*("BBB", "onPause: ");  
}  
  
@Override  
protected void onStop() {  
 super.onStop();  
 Log.*d*("BBB", "onStop: ");  
}  
  
@Override  
protected void onDestroy() {  
 super.onDestroy();  
 Log.*d*("BBB", "onDestroy: ");  
}