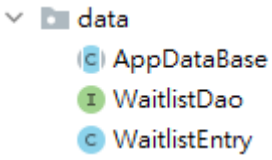
行動裝置_hw2

本次程式碼大多參考 T09b.06 完成。

data、AppExecutors



創建符合 Room 需求之 AppDataBase、Dao、Entity，並修改 Entity 變數名稱以符合對接。

```java
@Entity(tableName = "waitlist") //改 tableName (若 class 和 tableName 要不同名稱)
public class WaitlistEntry {

    @PrimaryKey(autoGenerate = true)
    private int id;
    private String guest_name;
    private String guest_party_size;

    @Ignore
    //忽略自動新增
    public WaitlistEntry(String guest_name,String guest_party_size){
        this.guest_name = guest_name;
        this.guest_party_size = guest_party_size;
    }

    public WaitlistEntry(int id,String guest_name,String guest_party_size){
        this.id = id;
        this.guest_name = guest_name;
        this.guest_party_size = guest_party_size;
    }
```

TaskAdapter

修改變數名稱、對接、Set、Get、Listener，並刪除未使用到的程式碼

```
@Override
public void onBindViewHolder(TaskViewHolder holder, int position) {
    // Determine the values of the wanted data
    WaitlistEntry taskEntry = mWaitlistEntries.get(position);

    String guestname = taskEntry.getGuest_name();
    String partysize = taskEntry.getGuest_party_size();

    //Set values
    holder.GuestNameView.setText(guestname);
    holder.GuestPartySizeView.setText(partysize);
}

// Inner class for creating ViewHolders
class TaskViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener {

    // Class variables for the task description and priority TextViews
    TextView GuestNameView;
    TextView GuestPartySizeView;

    public TaskViewHolder(View itemView) {
        super(itemView);

        GuestNameView = (TextView) itemView.findViewById(R.id.name_text_view);
        GuestPartySizeView = (TextView) itemView.findViewById(R.id.party_size_text_view);
        itemView.setOnClickListener(this);
    }
```

MainActivity

修改 findViewById 對接、變數名，改寫原 Waitlist 的 addToWaitList 用 executer 的方式新增 task。

```
public void addToWaitlist(View view) {
    if (mNewGuestNameEditText.getText().length() == 0 || mNewPartySizeEditText.getText().length() == 0) {
        return;
    }

    String GuestName = mNewGuestNameEditText.getText().toString();
    String PartySize = mNewPartySizeEditText.getText().toString();

    final WaitlistEntry task = new WaitlistEntry(GuestName,PartySize);
    AppExecutors.getInstance().diskIO().execute(new Runnable() {
        @Override
        public void run() {
            mDb.taskDao().insertTask(task);
        }
    });

    retrieveTasks();
    //clear UI text fields
    mNewPartySizeEditText.clearFocus();
    mNewGuestNameEditText.getText().clear();
    mNewPartySizeEditText.getText().clear();
}
```

build.gradle

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'

    // Add RecyclerView dependency; must match SDK version
    implementation 'com.android.support:recyclerview-v7:28.0.0'

    // Add FAB dependency
    implementation 'com.android.support:design:28.0.0'

    implementation "android.arch.persistence.room:runtime:1.1.1"
    annotationProcessor "android.arch.persistence.room:compiler:1.1.1" // 可改成jetpacks

    // Testing
    // Instrumentation dependencies use androidTestImplementation
    // (as opposed to testImplementation for local unit tests run in the JVM)
    androidTestImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support:support-annotations:28.0.0'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test:rules:1.0.2'
}
```