Programmation mobile

Cours 2 : `Intent`

Julien Grange <julien.grange@lacl.fr>

Mardi 23 septembre 2025

UPEC
UNIVERSITÉ PARIS-EST CRÉTEIL

Rappel : lancer une `Activity` sans attendre de résultat

Dans l'`Activity` appelante :

```
int value1 = ...;
String value2 = ...;

// Intent explicite
Intent intent = new Intent(context, NewActivity.class);
intent.putExtra("Key1",value1);
intent.putExtra("Key2",value2);

startActivity(intent);
```

Dans `NewActivity` :

```
Intent intent = getIntent();

int myInt = intent.getIntExtra("Key1",42);
String myString = intent.getStringExtra("Key2");
```

Quatre façons de lancer une `Activity`

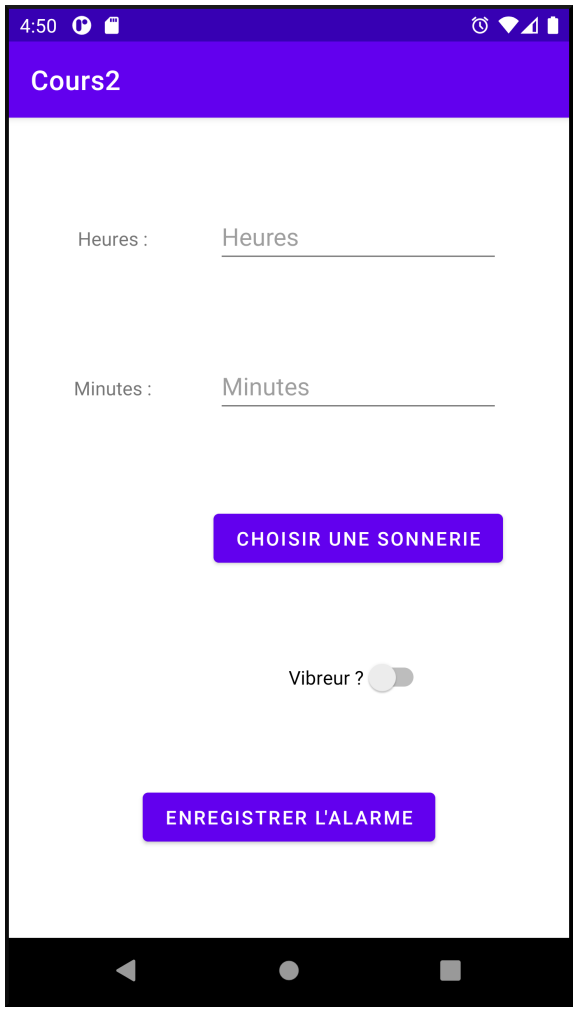
Deux types d'`Intent` :

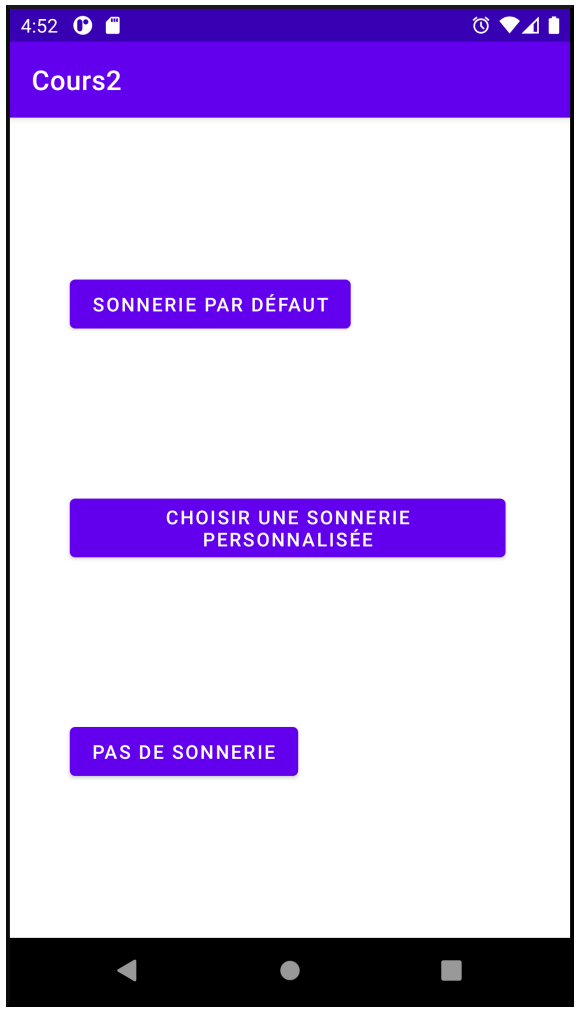
- explicite (en nommant l'`Activity`)
- implicite (en spécifiant le type d'action)

Deux manières de lancer une `Activity` :

- sans attendre de résultat : `startActivity()`
- dans l'attente d'un résultat : `registerForActivityResult()`

Application du jour : Alarme





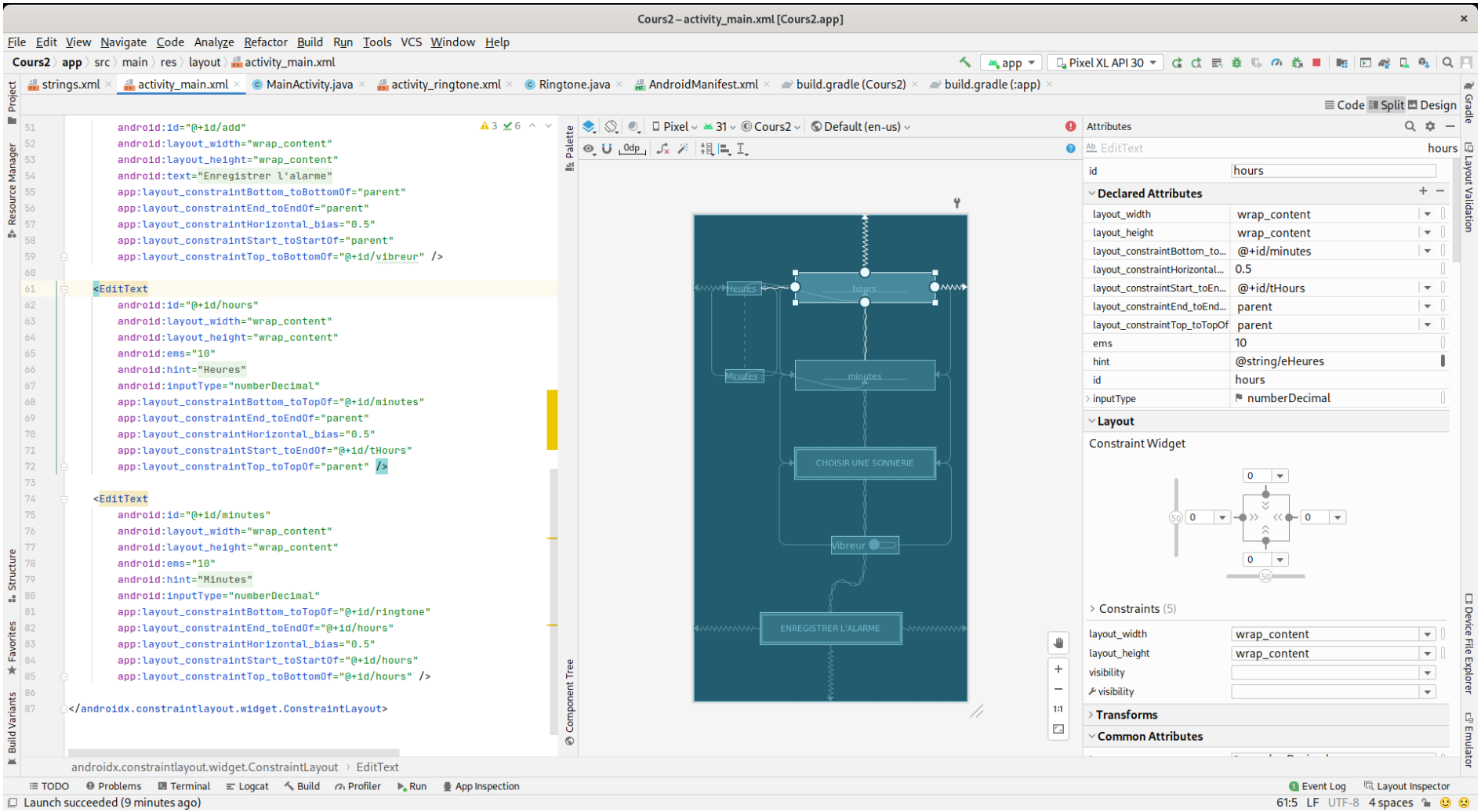
Application du jour : Alarme

Trois appels à une autre `Activity` :

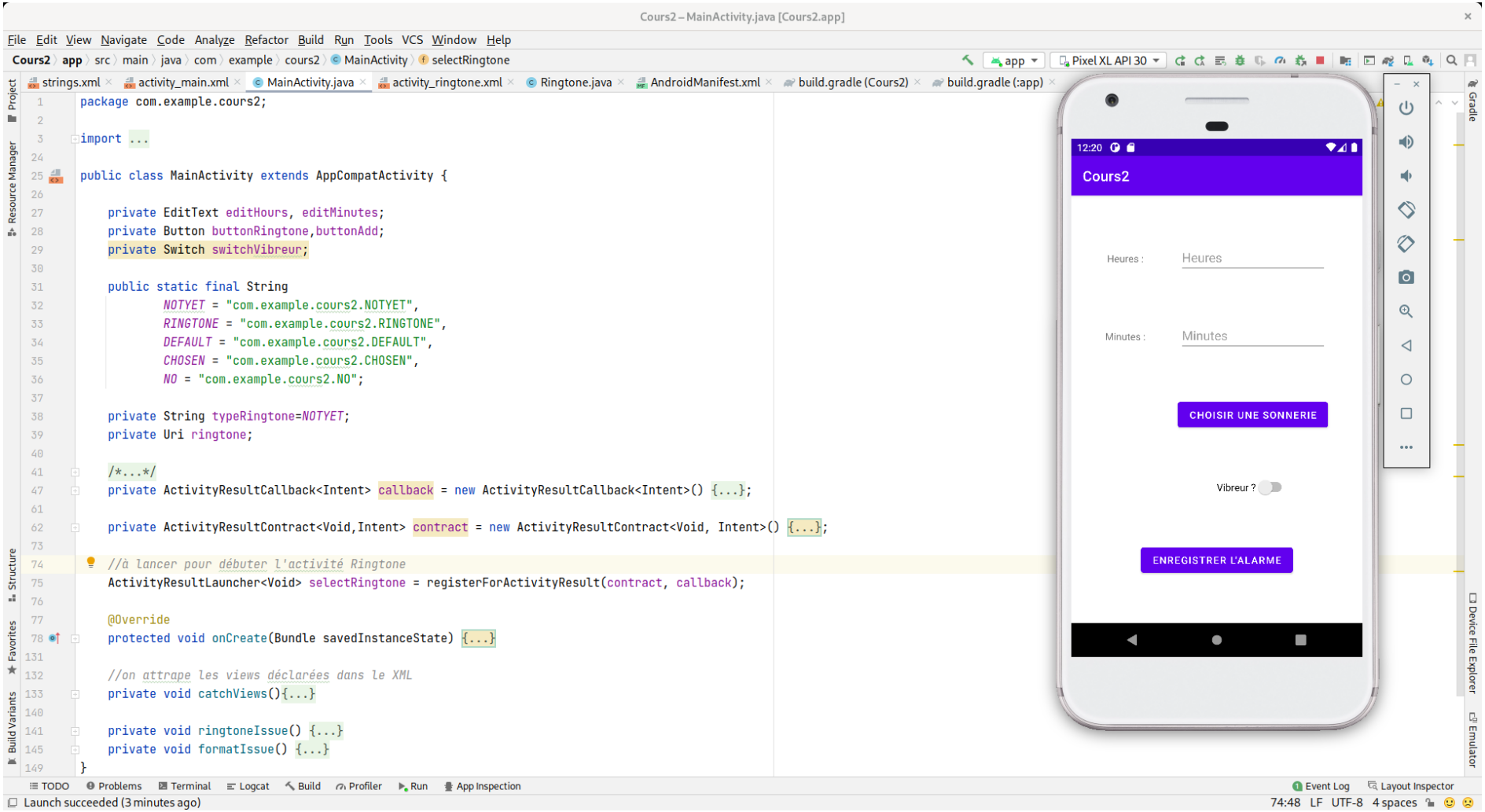
1. “Choisir une sonnerie” : explicite, avec résultat
2. “Choisir une sonnerie personnalisée” : implicite, avec résultat
3. “Enregistrer l’alarme” : implicite, sans résultat

On a déjà vu les appels explicites sans résultat.

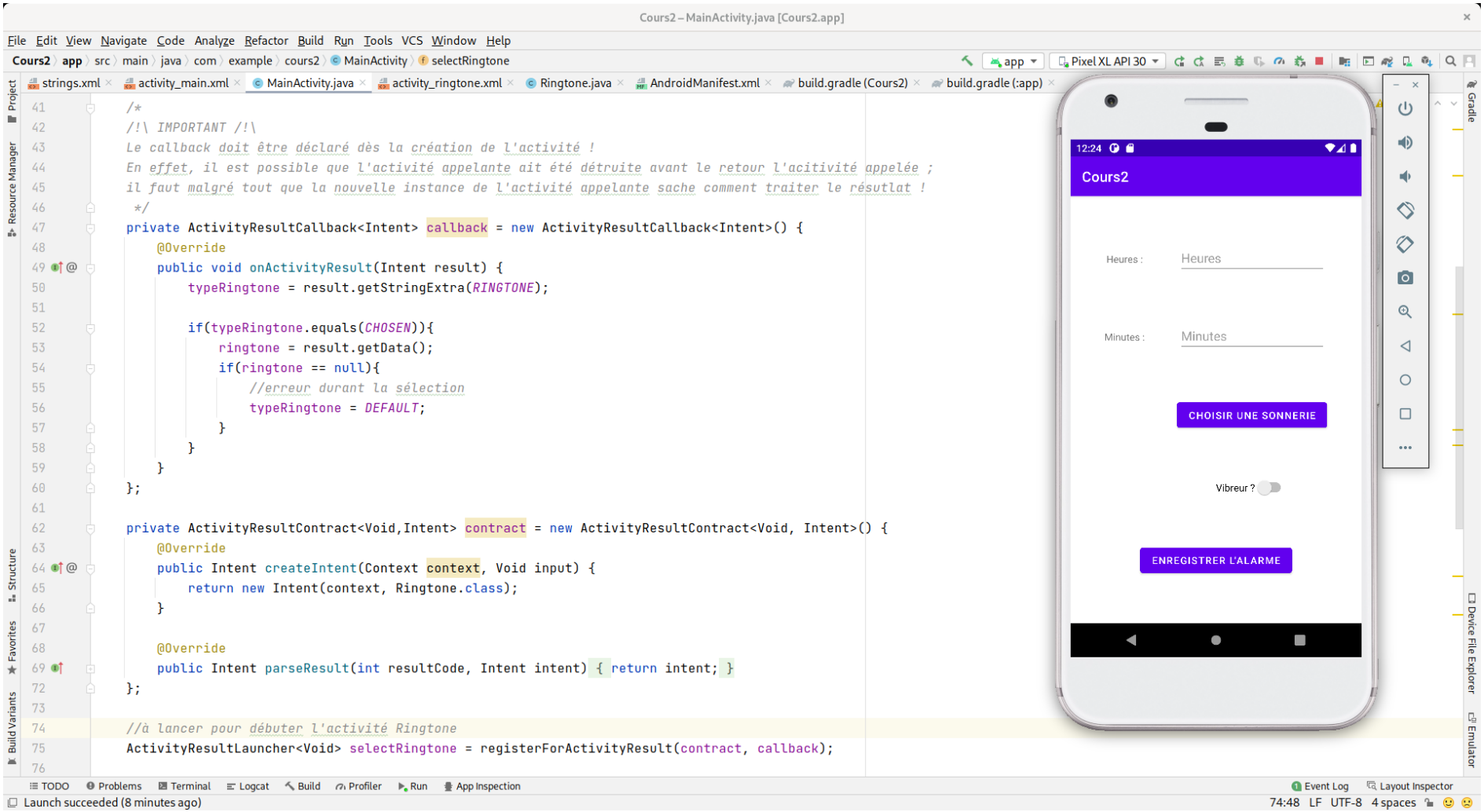
MainActivity layout



MainActivity code



registerForActivityResult()



registerForActivityResult()

Paramétré par deux classes : `I` (input) et `O` (output)

```
public ActivityResultLauncher<I> registerForActivityResult
    (ActivityResultContract<I,O> contract,
     ActivityResultCallback<O> callback)
```

registerForActivityResult()

Paramétré par deux classes : `I` (input) et `O` (output)

```
public ActivityResultLauncher<I> registerForActivityResult
   (ActivityResultContract<I,O> contract,
    ActivityResultCallback<O> callback)
```

- callback : implémente l'interface `ActivityResultCallback<O>`

```
public void onActivityResult(O result)
```

registerForActivityResult()

Paramétré par deux classes : `I` (input) et `O` (output)

```
public ActivityResultLauncher<I> registerForActivityResult
   (ActivityResultContract<I,O> contract,
    ActivityResultCallback<O> callback)
```

- callback : implémente l'interface `ActivityResultCallback<O>`

```
public void onActivityResult(O result)
```

- contract : implémente l'interface `ActivityResultContract<I,O>`

```
public Intent createIntent(Context context, I input)
public O parseResult(int resultCode, Intent intent)
```

registerForActivityResult()

Paramétré par deux classes : `I` (input) et `O` (output)

```
public ActivityResultLauncher<I> registerForActivityResult
   (ActivityResultContract<I,O> contract,
    ActivityResultCallback<O> callback)
```

- callback : implémente l'interface `ActivityResultCallback<O>`

```
public void onActivityResult(O result)
```

- contract : implémente l'interface `ActivityResultContract<I,O>`

```
public Intent createIntent(Context context, I input)
public O parseResult(int resultCode, Intent intent)
```

- l'appel peut ensuite être fait via la méthode `launch(I input)` du résultat de `registerForActivityResult()`

registerForActivityResult()

Paramétré par deux classes : `I` (input) et `O` (output)

```
public ActivityResultLauncher<I> registerForActivityResult
   (ActivityResultContract<I,O> contract,
    ActivityResultCallback<O> callback)
```

- callback : implémente l'interface `ActivityResultCallback<O>`

```
public void onActivityResult(O result)
```

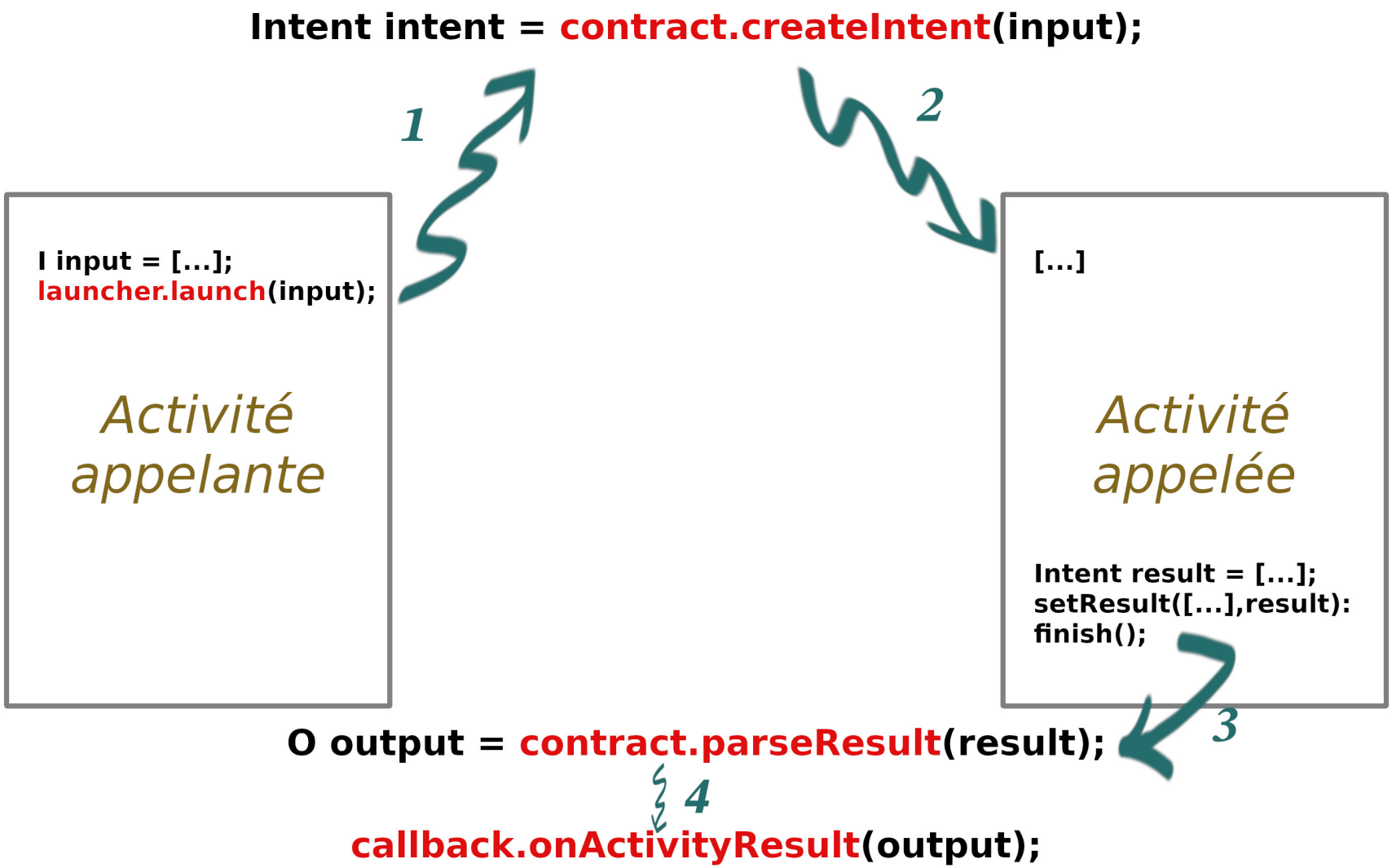
- contract : implémente l'interface `ActivityResultContract<I,O>`

```
public Intent createIntent(Context context, I input)
public O parseResult(int resultCode, Intent intent)
```

- l'appel peut ensuite être fait via la méthode `launch(I input)` du résultat de `registerForActivityResult()`

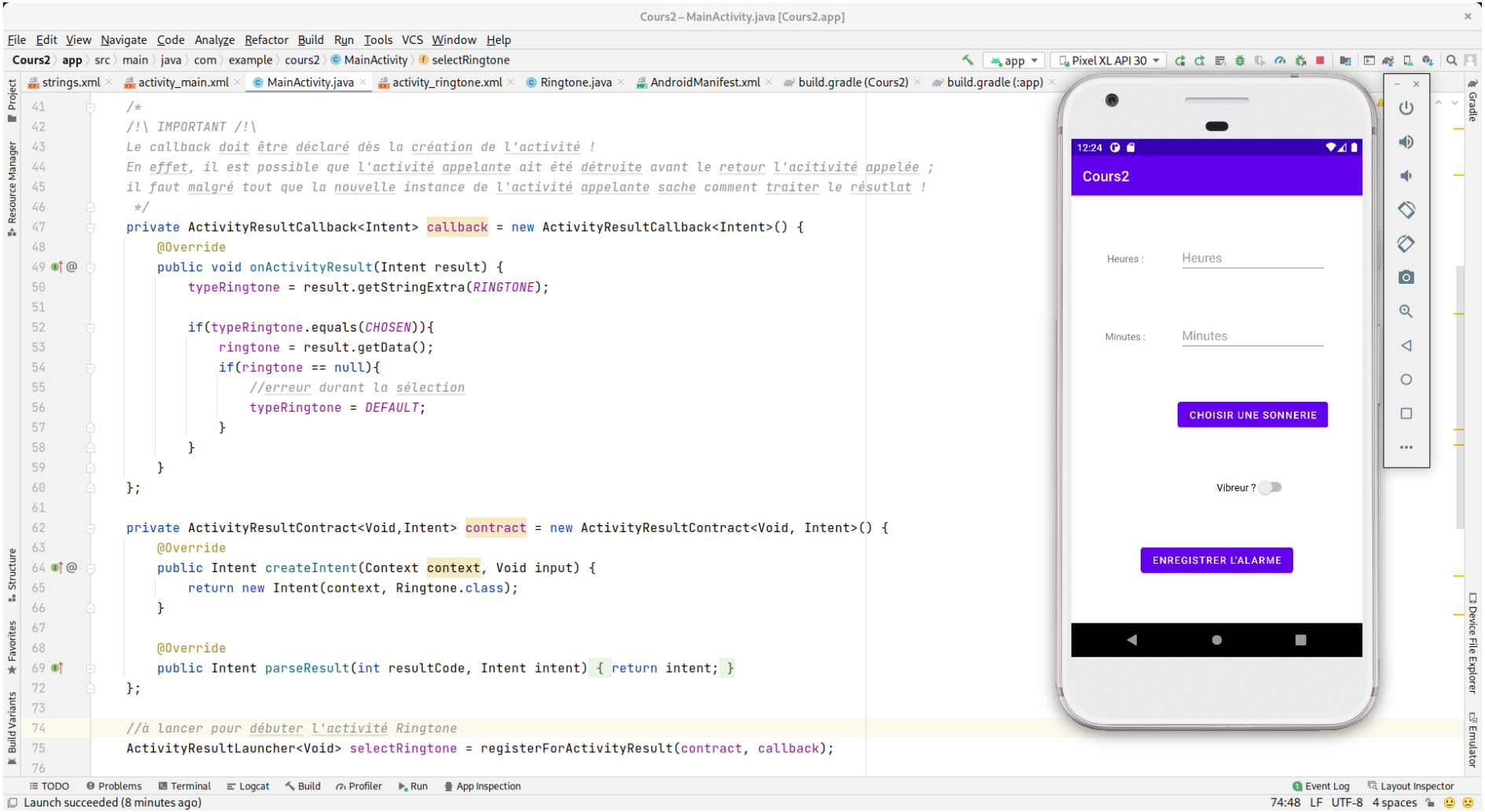
Le callback doit être défini dès la création de l'`Activity` appelante.

registerForActivityResult() : derrière la scène

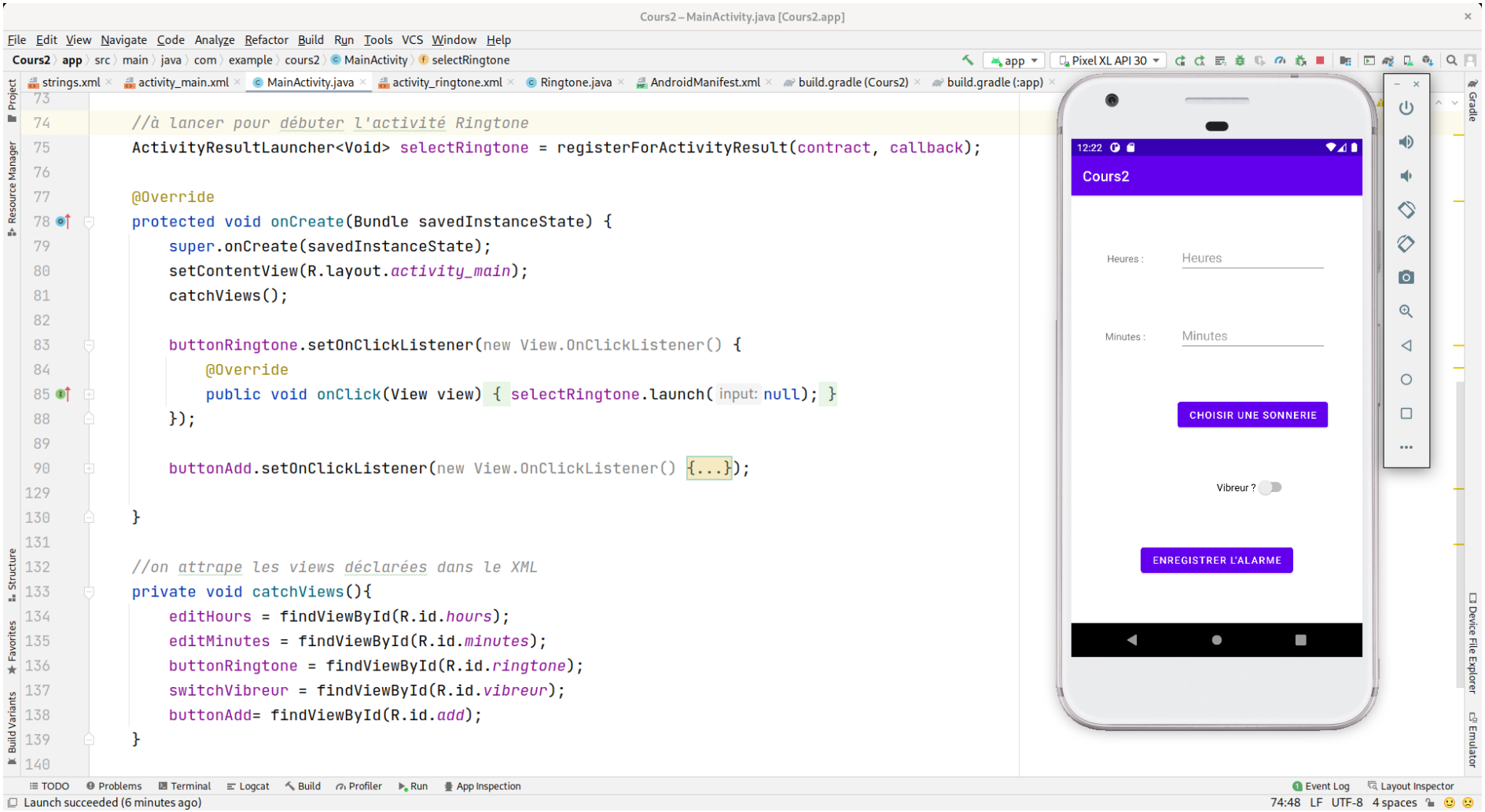


registerForActivityResult()

Ici, `I` = `Void` (aucun input) et `O` = `Intent`.

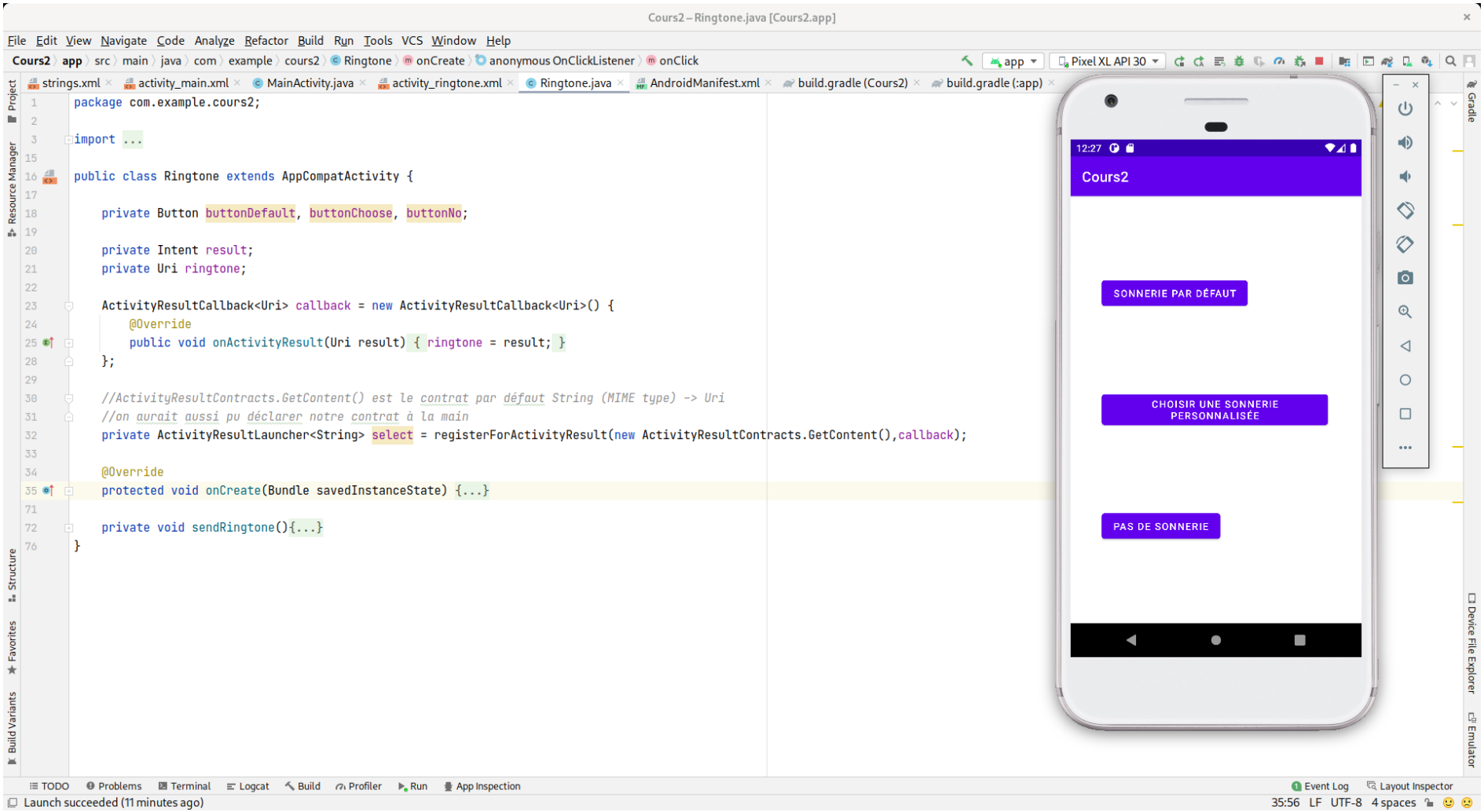


registerForActivityResult()

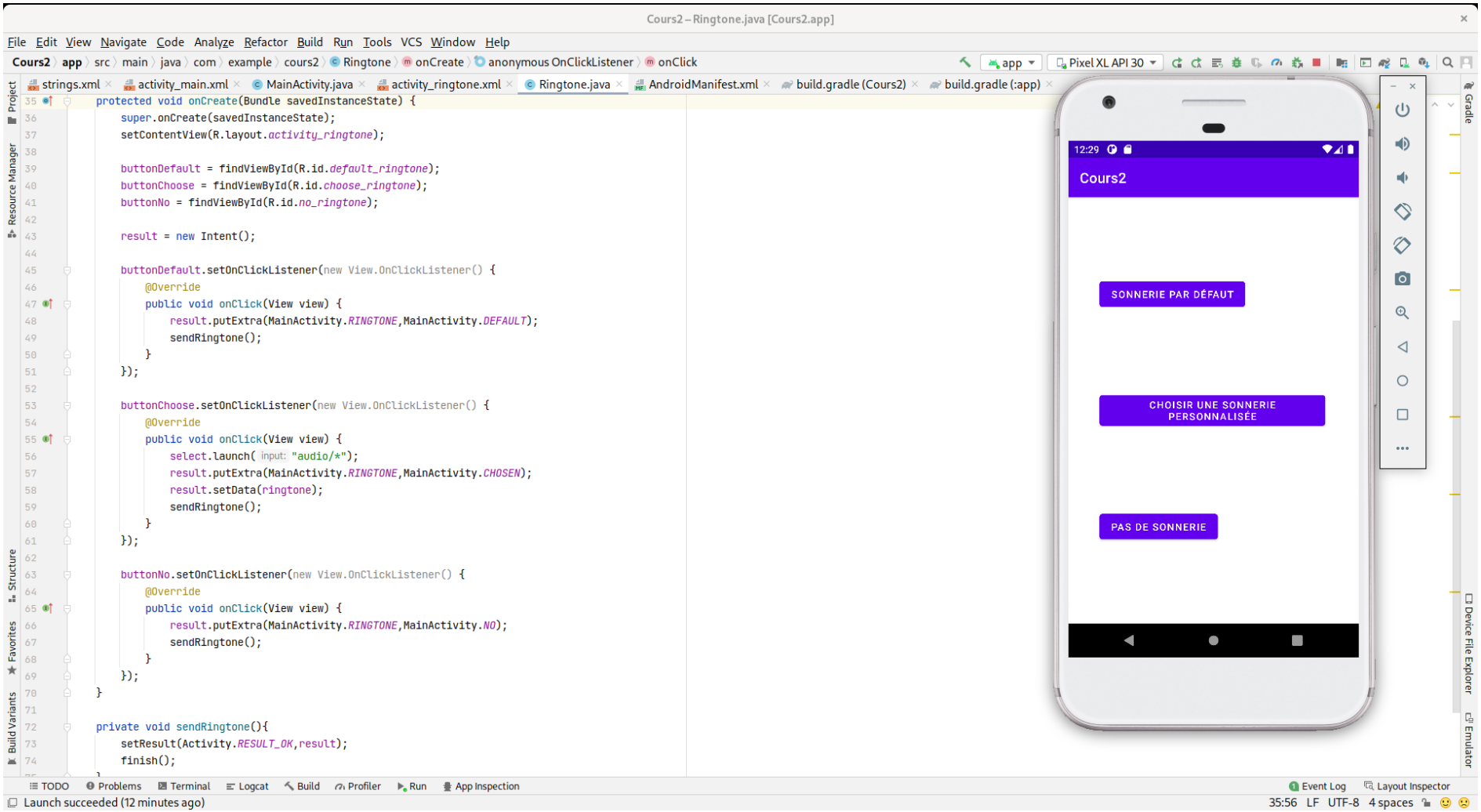


Ringtone

Ici, `I` = `String` (MIME type) et `O` = `Uri` (fichier audio).



Ringtone


Cours2 – Ringtone.java [Cours2.app]
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
strings.xml
activity_main.xml
MainActivity.java
activity_ringtone.xml
Ringtone.java
AndroidManifest.xml
build.gradle (Cours2)
build.gradle (:app)
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_ringtone);
buttonDefault = findViewById(R.id.default_ringtone);
buttonChoose = findViewById(R.id.choose_ringtone);
buttonNo = findViewById(R.id.no_ringtone);
result = new Intent();
buttonDefault.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
result.putExtra(MainActivity.RINGTONE,MainActivity.DEFAULT);
sendRingtone();
}
});
buttonChoose.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
select.launch(input: "audio/*");
result.putExtra(MainActivity.RINGTONE,MainActivity.CHOSEN);
result.setData(ringtone);
sendRingtone();
}
});
buttonNo.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
result.putExtra(MainActivity.RINGTONE,MainActivity.NO);
sendRingtone();
}
});
}
private void sendRingtone(){
setResult(Activity.RESULT_OK,result);
finish();
12:29
Cours2
SONNERIE PAR DÉFAUT
CHOISIR UNE SONNERIE PERSONNALISÉE
PAS DE SONNERIE
TODO Problems Terminal Logcat Build Profiler Run App Inspection
Event Log Layout Inspector
Launch succeeded (12 minutes ago)
35:56 LF UTF-8 4 spaces

setResult()

Pour passer un résultat à l'`Activity` appelante :

```
Intent result = new Intent();
result.putExtra(...);
result.setData(data);

setResult(code, result);
finish();
```

où code est `Activity.RESULT_OK` ou `Activity.RESULT_CANCELED`.

MainActivity Enregistrement de l'alarme

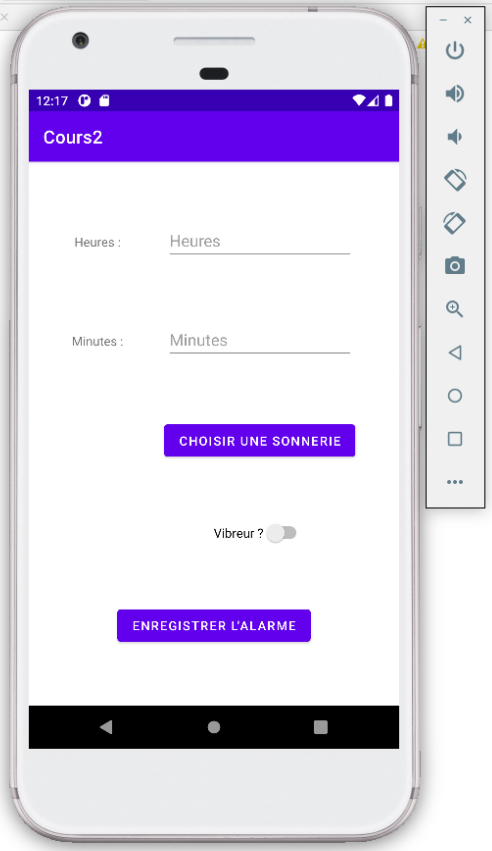
Cours2 – MainActivity.java [Cours2.app]

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Cours2 › app › src › main › java › com › example › cours2 › MainActivity › selectRingtone

strings.xml | activity_main.xml | MainActivity.java | activity_ringtone.xml | Ringtone.java | AndroidManifest.xml | build.gradle (Cours2) | build.gradle (:app)

```
90          buttonAdd.setOnClickListener(new View.OnClickListener() {
91              @Override
92              public void onClick(View view) {
93                  int hours, minutes;
94                  try {
95                      hours = Integer.valueOf(String.valueOf(editHours.getText()));
96                      minutes = Integer.valueOf(String.valueOf(editMinutes.getText()));
97
98                      if(0 <= hours && 0 <= minutes && hours < 24 && minutes < 60){
99                          // l'heure est valide ; on lance l'alarme
100                         Intent setAlarm = new Intent(AlarmClock.ACTION_SET_ALARM);
101                         setAlarm.putExtra(AlarmClock.EXTRA_HOUR,Integer.valueOf(hours));
102                         setAlarm.putExtra(AlarmClock.EXTRA_MINUTES,Integer.valueOf(minutes));
103                         setAlarm.putExtra(AlarmClock.EXTRA_VIBRATE,switchVibreur.isChecked());
104                         switch (typeRingtone){
105                             case DEFAULT:
106                                 //si pas de EXTRA_RINGTONE, la sonnerie par défaut est utilisée
107                                 startActivity(setAlarm);
108                                 break;
109                             case CHOSEN:
110                                 setAlarm.putExtra(AlarmClock.EXTRA_RINGTONE,ringtone);
111                                 startActivity(setAlarm);
112                                 break;
113                             case NO:
114                                 setAlarm.putExtra(AlarmClock.EXTRA_RINGTONE,AlarmClock.VALUE_RINGTONE_SILENT);
115                                 startActivity(setAlarm);
116                                 break;
117                             default:
118                                 ringtoneIssue();
119                         }
120                     } else {
121                         formatIssue();
122                     }
123                 } catch(NumberFormatException e) {
124                     formatIssue();
125                 }
126
127             }
128         });
129
```

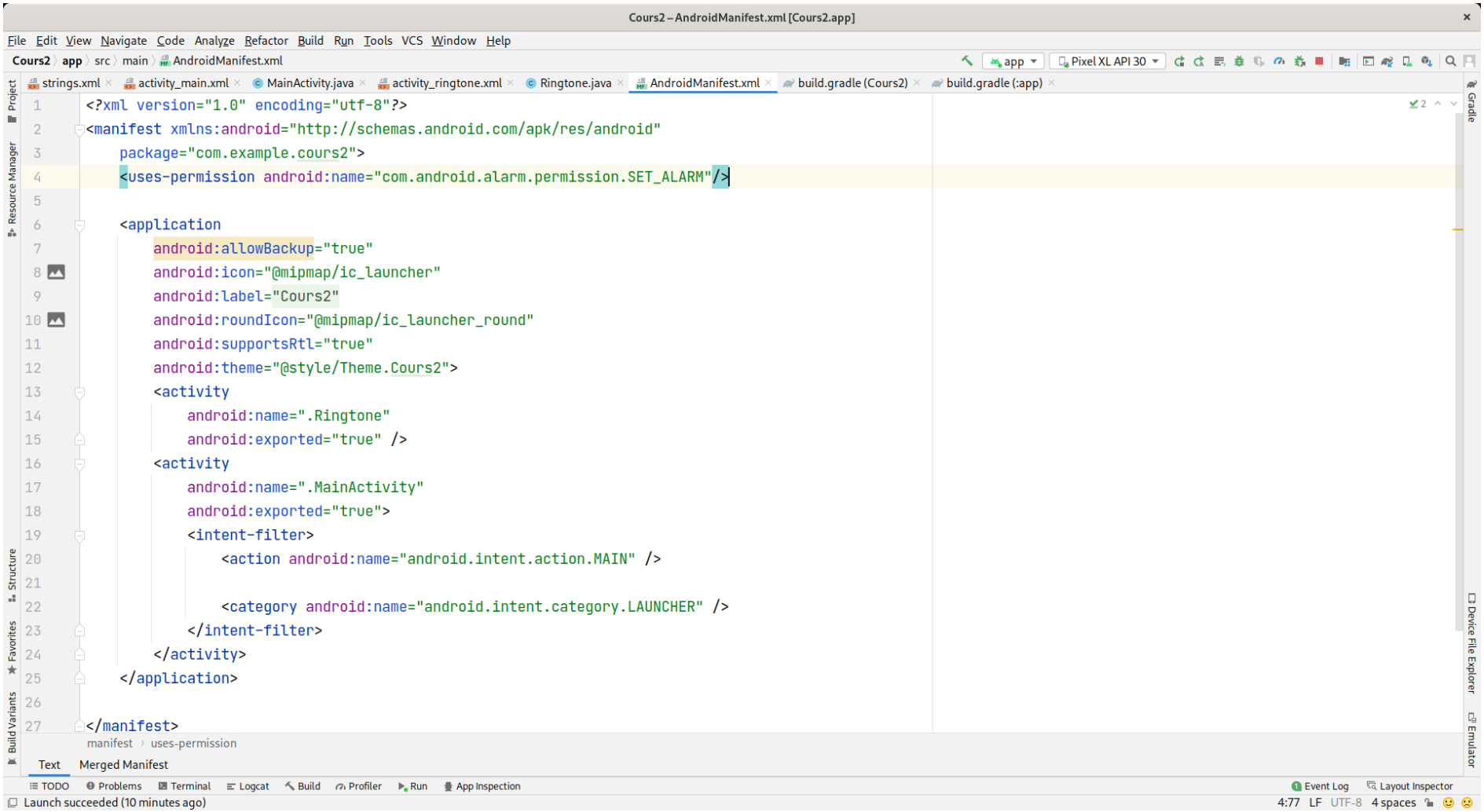


TODO Problems Terminal Logcat Build Profiler Run App Inspection Event Log Layout Inspector

Launch succeeded (moments ago) 74:48 LF UTF-8 4 spaces

Permissions

On demande la permission *“com.android.alarm.permission.SET_ALARM”*.



Signaler qu'on sait traiter un `Intent` implicite

Pour déclarer qu'on sait envoyer des images par mail, ajouter dans le Android Manifest :

```
<activity android:name="MyMailer">
    <intent-filter>
        <action android:name="android.intent.action.SEND"/>
        <category android:name="android.intent.category.DEFAULT"/>
        <data android:mimeType="image/*"/>
    </intent-filter>
</activity>
```