

# TP 1

## Les Bases d'Android

Chamy Kaci  
Ingo Diab

Février 2023



## Table des matières

<b>1</b>	<b>Exercice 2</b>	<b>3</b>
<b>2</b>	<b>Exercice 3</b>	<b>4</b>
<b>3</b>	<b>Exercice 4</b>	<b>6</b>
<b>4</b>	<b>Exercice 5</b>	<b>7</b>
<b>5</b>	<b>Exercice 6</b>	<b>8</b>
<b>6</b>	<b>Exercice 7</b>	<b>10</b>
<b>7</b>	<b>Exercice 8</b>	<b>11</b>
<b>8</b>	<b>Exercice 9</b>	<b>12</b>

## 1 Exercice 2

### hello world

En créant un nouveau projet mobile, on peut afficher un "Hello World!"

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

Code XML



Vue de l'activité Hello World

## 2 Exercice 3

### Une première application- Interface simple

Pour cet exercice, nous avons, en premier lieu, créé une vue exclusivement en XML.

```

<ScrollView xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        tools:context=".MainActivity">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="16dp"
            android:padding="8dp"
            android:text="Nom : " />

        <EditText
            android:id="@+id/nameEdit"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            android:layout_marginStart="16dp"
            android:layout_marginEnd="16dp"
            android:hint="Insérez votre nom" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="16dp"
            android:padding="8dp"
            android:text="Prenom : " />
    </LinearLayout>
</ScrollView>

```

Morceau du Code XML

Vue de l'activité (seulement XML)

Ensuite, nous avons créé une vue exclusivement en Java.

```
// MyPhone.java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    LinearLayout _verticalMain = new LinearLayout( context, this);
    _verticalMain.setOrientation(LinearLayout.VERTICAL);

    //Name
    TextView _nameLabel = new TextView( context, this);
    _nameLabel.setText("Nom :");
    _verticalMain.addView(_nameLabel);

    EditText _nameEdit = new EditText( context, this);
    _nameEdit.setInputType(InputType.TYPE_TEXT_VARIATION_PERSON_NAME);
    _nameEdit.setHint("Insérez votre nom");
    _verticalMain.addView(_nameEdit);
    //

    //Surname
    TextView _surnameLabel = new TextView( context, this);
    _surnameLabel.setText("Prénom :");
    _verticalMain.addView(_surnameLabel);

    EditText _surnameEdit = new EditText( context, this);
    _surnameEdit.setInputType(InputType.TYPE_TEXT_VARIATION_PERSON_NAME);
    _surnameEdit.setHint("Insérez votre Prénom");
    _verticalMain.addView(_surnameEdit);
    //

    //Age
    TextView _ageLabel = new TextView( context, this);
    _ageLabel.setText("Age :");
```

Morceau du Code Java

Vue de l'activité (seulement Java)

### 3 Exercice 4

#### Internationalisation des interfaces

Dans cet exercice, nous avons pris en compte le changement de langue. Le texte n'est pas directement écrit dans le code XML, on utilise des références vers des ressources strings. Lorsque le téléphone est en anglais, ce sera la ressource strings pour la version anglaise qui sera utilisée.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="hello">Hello World, HelloActivity!</string>
  <string name="app_name">My Application</string>

  <string name="nom">Name :</string>
  <string name="prenom">Firstname :</string>
  <string name="age">Age :</string>
  <string name="domaine_de_compétence">Expertise Aera :</string>
```

Morceau de la ressource strings  
pour la version anglaise

Vue de l'activité en anglais

## 4 Exercice 5

### Événements associés aux objets graphiques d'une vue

Dans cet exercice, nous avons associé au clique sur le bouton Valider/Submit, l'ouverture d'une Alert Box. Cette Alert Box contient deux boutons, un pour valider et un pour annuler.

```
protected void AlertBox()
{
    AlertDialog.Builder _alert = new AlertDialog.Builder(context);
    _alert.setTitle(R.string.confirmationUserInfos);
    _alert.setCancellable(true);

    A Shampyo? "
    _alert.setPositiveButton(R.string.validateButton, new DialogInterface.OnClickListener() {
        A Shampyo? "
        @Override
        public void onClick(DialogInterface dialog, int which) {
            // ...
        }
    });

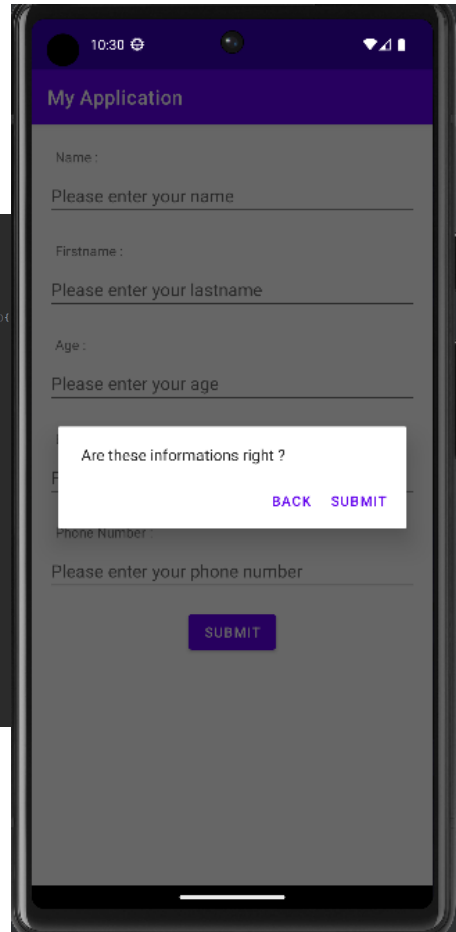
    A Shampyo? "
    _alert.setNegativeButton(R.string.backButton, new DialogInterface.OnClickListener() {
        A Shampyo? "
        @Override
        public void onClick(DialogInterface dialog, int which) {
            // ...
        }
    });

    _alert.show();
}

A Shampyo? "
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    A Shampyo? "
    ((Button)findViewById(R.id.validateButton)).setOnClickListener(new OnClickListener() {
        A Shampyo? "
        @Override
        public void onClick(View view) {
            AlertBox();
        }
    });
}
```

Morceau de code Java

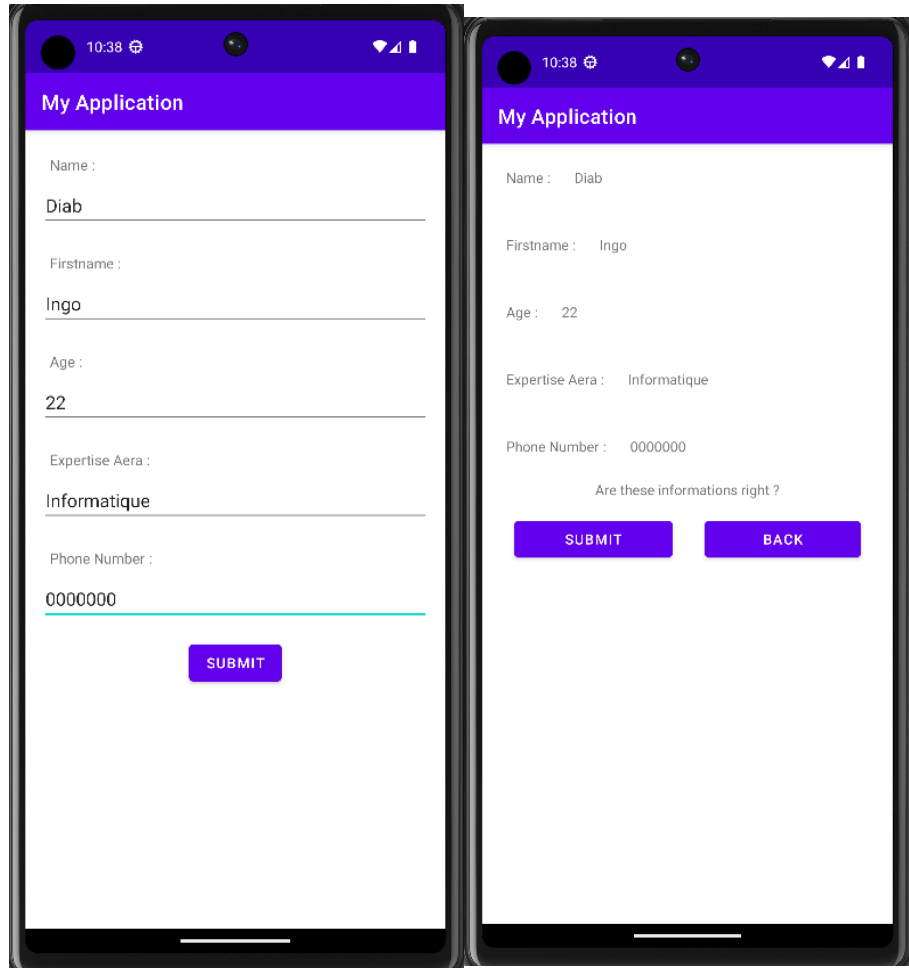


Vue de l'activité avec l'alerte box

## 5 Exercice 6

### Intent explicite

Dans cet exercice, nous avons associé au clique sur le bouton Valider/Submit, le lancement d'une seconde activité. Cette seconde activité récapitule les informations de l'utilisateur et lui demander de valider en cliquant sur un bouton. Cliquer sur ce bouton ouvre une 3e activité qui est un écran vide. Nous pouvons aussi cliquer sur un second bouton pour revenir en arrière.

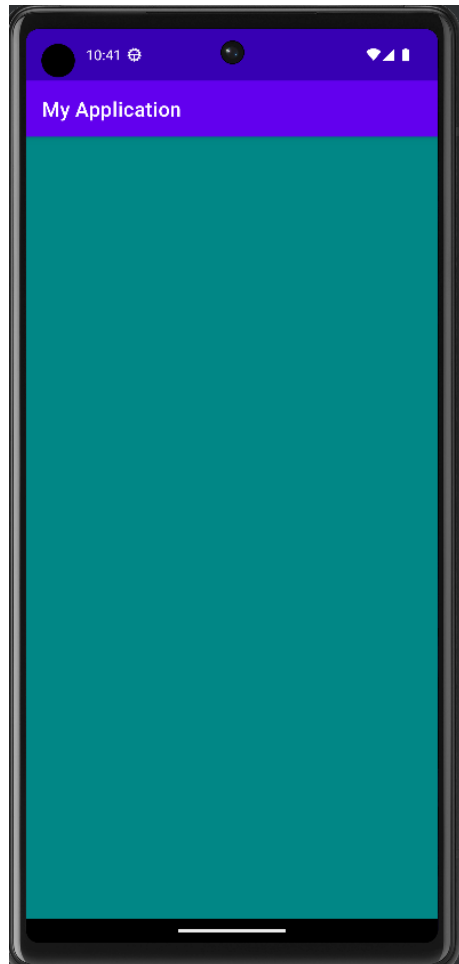


Vue de l'activité principale

Vue de l'activité de confirmation



Cliquer sur Retour/Back nous renvoie à l'activité principale. Si on clique sur Valider/Submit, on arrive sur une 3e activité vide

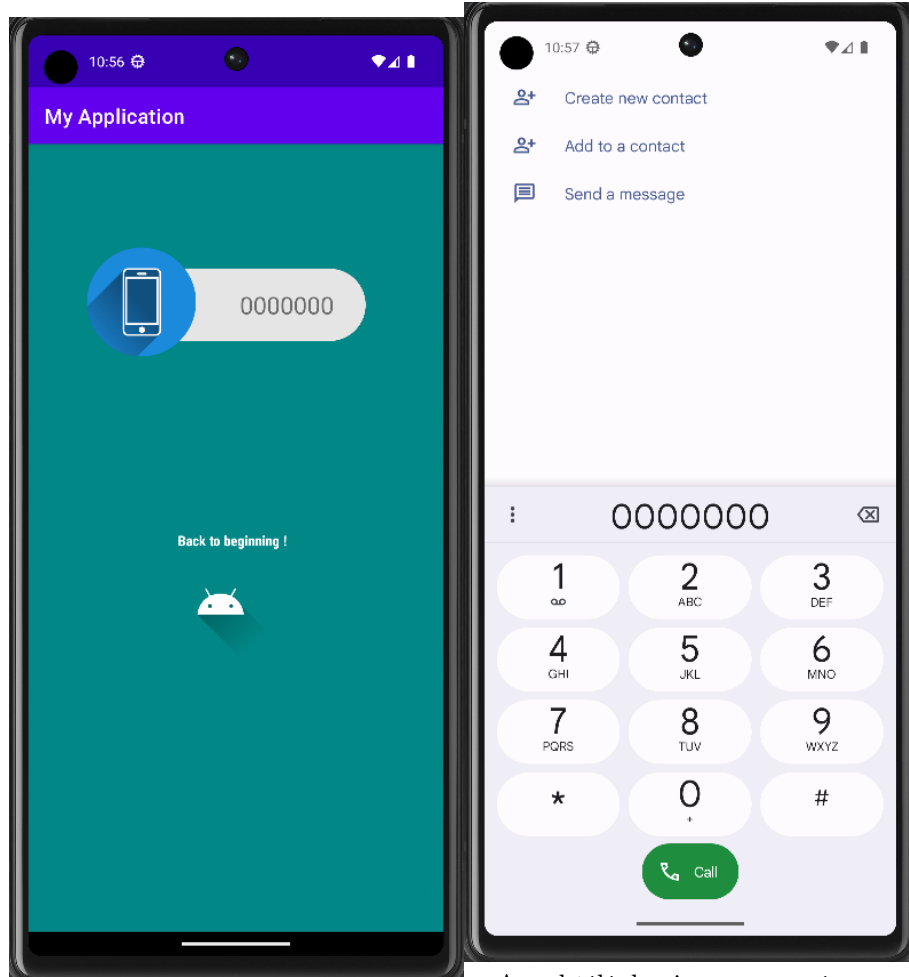


Vue de l'activité vide

## 6 Exercice 7

### Intent implicite

Nous allons remplir la 3e activité précédente qui était vide. Nous lui ajoutons un bouton avec une icône de téléphone. Quand nous cliquons sur ce bouton, on sera redirigé sur un appel téléphonique vers le numéro qui a été fournit par l'utilisateur (ici le numéro de téléphone 0000000). On a aussi un bouton pour revenir à l'activité principale.



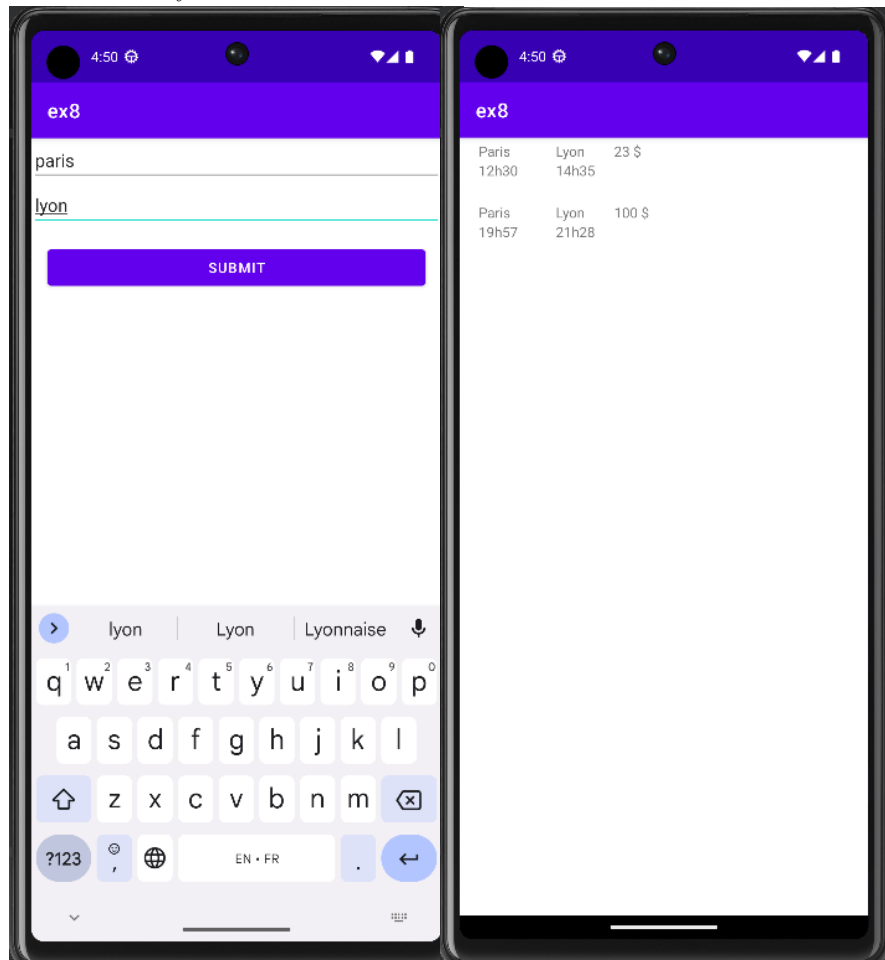
Vue de la 3e activité

Appel téléphonique au numéro  
0000000

## 7 Exercice 8

### Application simple pour consulter les horaires de trains

Dans cet exercice, nous devons créer une activité où l'on peut choisir un trajet. Après validation, l'application crée une nouvelle activité et affiche tous les trajets disponibles, leurs horaires et leurs prix. Pour cela, mon activité contient une liste de Voyage (Trip). Pour cet exemple, nous avons deux voyages disponibles de Paris à Lyon.



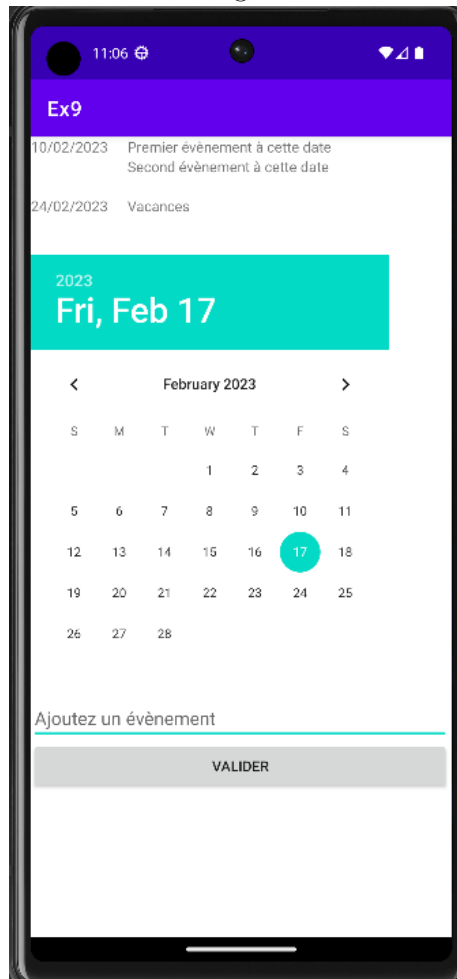
Vue de l'activité (saisie d'un voyage)

Vue de l'activité (tous les horaires disponibles avec leur prix)

## 8 Exercice 9

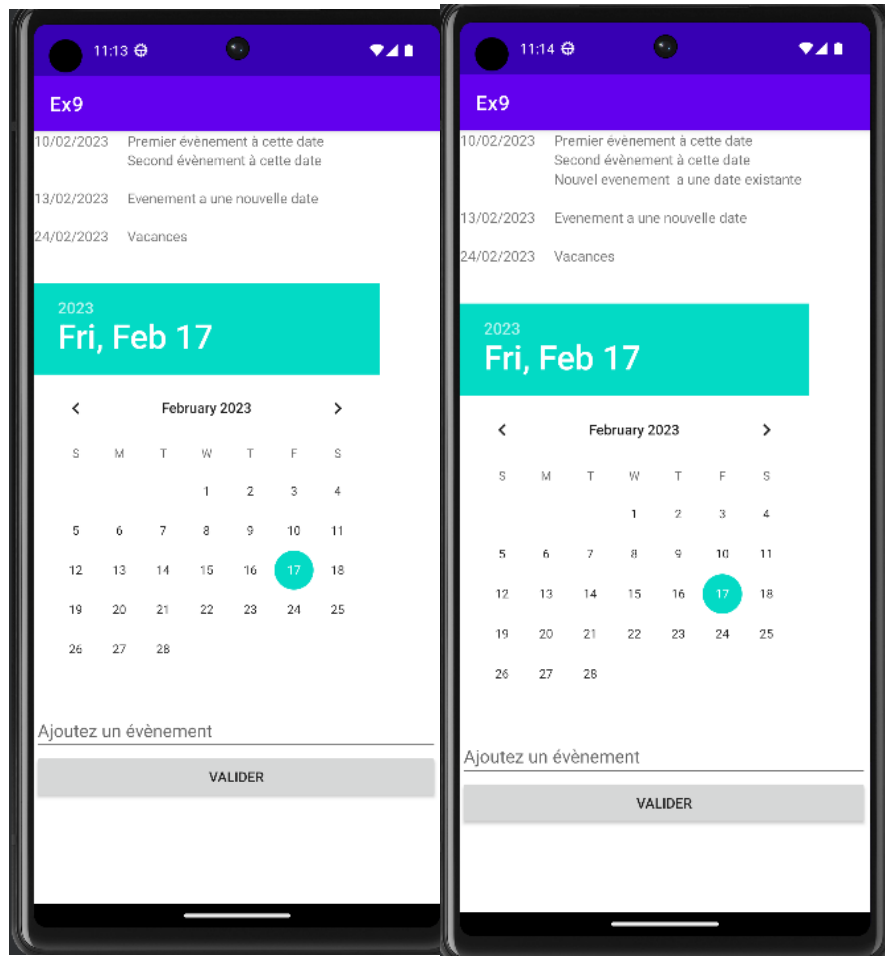
### Application simple d'agenda

Dans cet exercice, nous devons créer un agenda électronique. Pour cela, mon activité contient un dictionnaire qui a un String Date fait correspondre une liste de String qui sont les Evènements prévus. La vue consiste à parcourir ce dictionnaire et afficher chaque Date suivi de sa liste d'Evènement. Enfin j'ai un DatePicker ainsi qu'un EditText et un bouton Valider pour pouvoir ajouter un nouvel évènement à l'agenda.



Vue de l'activité agenda

Je peux ajouter un évènement à une date différente, dans ce cas l'évènement sera ajouter à l'agenda dans une nouvelle entrée et l'agenda sera rechargé. Je peux aussi ajouter un évènement à une date existante, dans ce cas l'évènement se rajoutera à la suite des autres prévus à cette date et l'agenda sera aussi rechargé.



Vue de l'activité agenda (rajout d'évènement à une nouvelle date)

Vue de l'activité agenda (rajout d'évènement à une date existante)