

TP_B2 : NOUVELLE ACTIVITY

Objectif du TP :

- Création d'une nouvelle activity sous Android
- Lien avec nos classes existantes codées en JAVA

CE QU'IL FAUT SAVOIR ET COMPRENDRE :

- Une activité (Activity) est une Vue de notre future application.
- La classe MainActivity hérite de la classe Activity
- Chaque Activity est indépendante
- Toute activité possède au moins une méthode onCreate

CRÉATION DE NOTRE NOUVELLE ACTIVITY



Dans le même package que MainActivity, créer en une nouvelle « blank Activity » nommée ActivityParticipant.

☛ par défaut, cette activity possède déjà un TextView Hello World que l'on ne va pas garder !



Créer dans cette activité, les **différentes zones** nécessaires pour la création d'un participant



Utiliser les textFields à votre disposition



Les mettre dans le bon layout approprié (cf TP initiation Android)



Respecter l'architecture MVC (cf TP initiation Android)

Rappel : `public class` Participant {

```
private String nom;  
private String prenom;  
private Date dateNais;  
private byte numDepartement;
```

exemple :

```
<EditText  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:inputType="date"  
    android:ems="10"  
    android:id="@+id/editTextDateNaisParticipant"  
    android:hint="@string/editTextDateNaisParticipant" />
```

Conseils :

- Utiliser le mode Design pour placer vos différents contrôles graphiques
- Utiliser le mode Text pour paramétrer ces contrôles comme leur nom : (Strings.xml)



Lancer votre application sur votre téléphone en faisant Run sur la classe ActivityParticipant

DECLARER NOTRE NOUVELLE ACTIVITY ET LA LANCER

1) Déclaration de l'activité (AndroidManifest.xml)

Toute nouvelle activité, doit être déclarée dans le fichier `AndroidManifest.xml` pour être prise en compte.

Changer le label afin de nommer cette activité : **Gestion d'un participant**

2) Lancer cette activité

Par défaut, c'est l'activity `MainActivity` qui se lance en premier. Comme chaque activity est indépendante, il faut bien préciser laquelle va être lancée.

Ce code permet de spécifier quelle activité est principale(Main) et quelle activité va se lancer en premier !

```
<intent-filter>
    <action android:name="android.intent.action.MAIN" />

    <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
```



Mettre ce code sur notre nouvelle activité.



Lancer votre application sur votre AVD en faisant Run sur la classe `ActivityParticipant`.

LIEN AVEC LA CLASSE PARTICIPANT

Le but est de récupérer toutes les informations saisies afin de créer un nouveau participant. Pour cela, il faut un bouton pour valider les informations et déclencher le traitement.

1) Créer un nouveau bouton sur l'IHM

Créer un nouveau bouton et nommer le
Enregistrer

Vous devez avoir une IHM comme ci-contre :

2) Gestion du bouton dans le fichier ActivityParticipant.JAVA

Nous allons récupérer le bouton et voir s'il est cliqué.

Si oui, on va créer un nouveau participant avec les informations saisies.

Puis on va afficher ses informations dans un Toast



Rappel : Tout objet GRAPHIQUE possède un id que l'on va pouvoir utiliser pour l'identifier à coup sûr ! (même principe qu'en Javascript ;)



TRAVAIL à faire : dans le fichier ActivityParticipant.java

Dans la méthode onCreate() : lors de la création de l'activité

- Récupérer le bouton par son id dans la vue courante :
Button btnEnregistrer=(Button) findViewById(R.id.buttonEnregistrer);
- On paramètre le bouton pour lui dire qu'il va être « écouter »! principe des écoutes téléphoniques !
btnEnregistrer.setOnClickListener(btnclick); // btnclick à garder

EN DESSOUS de la méthode onCreate() :

- On va « écouter » si ce bouton est cliqué ! Création de l'objet btnclick qui va réellement être dans l'écoute du click. Cette Classe OnClickListener est une classe **Abstraite** qui contient une **méthode onclick Abstraite** qu'il faut Obligatoirement **redéfinir** !
Particularité : on le fait ici !

```
private View.OnClickListener btnclick = new View.OnClickListener() {

    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.buttonEnregistrer:
                /*le clic sur le bouton a eu lieu, qu'est-ce que l'on fait*/
                Toast.makeText(getApplicationContext(), "Enregistrement du participant
en cours", Toast.LENGTH_SHORT).show();
                try {
                    //création d'un nouveau participant en récupérant les id des différents champs
                    //aide : new SimpleDateFormat("dd/MM/yyyy").parse("date à récupérer "),
                    //pour récupérer un élément par son id on utilise la méthode findViewById :
                    ((EditText)findViewById(R.id.editTextNomParticipant)).getText().toString()

                    //affichage de votre nouveau participant dans un Toast pour vérifier de sa bonne
                    création

                }catch(ParseException pe){
                    pe.printStackTrace();
                    //pour debugger plus visuellement, juste pour cette fois-ci
                    Toast.makeText(getApplicationContext(), "ERREUR du PARSE date",
Toast.LENGTH_SHORT).show();
                }
                break;
            }
        }
    };
};
```