

PROGRAMMATION DE COMPOSANTS MOBILES (ANDROID)

WIESLAW ZIELONKA

WWW.IRIF.UNIV-PARIS-DIDEROT.FR/~ZIELONKA

view binding

Le code de l'activité commence d'habitude par la récupération de références vers les views qui constituent l'activité.

Cette étape peut être automatisée.

Le code de l'activité qui n'utilise pas view binding :

```
class MainActivity : AppCompatActivity() {

    val text by lazy { findViewById(R.id.text) as TextView }
    val spinner by lazy { findViewById(R.id.spinner) as Spinner }
    val checkedTextView by lazy { findViewById(R.id.checked) as CheckedTextView }
    val group by lazy { findViewById(R.id.group) as RadioGroup }
    val gauche by lazy { findViewById(R.id.gauche) as RadioButton }
    val droit by lazy { findViewById(R.id.droit) as RadioButton }
    val up by lazy { findViewById(R.id.up) as CheckBox }
    val down by lazy { findViewById(R.id.down) as CheckBox }
    val image by lazy { findViewById(R.id.image) as ImageView }
    val switch by lazy { findViewById(R.id.myswitch) as Switch }
    val toggle by lazy { findViewById(R.id.toggleButton) as ToggleButton }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        spinner.setOnItemClickListener( ... )
        group.setOnCheckedChangeListener { ... }
        toggle.setOnCheckedChangeListener{ ... }
        switch.setOnCheckedChangeListener { ... }
    }

    // callback de bouton
    fun updown(v: View) {
        when(true){
            up.isChecked && down.isChecked -> image.setImageResource(R.drawable.updown)
            up.isChecked -> image.setImageResource(R.drawable.up)
            down.isChecked -> image.setImageResource(R.drawable.down)
            else -> image.setImageDrawable(null)
        }
    }
}
```


view binding

Ajouter dans `build.gradle(Module ...)` dans la section `android` :

```
android{  
    .....  
    buildFeatures{  
        viewBinding true //enable view binding  
    }  
}
```


view binding

Le view binding génère à partir de fichier layout de l'activité une classe qui contient toutes les références vers tous les views à qui le layout attribue un identifiant avec `android:id`. Le nom de la classe générée par le view binding dépend de nom de fichier layout.

Supposant par exemple que le fichier layout porte le nom **activity_main.xml**

Alors la classe générée par le view binding porte le nom :



ActivityMainBinding

On écrit le nom du fichier en notation "camel case" et on ajoute le mot Binding.

view binding

```
class MainActivity : AppCompatActivity() {  
  
    private lateinit var binding : ActivityMainBinding  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        // setContentView(R.layout.activity_main)  
  
        binding = ActivityMainBinding.inflate( layoutInflater )  
  
        setContentView( binding.root )  
    }  
}
```

- On utilise la classe `....Binding` pour créer un objet **binding** de cette classe à l'aide de la méthode `inflate()`.
- On passe **binding.root** dans **setContentView()**. `binding.root` est la référence vers le layout à la racine.
- A partir de ce moment une référence vers une view nommé, par exemple, **toggle** s'obtient avec **binding.toggle**
- quand vous déclarez binding acceptez import proposé par AndroidStudio

view binding

```
binding.spinner.setOnItemSelectedListener( ... )
binding.group.setOnCheckedChangeListener { .... }
binding.toggleButton.setOnCheckedChangeListener { ... }
binding.myswitch.setOnCheckedChangeListener { ... }
```

```
}//fin onCreate()
```

```
//callback de bouton
```

```
fun updown(v: View) {
    when(true){
        binding.up.isChecked && binding.down.isChecked ->
            binding.image.setImageResource(R.drawable.updown)
        binding.up.isChecked -> binding.image.setImageResource(R.drawable.up)
        binding.down.isChecked -> binding.image.setImageResource(R.drawable.down)
        else -> binding.image.setImageDrawable(null)
    }
}
```


view binding

La documentation d'android consacrée aux view binding explique comment procéder si l'activité utilise le fichier layout différent pour les positions verticale et horizontale avec les views différents dans les deux cas.