

# LẬP TRÌNH ANDROID CĂN BẢN

## Bài 9: View Binding & Data Binding

Ths. Trần Xuân Thanh Phúc | Trường Đại học Công Nghiệp Thực Phẩm

# Nội dung

- View Binding
- Data Binding

# 1. View Binding

- Cách truyền thống

```
setContentView(R.layout.activity_main);  
txtViewMainTitle = findViewById(R.id.mainTitle);  
txtViewSubTitle = findViewById(R.id.subTitle);
```

- Lặp đi lặp lại, thời gian phát triển kéo dài
- Dễ bị lỗi NullPointerException, ClassCast
- Lập trình viên vừa phải nhớ tên biến, vừa phải nhớ ID resource

# 1. View Binding

- Gradle App
- Activity Class

```
android {  
    ...  
    buildFeatures {  
        viewBinding true  
    }  
}
```

```
private <LayoutName>Binding binding;  
  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    binding = <LayoutName>Binding.inflate(getLayoutInflater());  
    setContentView(binding.getRoot());  
    binding.<ID_Resource>....  
        // ID_Resource & LayoutName in camelCase form  
}
```

# 1. View Binding

- Fragment

```
private <LayoutName>Binding binding;

@Override
public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    binding = <LayoutName>Binding.inflate(inflater, container, false);
    binding.<ID_Resource>....
    return binding.getRoot();
}
```

# 1. View Binding

- Recycler Adapter

```
@NonNull
@Override
public DemoViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {
    <LayoutName>Binding binding =
    <LayoutName>Binding.inflate(LayoutInflater.from(parent.getContext()), parent,
false);
    return new DemoViewHolder(binding);
}

public class DemoViewHolder extends RecyclerView.ViewHolder {

    <LayoutName>Binding binding;

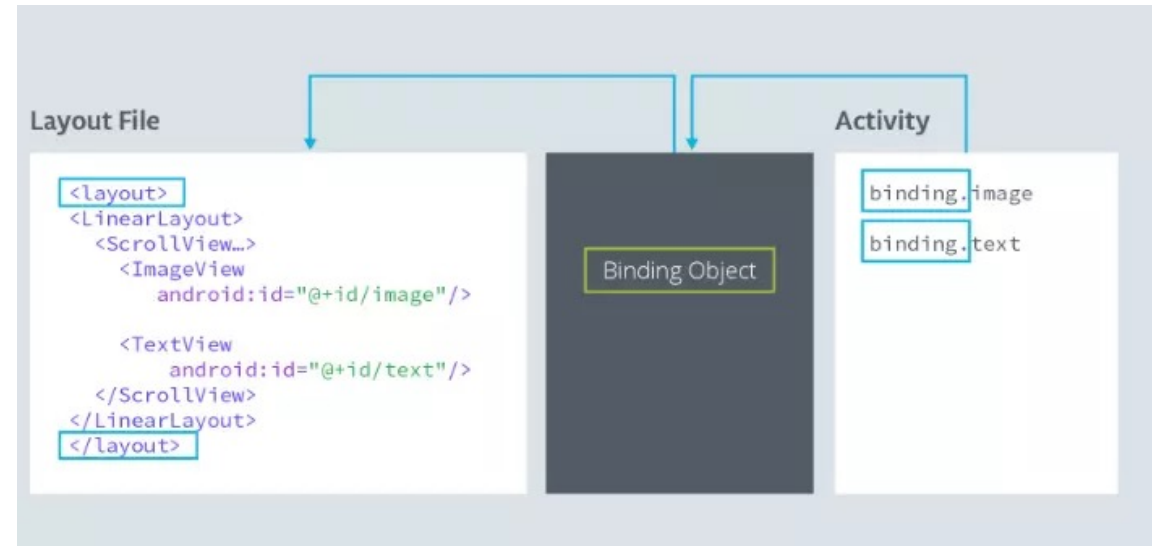
    public DemoViewHolder(@NonNull <LayoutName>Binding itemRowBinding) {
        super(itemRowBinding.getRoot());
        binding = itemRowBinding;
    }
}
```

# 1. View Binding

- Ưu điểm
  - Null Safety
  - Type Safety
  - Speed

## 2. Data Binding

- Data Binding là một thư viện được tích hợp trong gói Android Jetpack. Nó cho phép liên kết giữa dữ liệu logic với các UI Element
- Khác với việc sử dụng findViewById:
  - Data binding tạo ra đối tượng ràng buộc tại thời điểm compile (thay vì Runtime với findViewById) cho tất cả các view trong layout.
  - Các view được truy cập thông qua đối tượng Binding sẽ hiệu quả hơn so với việc sử dụng findViewById.





## 2. Data Binding

- Gradle App

```
android {  
    ...  
    buildFeatures {  
        dataBinding true  
    }  
}
```

## 2. Data Binding

- Layout

```
<?xml version="1.0" encoding="utf-8"?>
<layout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <LinearLayout>...</ LinearLayout>
</layout>
```

- Activity Class

```
private <LayoutName>Binding binding;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    binding = <LayoutName>Binding.inflate(getLayoutInflater());
    setContentView(binding.getRoot());
}
```

## 2. Data Binding

- View Model Class

```
public class MainViewModel {  
    private String name;  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public void buttonClick(View view) {  
        Log.e("AAA", "click roi");  
    }  
}
```

## 2. Data Binding

- Layout

```
<?xml version="1.0" encoding="utf-8"?>
<layout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <data>
        <variable
            name="m"
            type="com.example.hellobinding.MainViewModel" />
        </data>
    ....
    <TextView
        android:text="@={m.name}"
        .../>
</layout>
```

- Activity class

```
binding.setM(m);
m.setName("Day la Data Binding");
```

## 2. Data Binding

- Event Binding

```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Click here"  
    android:onClick="@{m::buttonClick}"  
>
```

- Listener Binding

```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Click here"  
    android:onClick="@{() -> m.buttonClick()}"  
>
```

## 2. Data Binding

- 1-way Binding

```
public class MainViewModel extends BaseObservable {  
  
    private int so = 0;  
  
    @Bindable  
    public int getSo() {  
        return so;  
    }  
  
    public void setSo(int so) {  
        this.so = so;  
        notifyPropertyChanged(BR.so);  
    }  
  
    public void tangDan() {  
        setSo(so + 1);  
    }  
}
```

## 2. Data Binding

- 2-way Binding

```
<EditText  
    android:text="@={`` + m.so}"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" />
```

## 2. Data Binding

- ObservableField

```
public class MainViewModel {  
  
    public ObservableField<Integer> so = new ObservableField<>(0);  
  
    public MainViewModel() {  
        so.addOnPropertyChangedCallback(new  
Observable.OnPropertyChangedCallback() {  
            @Override  
            public void onPropertyChanged(Observable sender, int propertyId) {  
                // Call when value in observable field value changed  
            }  
        });  
    }  
  
    public void tangDan() {  
        so.set(so.get() + 1);  
    }  
}
```



## 2. Data Binding

- BindingAdapter

- Binding Adapter chỉ đơn giản là static method hoặc instance method được sử dụng để làm thế nào xác định một số thuộc tính người dùng định nghĩa các thuộc tính dữ liệu để ràng buộc với thuộc tính của View

```
@BindingAdapter({"textParam"})  
public static void loadText(View view, String text)  
{  
    view.setText(text);  
}
```

```
<TextView  
    app:textParam="@{String.valueOf(m.so)}"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content" />
```

## Doc

- <https://developer.android.com/topic/libraries/view-binding>
- <https://developer.android.com/topic/libraries/data-binding>
- <https://developer.android.com/topic/libraries/data-binding/binding-adapters>

# LẬP TRÌNH ANDROID CĂN BẢN

Kết thúc 🤗

Ths. Trần Xuân Thanh Phúc | Trường Đại học Công Nghiệp Thực Phẩm