



Aplikacje typu single Activity

Rafał Zientara



- Sposób na przedstawianie danych przez XML
- Możliwość „wszczepienia” danych do widoku przez automatycznie generujące się klasy
- Dla pliku np. `activity_name.xml` wygeneruje się klasa `ActivityNameBinding`
- Przy tworzeniu pliku layout korzeniem całego widoku ma być `<layout>`
- W sekcji `<data>` wpisujemy nazwy zmiennych i typ jaki chcemy obsługiwać
- Obsługę powiązań i zdarzeń piszemy w atrybutach w formie `@{wartość_lub_funkcja}`
- W sekcji `@{}` możemy użyć również wyrażeń lambda

Data Binding – konfiguracja Gradle



```
android {  
    ....  
    dataBinding {  
        enabled = true  
    }  
}
```



Data Binding – XML – plik activity_main.xml

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

    <data>

        <variable name="user" type="com.sdaacademy.zientara.rafal.databinding.models.User"/>

    </data>

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@{user.firstName}"/>

        <TextView android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@{user.lastName}"/>

    </LinearLayout>

</layout>
```

Data Binding - Java



```
public class MainActivity extends AppCompatActivity {  
  
    private ActivityMainBinding binding;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        binding = DataBindingUtil.setContentView(this, R.layout.activity_main);  
        binding.setUser(getFakeUser());  
    }  
  
    .  
    .  
    .  
}
```



Data Binding – przechwytywanie zdarzeń

- Można tworzyć dowolne własne zmienne i przypisywać obiekty

```
<variable name="handlers" type="com.sdaacademy.zientara.rafal.databinding.Handler"/>
```

- Akcje mogą zostać łatwo przypisane przez xml do danego obiektu

```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Kliknij!"  
    android:onClick="@{handlers::onCustomClick}"/>
```

- Należy jednak pamiętać o ustawieniu obiektu ponieważ na początku ma wartość null

```
binding.setUser(user);  
binding.setHandlers(new Handler());
```



Data Binding – sekcja <data>

- Można importować klasy i korzystać ze statycznych funkcji

```
<import type="android.graphics.drawable.Drawable"/>  
<import type="android.view.View"/>  
<import type="com.sdaacademy.zientara.rafal.databinding.MyStringUtils"/>
```

- Tworzyć zmienne do wykorzystania w layout

```
<variable name="image" type="Drawable"/>  
<variable name="note" type="String"/>
```



Data Binding - @{}

Dozwolone akcje

- Mathematical + - / * %
- String concatenation +
- Logical && ||
- Binary & | ^
- Unary + - ! ~
- Shift >> >>> <<
- Comparison == > < >= <=
- instanceof
- Grouping ()
- Literals - character, String, numeric, null
- Cast
- Method calls
- Field access
- Array access []
- Ternary operator ?:

Zakazane

- new
- this
- Super



Przykłady

```
android:text="@{String.valueOf(index + 1)}"  
android:visibility="@{age < 13 ? View.GONE : View.VISIBLE}"
```

Wyrażenia lambda

```
android:onClick="@{(theView) -> presenter.onLongClick(theView, task)}"
```

```
public boolean onLongClick(View view, Task task){}
```



Data Binding – obserwacja zmian

```
private static class User extends BaseObservable {
    private String firstName;
    private String lastName;
    @Bindable
    public String getFirstName() {
        return this.firstName;
    }
    @Bindable
    public String getLastName() {
        return this.lastName;
    }
    public void setFirstName(String firstName) {
        this.firstName = firstName;
        notifyPropertyChanged(BR.firstName);
    }
    public void setLastName(String lastName) {
        this.lastName = lastName;
        notifyPropertyChanged(BR.lastName);
    }
}
```



Zadanie



Informacje o piwie

Na górze ekranu stwórz parę przycisków – po jednym dla każdego piwa.

Po kliknięciu przycisku należy na Activity ustawić model piwa z informacjami:

- Nazwa
- Woltarz
- Cena