

The Android Data Binding Library

Christian Schmitz

22. November 2016

- 1 Standard Layouts
- 2 Android Data Binding
- 3 Keep in mind

Setting a text to a view

- get the view with `findViewById(R.id.view_id)`

Setting a text to a view

- get the view with `findViewById(R.id.view_id)`
- cast the view with `(TextView)view`

Setting a text to a view

- get the view with `findViewById(R.id.view_id)`
- cast the view with `(TextView)view`
- set the text with `textView.setText(text)`

Setting a text to a view

After setting up everything else.
So let's start from the beginning ...

Creating a simple layout

```
1  <LinearLayout
2      android:layout_width="match_parent"
3      android:layout_height="match_parent"
4      android:orientation="vertical">
5      <EditText
6          android:id="@+id/text"
7          android:layout_width="match_parent"
8          android:layout_height="wrap_content" />
9      <Button
10         android:id="@+id/button"
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content" />
13  </LinearLayout>
```

Binding a simple layout

```
1      private String text = "...";
```


Binding a simple layout

```
1     private String text = "...";  
2  
3     public void onClick(View v) {  
4         ...  
5     }
```

Binding a simple layout

```
1     private String text = "...";
2
3     public void onClick(View v) {
4         ...
5     }
6
7     protected void onCreate(Bundle savedInstanceState) {
8         super.onCreate(savedInstanceState);
9         setContentView(R.layout.simple_layout);
10        ((TextView) findViewById(R.id.text)).setText(text);
11        findViewById(R.id.button).setOnClickListener(this);
12    }
```

Binding a simple layout

```
1     private String text = "...";
2
3     public void onClick(View v) {
4         ...
5     }
6
7     protected void onCreate(Bundle savedInstanceState) {
8         super.onCreate(savedInstanceState);
9         setContentView(R.layout.simple_layout);
10        ((TextView) findViewById(R.id.text)).setText(text);
11        findViewById(R.id.button).setOnClickListener(this);
12        // additional work to observe or reset text
13    }
```

- ① Standard Layouts
- ② **Android Data Binding**
- ③ Keep in mind

Advantages of Android Data Bindings

- no getting the view with `findViewById(R.id.view_id)`

Advantages of Android Data Bindings

- no getting the view with `findViewById(R.id.view_id)`
- no casting the view with `(TextView)view`

Advantages of Android Data Bindings

- no getting the view with `findViewById(R.id.view_id)`
- no casting the view with `(TextView)view`
- no setting the text with `textView.setText(text)`

Installing Android Data Binding

Enable Android Data Bindings in `app/build.gradle`

```
1  android {  
2      dataBinding {  
3          enabled true  
4      }  
5  }
```


Android Data Binding layout

```
1    <layout>  
2    </layout>
```

Android Data Binding layout

```
1  <layout>
2    <data>
3      <variable name="data" type="com.example.Data"/>
4    </data>
5  </layout>
```

Android Data Binding layout

```
1  <layout>
2    <data>
3      <variable name="data" type="com.example.Data"/>
4    </data>
5
6    <LinearLayout
7      android:layout_width="match_parent"
8      android:layout_height="match_parent"
9      android:orientation="vertical">
10     <EditText
11       android:layout_width="match_parent"
12       android:layout_height="wrap_content" />
13     <Button
14       android:layout_width="match_parent"
15       android:layout_height="wrap_content" />
16   </LinearLayout>
17 </layout>
```

Android Data Binding layout

```
1  <layout>
2    <data>
3      <variable name="data" type="com.example.Data"/>
4    </data>
5
6    <LinearLayout
7      android:layout_width="match_parent"
8      android:layout_height="match_parent"
9      android:orientation="vertical">
10     <EditText
11       android:layout_width="match_parent"
12       android:layout_height="wrap_content"
13       android:text="@{data.text}" />
14     <Button
15       android:layout_width="match_parent"
16       android:layout_height="wrap_content" />
17   </LinearLayout>
18 </layout>
```

Android Data Binding layout

```
1  <layout>
2    <data>
3      <variable name="data" type="com.example.Data"/>
4    </data>
5
6    <LinearLayout
7      android:layout_width="match_parent"
8      android:layout_height="match_parent"
9      android:orientation="vertical">
10      <EditText
11        android:layout_width="match_parent"
12        android:layout_height="wrap_content"
13        android:text="@{data.text}" />
14      <Button
15        android:layout_width="match_parent"
16        android:layout_height="wrap_content"
17        android:onClick="@{data::onClick}" />
18    </LinearLayout>
19  </layout>
```

Android Data Binding layout

```
1  <layout>
2    <data>
3      <variable name="data" type="com.example.Data"/>
4    </data>
5
6    <LinearLayout
7      android:layout_width="match_parent"
8      android:layout_height="match_parent"
9      android:orientation="vertical">
10      <EditText
11        android:layout_width="match_parent"
12        android:layout_height="wrap_content"
13        android:text="@={data.text}" />
14      <Button
15        android:layout_width="match_parent"
16        android:layout_height="wrap_content"
17        android:onClick="@{data::onClick}" />
18    </LinearLayout>
19  </layout>
```

Android Data Binding variable

Create the view model containing all bindable fields

```
1      public class Data {  
2      }
```

Android Data Binding variable

Create the view model containing all bindable fields

```
1      public class Data {  
2          public final ObservableField<CharSequence> text  
3              = new ObservableField<>();  
4      }
```


Android Data Binding variable

Create the view model containing all bindable fields

```
1    public class Data {  
2        public final ObservableField<CharSequence> text  
3            = new ObservableField<>();  
4  
5        public void onClick(View v) {  
6            ...  
7        }  
8    }
```

Doing the actual data binding

```
1    protected void onCreate(Bundle savedInstanceState) {  
2        super.onCreate(savedInstanceState);  
3        BindingLayoutBinding binding = DataBindingUtil  
4            .setContentView(this, R.layout.binding_layout);  
5        binding.setData(data);  
6    }
```

- ① Standard Layouts
- ② Android Data Binding
- ③ Keep in mind

Android Data Binding reference

- `@{...}` one-way data binding

Android Data Binding reference

- `@{...}` one-way data binding
 - read the value and bind it to the view

Android Data Binding reference

- `@{...}` one-way data binding
 - read the value and bind it to the view
 - observe changes to the value and update the view

Android Data Binding reference

- `@{...}` one-way data binding
 - read the value and bind it to the view
 - observe changes to the value and update the view
 - null safety

Android Data Binding reference

- `@{...}` one-way data binding
 - read the value and bind it to the view
 - observe changes to the value and update the view
 - null safety
 - thread safety

Android Data Binding reference

- `@{...}` one-way data binding
 - read the value and bind it to the view
 - observe changes to the value and update the view
 - null safety
 - thread safety
- `@={...}` two-way data binding

Android Data Binding reference

- `@{...}` one-way data binding
 - read the value and bind it to the view
 - observe changes to the value and update the view
 - null safety
 - thread safety
- `@={...}` two-way data binding
 - all the above

Android Data Binding reference

- `@{...}` one-way data binding
 - read the value and bind it to the view
 - observe changes to the value and update the view
 - null safety
 - thread safety
- `@={...}` two-way data binding
 - all the above
 - update the value with changes from UI

Android Data Binding operators

- Most of Java expression language + - || <= ?: etc

Android Data Binding operators

- Most of Java expression language + - || <= ?: etc
- Not `this` super new though

Android Data Binding operators

- Most of Java expression language + - || <= ?: etc
- Not this super new though
- `data.text` to get a property like `data.getText()`

Android Data Binding operators

- Most of Java expression language `+` `-` `||` `<=` `?:` etc
- Not `this` super new though
- `data.text` to get a property like `data.getText()`
- The null coalescing operator `data.text ?? "abc"`

Android Data Binding operators

- Most of Java expression language `+` `-` `||` `<=` `?:` etc
- Not `this` super new though
- `data.text` to get a property like `data.getText()`
- The null coalescing operator `data.text ?? "abc"`
which is equivalent to `data.text != null ? data.text : "abc"`

Android Data Binding operators

- Most of Java expression language + - || <= ?: etc
- Not this super new though
- `data.text` to get a property like `data.getText()`
- The null coalescing operator `data.text ?? "abc"`
which is equivalent to `data.text != null ? data.text : "abc"`
- `data::onClick` to reference a method

Android Data Binding operators

- Most of Java expression language `+` `-` `||` `<=` `?:` etc
- Not this super new though
- `data.text` to get a property like `data.getText()`
- The null coalescing operator `data.text ?? "abc"`
which is equivalent to `data.text != null ? data.text : "abc"`
- `data::onClick` to reference a method
- `collection[key]` to reference an item of a collection

Android Data Binding operators

- Most of Java expression language `+` `-` `||` `<=` `?:` etc
- Not this super new though
- `data.text` to get a property like `data.getText()`
- The null coalescing operator `data.text ?? "abc"`
which is equivalent to `data.text != null ? data.text : "abc"`
- `data::onClick` to reference a method
- `collection[key]` to reference an item of a collection
- `@string/app_name` to reference a resource value