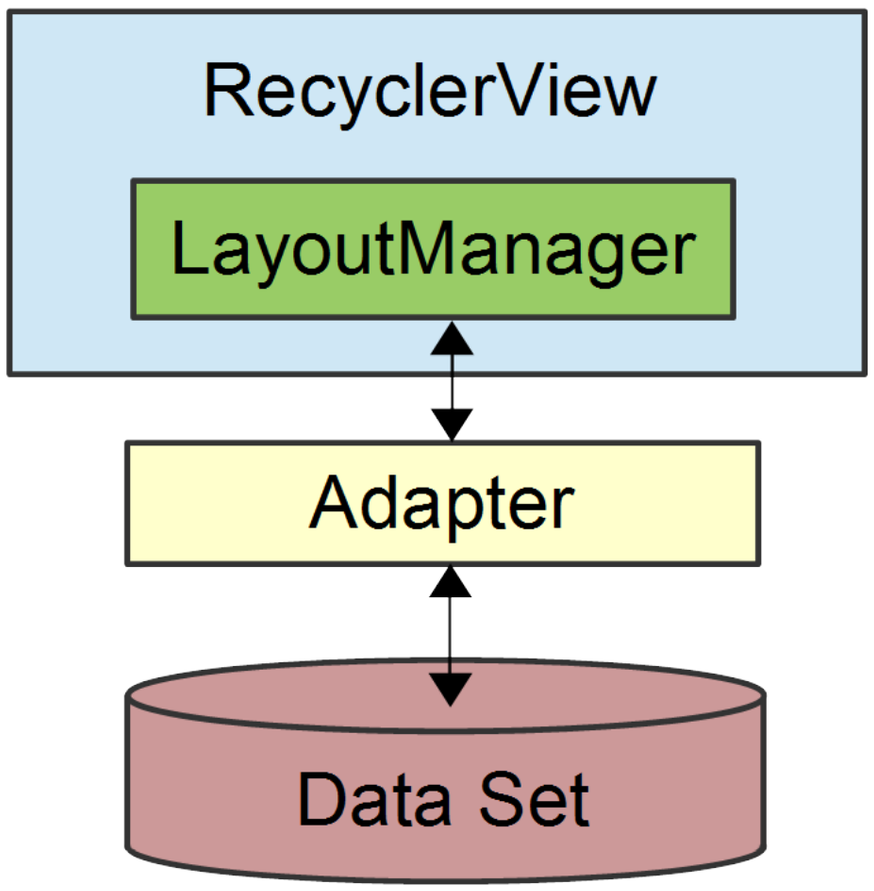
RecyclerView

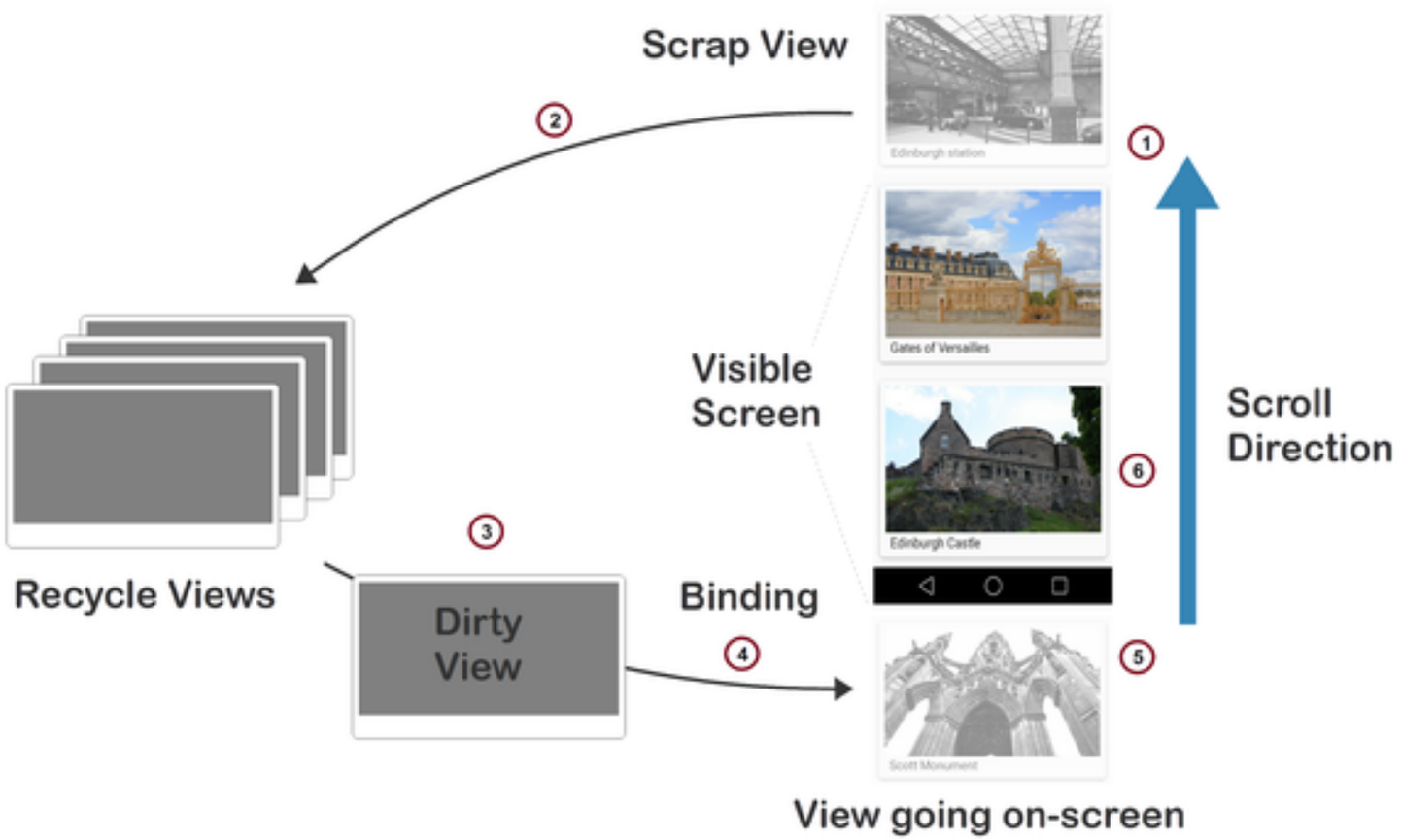
SimpleRecyclerView

Adapter



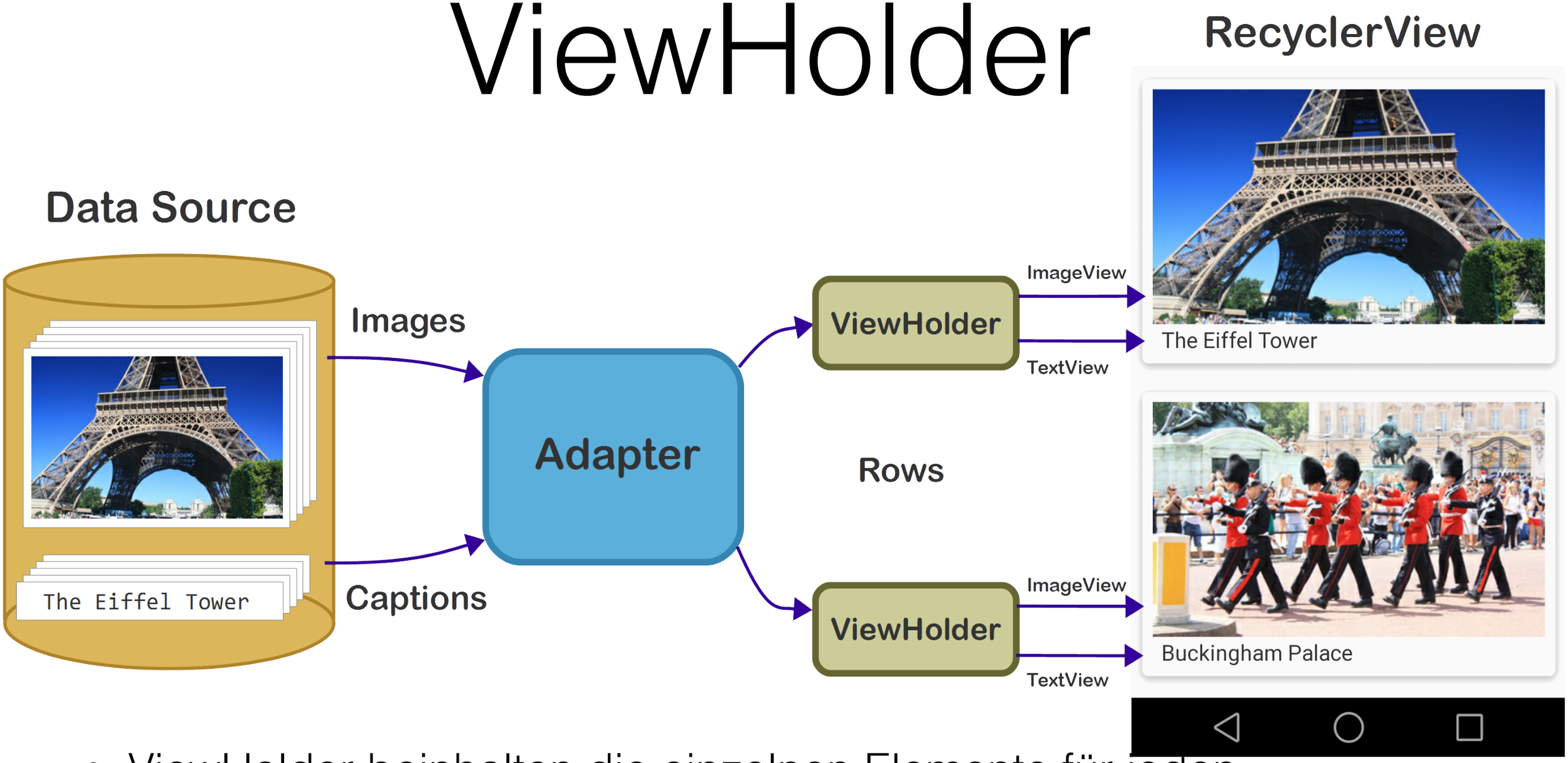
- Ein Adapter ist für die Versorgung der RecylerView mit Daten verantwortlich
- Der LayoutManager für das Aussehen der RecyclerView
- Man kann hier gut das modulare Konzept von Android erkennen

RecyclerView



- Eine RecyclerView „recyclet" Ihre einzelnen Elemente beim Scrollen
- Es werden nicht neue Elemente erzeugt, sondern die nicht mehr sichtbaren Elemente mit neuen Inhalten befüllt und an das sichtbare Ende der Liste angefügt

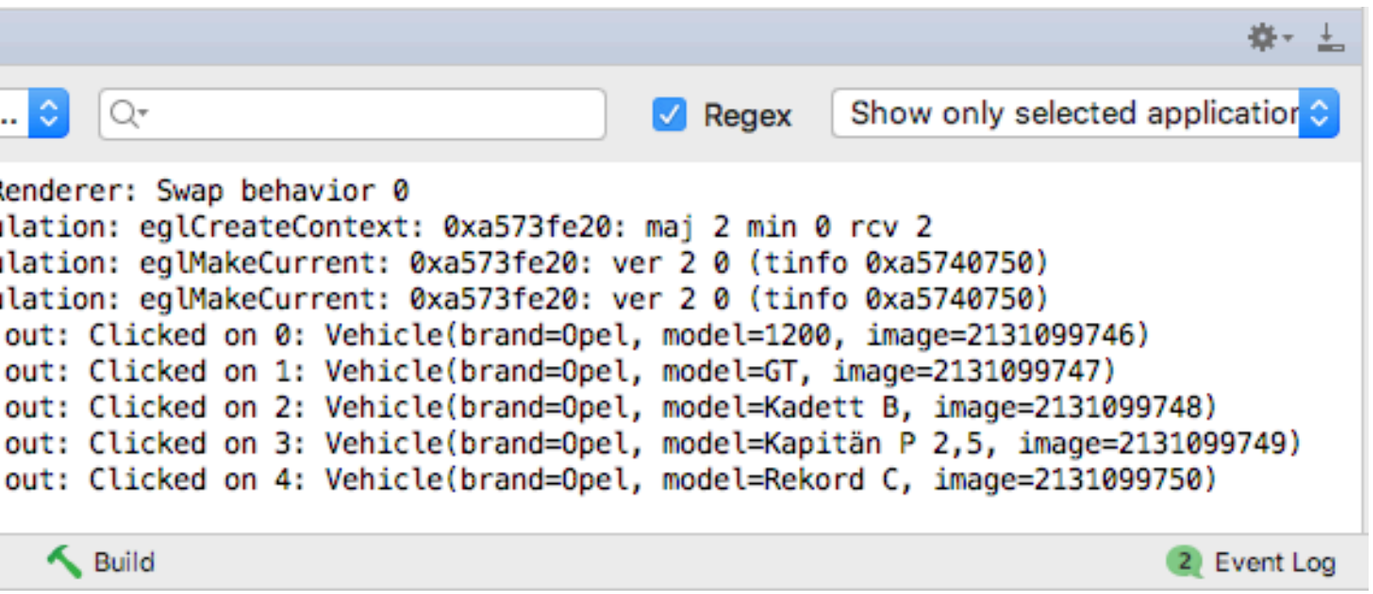
ViewHolder

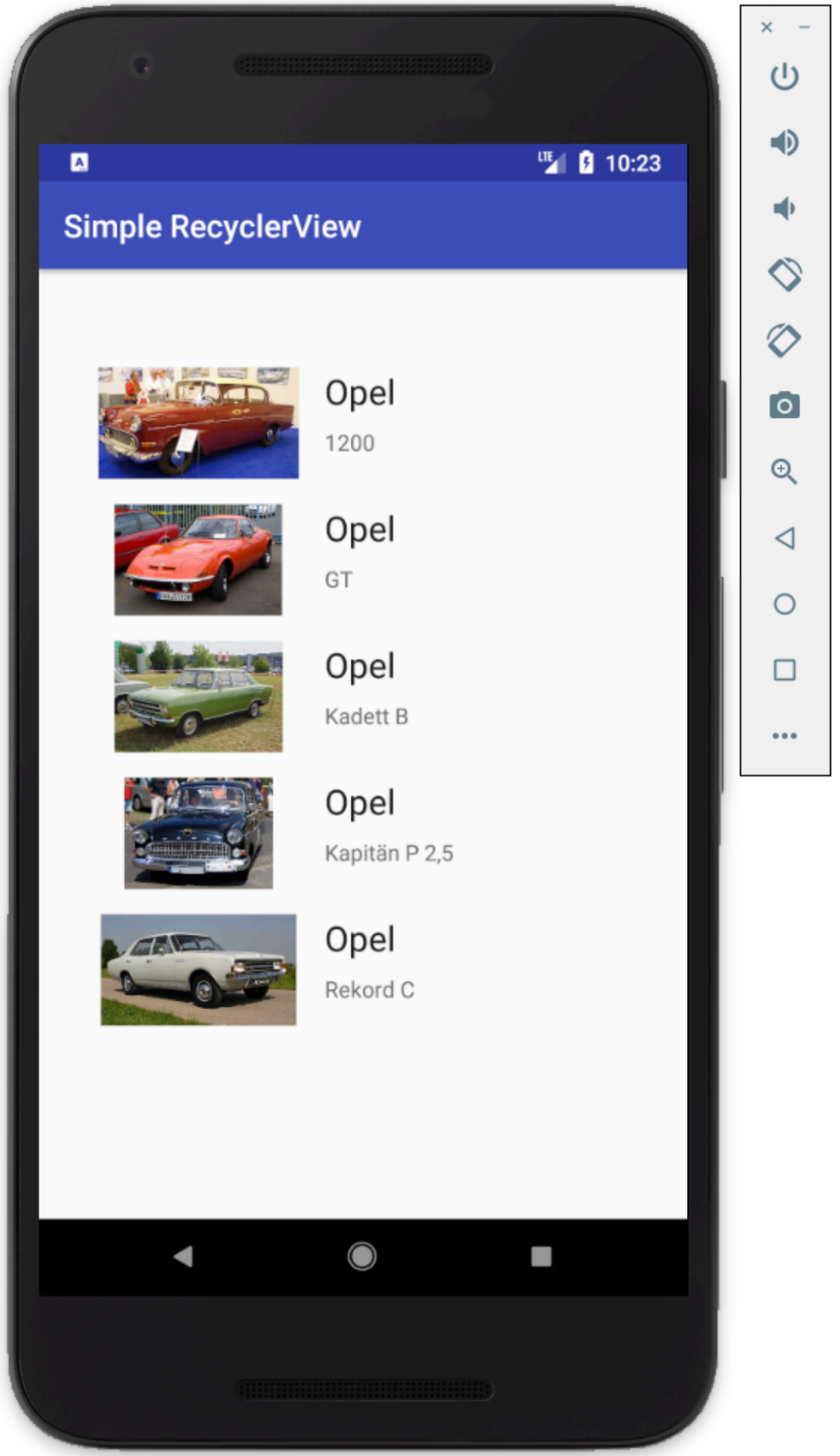


- ViewHolder beinhalten die einzelnen Elemente für jeden Listeneintrag und sind direkt dem Adapter zugeordnet. So muss man nicht (langwierig) über R.id.xxx suchen, sondern kann direkt über das tag-filed des Adapters auf sie zugreifen

Diese App wird erstellt

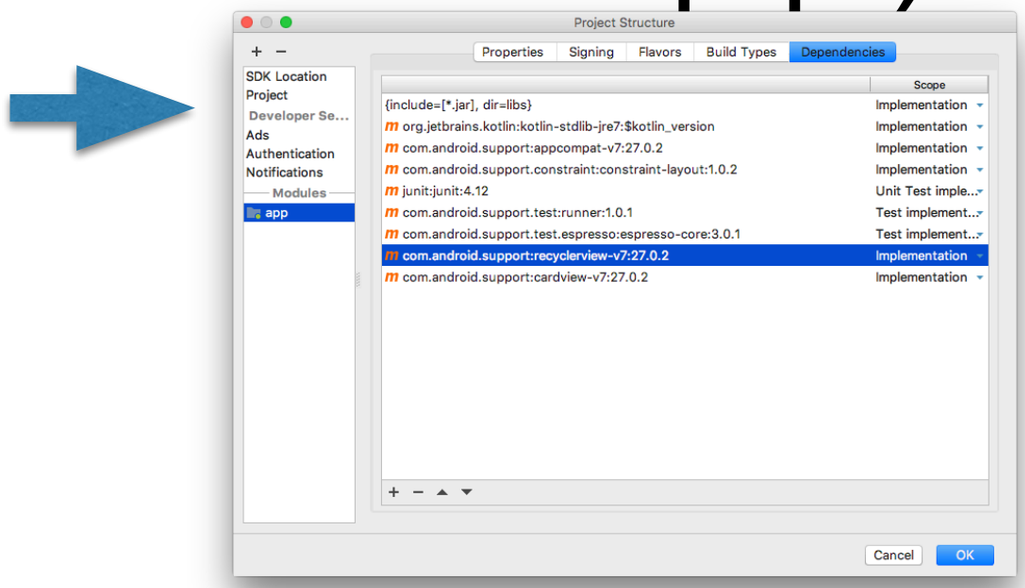
Beim Click auf einen Eintrag wird ein Text ausgegeben





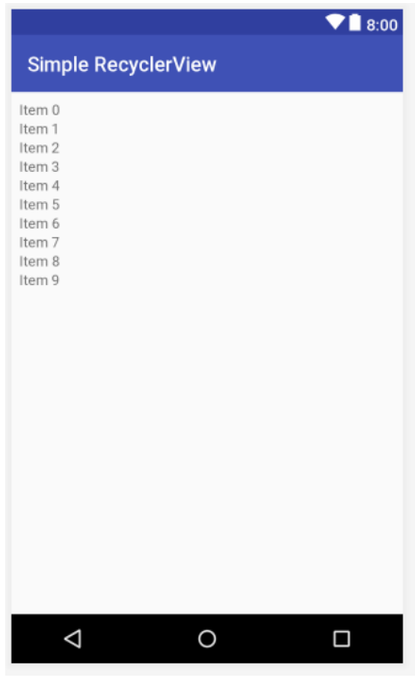
build.gradle (Module: app)

```
apply plugin: 'com.android.application'
apply plugin: 'kotlin-android'
apply plugin: 'kotlin-android-extensions'
```




```
android {
    compileSdkVersion 27
    defaultConfig {
        applicationId "at.htl.simplerecyclerview"
        minSdkVersion 21
        targetSdkVersion 27
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}
dependencies {
    implementation fileTree(include: ['*.jar'], dir: 'libs')
    implementation "org.jetbrains.kotlin:kotlin-stdlib-jre7:$kotlin_version"
    implementation 'com.android.support:appcompat-v7:27.0.2'
    implementation 'com.android.support.constraint:constraint-layout:1.0.2'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.1'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.1'
    implementation 'com.android.support:recyclerview-v7:27.0.2'
    implementation 'com.android.support:cardview-v7:27.0.2'
}
```

activity_main.xml



Erstellen Sie die Layouts mit dem grafischen Designer und kontrollieren Sie anschließend den xml-Code

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <android.support.v7.widget.RecyclerView
        android:id="@+id/rv"
        android:layout_width="368dp"
        android:layout_height="495dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
```

card_vehicle.xml

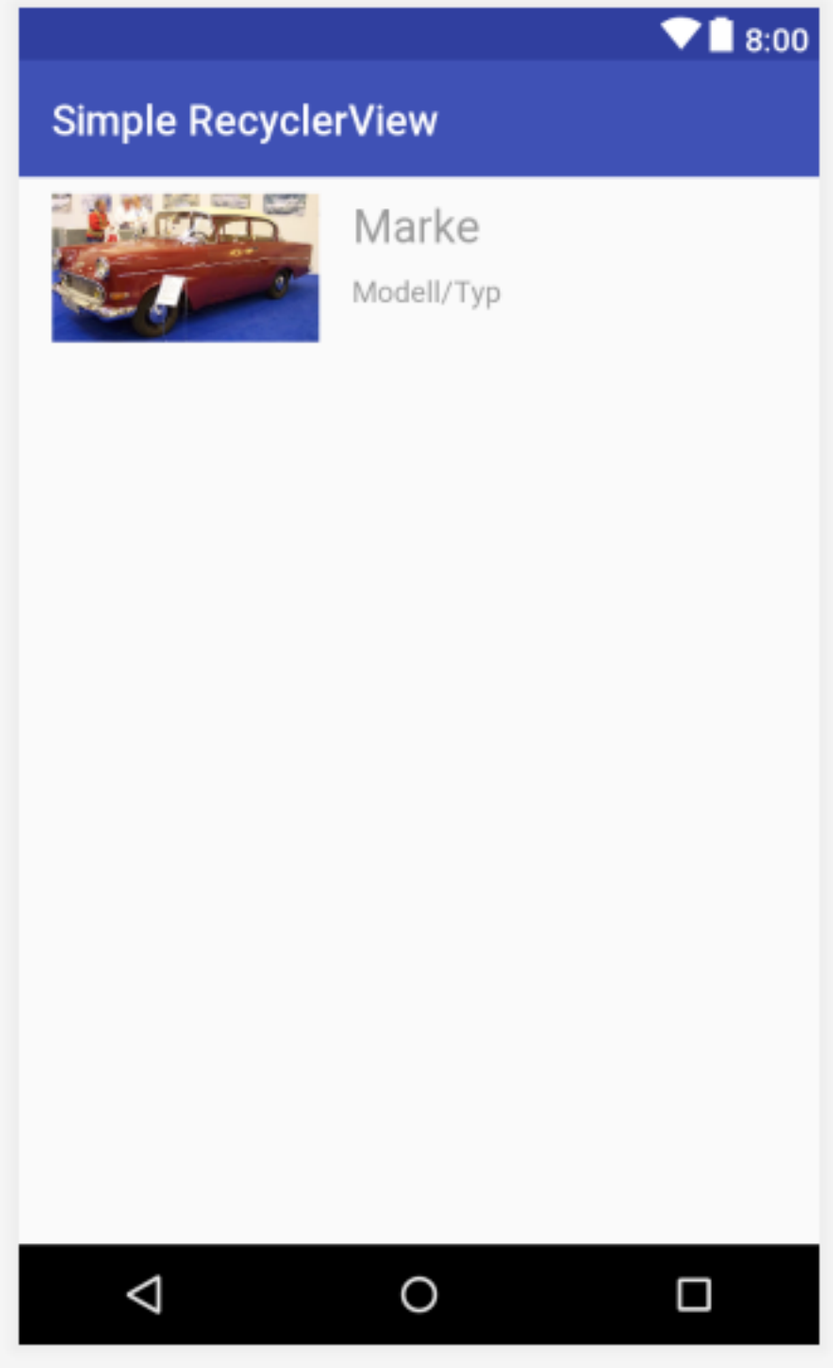
```xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <ImageView
        android:id="@+id/iv_icon"
        android:layout_width="144dp"
        android:layout_height="71dp"
        android:layout_marginBottom="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/opel_1200_1960"
        android:contentDescription="@string/image_of_vehicle" />

    <TextView
        android:id="@+id/tv_brand"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:hint="@string/brand"
        android:textAppearance="@android:style/TextAppearance.Material.Large"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/iv_icon"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/tv_model"
        android:layout_width="0dp"
        android:layout_height="19dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:hint="@string/model"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toEndOf="@+id/iv_icon"
        app:layout_constraintTop_toBottomOf="@+id/tv_brand"
        app:layout_constraintVertical_bias="0.004" />

</android.support.constraint.ConstraintLayout>
```



MainActivity.kt

```
package at.htl.simplerecyclerview

import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.support.v7.widget.LinearLayoutManager
import android.support.v7.widget.RecyclerView
import android.view.View
import android.widget.TextView
import at.htl.simplerecyclerview.model.getSampleVehicles
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {

    init {
        mainActivityContext = this
    }

    companion object {
        lateinit var mainActivityContext: MainActivity
            private set
    }

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

        rv.setHasFixedSize(true)

        rv.layoutManager = LinearLayoutManager(this)
        rv.adapter = VehicleAdapter(getSampleVehicles())

    }
}
```

Hier wird eine statische Konstante deklariert

Hier wird direkt auf die View „rv“ zugegriffen. Voraussetzung dafür sind

1. der kotlinx-Import
2. der kotlin-android-extensions plugin Eintrag in gradle.build

Vehicle.kt

```kotlin
package at.htl.simplerecyclerview.model

import at.htl.simplerecyclerview.R

data class Vehicle(val brand: String, val model: String, val image: Int)

fun getSampleVehicles(): List<Vehicle> {

    return listOf(
            Vehicle("Opel", "1200", R.drawable.opel_1200_1960),
            Vehicle("Opel", "GT", R.drawable.opel_gt_1970),
            Vehicle("Opel", "Kadett B", R.drawable.opel_kadett_b_1970),
            Vehicle("Opel", "Kapitän P 2,5", R.drawable.opel_kapitaen_1956),
            Vehicle("Opel", "Rekord C", R.drawable.opel_rekord_c_1970)
    )
}
```

VehicleAdapter.kt

```
package at.htl.simplerecyclerview

import android.app.Activity
import android.support.v7.widget.RecyclerView
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import at.htl.simplerecyclerview.model.Vehicle
import kotlinx.android.synthetic.main.card_vehicle.view.*

class VehicleAdapter(val vehicles: List<Vehicle>
            ) : RecyclerView.Adapter<VehicleAdapter.VehicleViewHolder>() {

    class VehicleViewHolder(val vehicleItem: View) : RecyclerView.ViewHolder(vehicleItem)

    override fun onBindViewHolder(holder: VehicleViewHolder?, position: Int) {
        if (holder != null) {
            val vehicle = vehicles[position]
            holder.vehicleItem.tv_brand.text = vehicle.brand
            holder.vehicleItem.tv_model.text = vehicle.model
            holder.vehicleItem.iv_icon.setImageResource(vehicle.image)
            holder.vehicleItem.setOnClickListener{MainActivity
                                                    .mainActivityContext
                                                    .onClickListenerWithPosition(position) }
        }
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): VehicleViewHolder {
        val view = LayoutInflater
                .from(parent.context)
                .inflate(R.layout.card_vehicle, parent, false)
        return VehicleViewHolder(view)

    }

    override fun getItemCount() = vehicles.size
}
```

Diese drei Methoden werden mit <Ctrl>-O überschrieben

MainActivity.kt

Nun wird noch der Click Listener zur Klasse MainActivity hinzugefügt

```
fun onClickListenerWithPosition(position: Int) {
    println("Clicked on $position: ${getSampleVehicles()[position]}")
}
```

Noch Fragen?