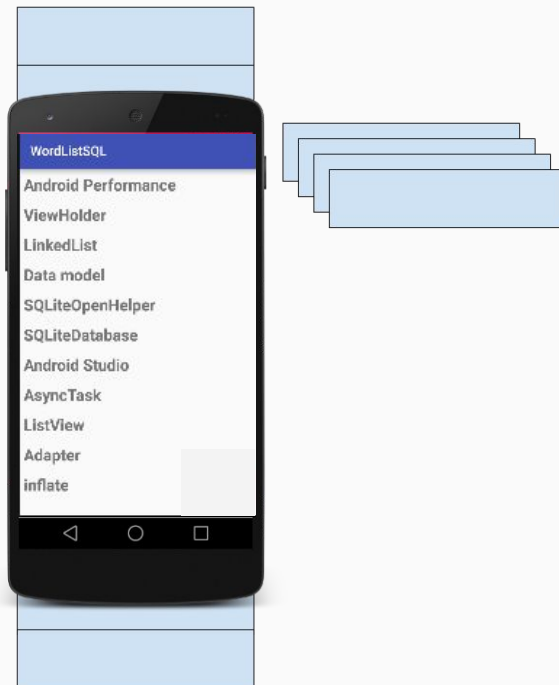# RecyclerView

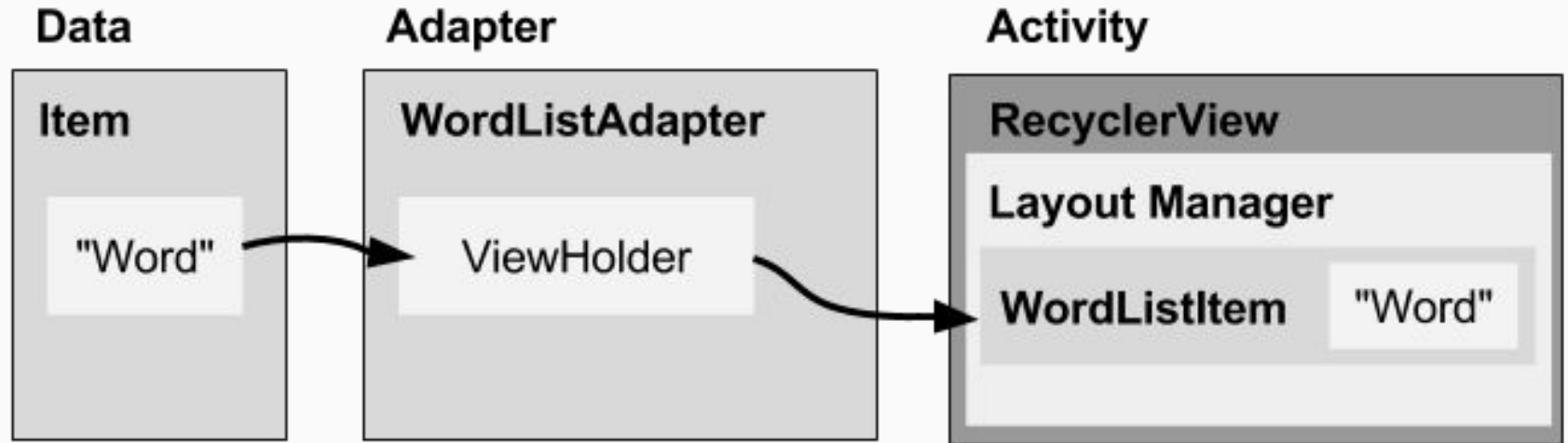# RecyclerView

RecyclerView est un conteneur scrollable pour afficher une grande quantité de donnée de façon efficace:

- Il crée un nombre limité de Views
- Il les réutilise en remplaçant les données et les listeners (re-bind) sans les recréer
- Met à jour les données rapidement

Composants

Add dependency to app/build.gradle

Add `RecyclerView` dependency to build.gradle if needed:

```gradle
dependencies {
  // ...
  implementation 'androidx.recyclerview:recyclerview:1.0.0'
  // ...
}
```

Add RecyclerView to XML Layout

```xml
<android.support.v7.widget.RecyclerView
    android:id="@+id/recyclerview"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layoutManager="android.support.v7.widget.LinearLayoutManager" />
```

Create layout for 1 list item

```xml
<LinearLayout

    android:layout_width="match_parent"

    android:layout_height="match_parent">

    <TextView

        android:id="@+id/word"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content" />

</LinearLayout>
```

```kotlin
class WordListAdapter(val wordList: Word)
: RecyclerView.Adapter<WordListAdapter.WordViewHolder>() {
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): WordViewHolder {}

    override fun getItemCount(): Int {}

    override fun onBindViewHolder(holder: WordViewHolder, position: Int) {}

    class WordViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
        fun bind(word: Word) {}
    }
}
```
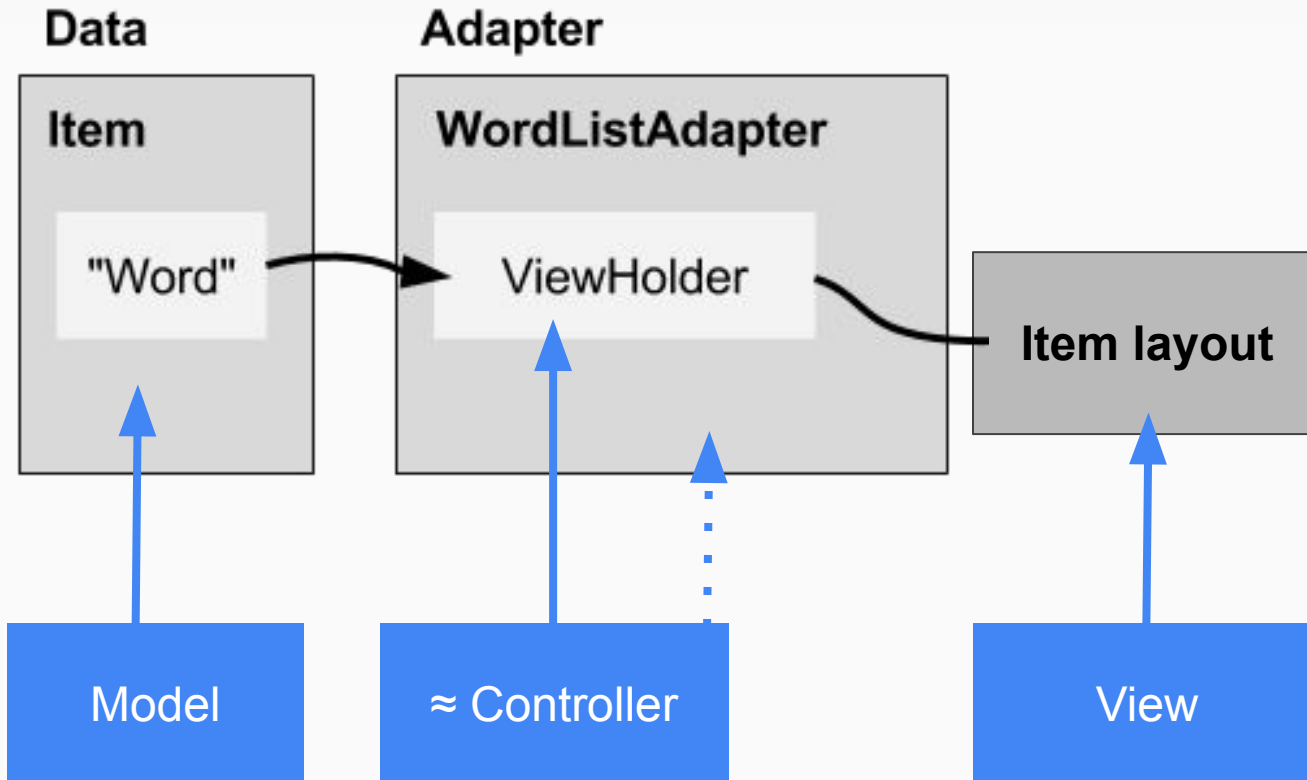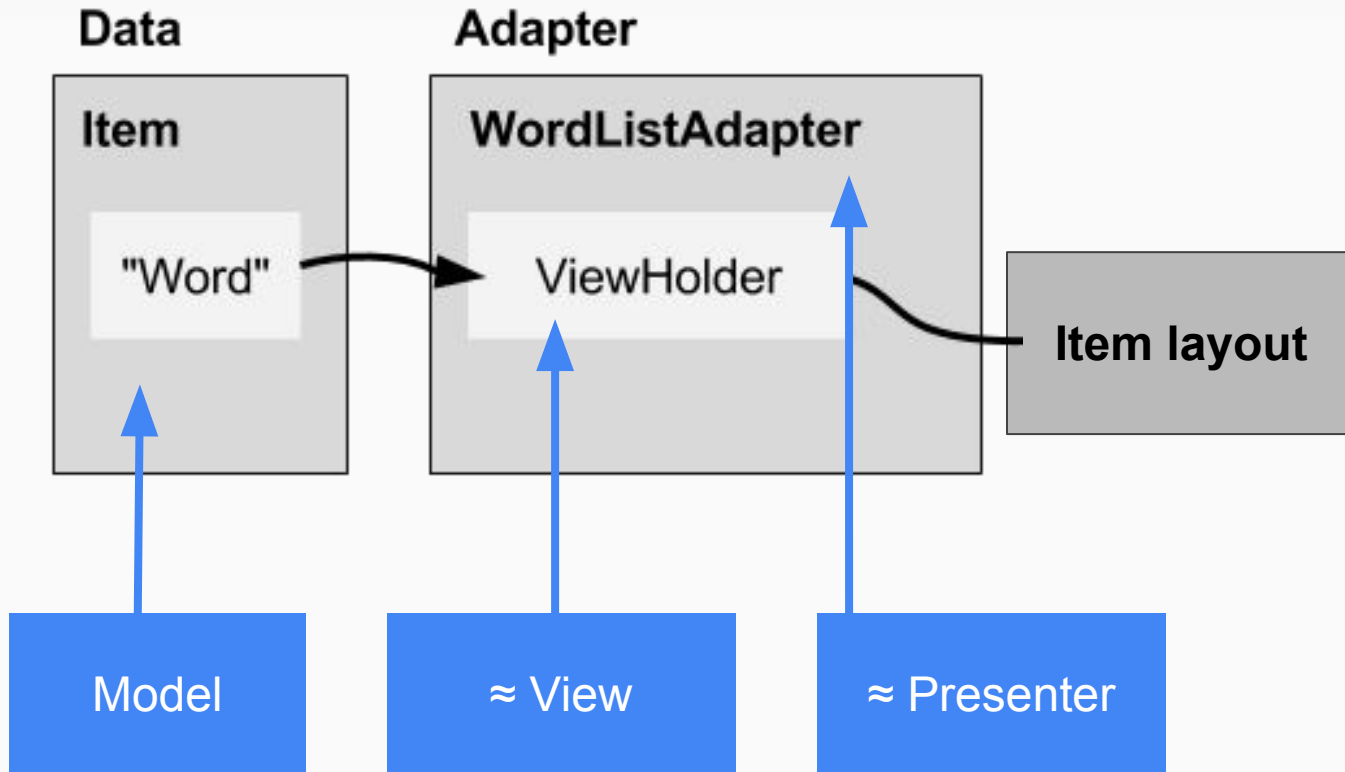
```kotlin
val wordList = listOf("word#1", "word #2")
val adapter = WordListAdapter(wordList)
recyclerview.adapter = adapter
recyclerview.layoutManager = LinearLayoutManager(this)
// on peut aussi utiliser app:layoutManager="..."
```

Implémentation de l'Adapter et configuration du RecyclerView dans onCreate()

Parenthèse Architecture: MVC

Parenthèse Architecture: MVP

Parenthèse Architecture: MVVM

Data

Adapter

Item

"Word"

WordListAdapter

ViewHolder

Item layout

Model

≈ View

WordViewModel