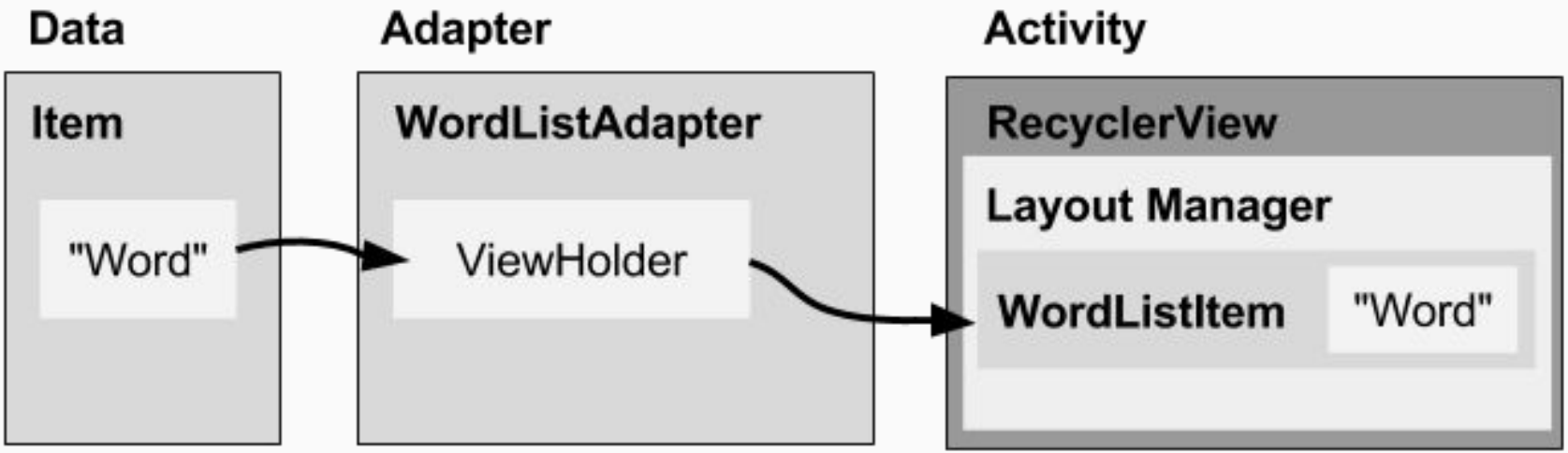
# RecyclerView

# 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


Data
Adapter
Activity
Item
"Word"
WordListAdapter
ViewHolder
RecyclerView
Layout Manager
WordListItem
"Word"

# Add dependency to app/build.gradle

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

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

# Add RecyclerView to XML Layout

```
<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

```
<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>
```

```
class WordListAdapter(val wordList: Word)
: RecyclerView.Adapter<WordListAdapter.WordViewHolder>() {
   override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): WordViewHolder {
       // Create the empty cell
   }
   override fun getItemCount(): Int {}

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

   inner class WordViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
     fun bind(word: Word) {} // Fill the cell with info
   }
}
```

```
    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()