

# Bài 5: ViewHolder & RecyclerView

Giáo viên: Lê Quốc Anh

# Nội dung

---

1. LayoutInflater trong Android
2. ViewHolder
3. RecyclerView

# 1. LayoutInflater trong Android

---

- LayoutInflater là 1 component giúp bạn chuyển layout file(Xml) thành View(Java code) trong Android. Sử dụng nó trong phương thức onCreateView của fragment hoặc phương thức getView khi custom adapter

# Tạo đối tượng LayoutInflater

---

- 1. LayoutInflater là 1 System Service của Android và cách sử dụng của nó giống như các System Service khác như khi bạn sử dụng WINDOW\_SERVICE, ALARM\_SERVICE hay LOCATION\_SERVICE.
  - `LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);`
- 2. Sử dụng static method của LayoutInflater
  - `LayoutInflater inflater = LayoutInflater.from(context);`

# Phương thức Inflate

---

- Công việc của LayoutInflater là đọc xml layout file và chuyển đổi các thuộc tính của nó thành 1 View trong Java code. Sau khi có đối tượng LayoutInflater, ta có thể dùng phương thức inflate để chuyển đổi 1 xml layout file thành 1 View trong java.
- Ta có 2 phương thức inflate với số lượng tham số khác nhau:
  - 1. `View view = inflater.inflate(int resource, ViewGroup parent)`
  - 2. `View view = inflater.inflate(int resource, ViewGroup parent, boolean attachToRoot)`

# Phương thức Inflate

---

- ❑ Tham số thứ nhất là: `int resource`, nó chính là xml layout file mà chúng ta muốn chuyển đổi thành View.
- ❑ Tham số thứ hai là: `ViewGroup parent`, nó là `ViewGroup` nơi mà xml layout file(tham số thứ nhất) có thể được nhúng vào, `LayoutInflater` sẽ chuyển đổi xml layout file thành View và sử dụng các thuộc tính phù hợp với `ViewGroup parent`.
- ❑ Tham số thứ ba là: `attachToRoot`, khi mà `attachToRoot=true` thì ngay sau khi quá trình chuyển đổi xml file(resource) thành View hoàn thành thì nó sẽ nhúng View đó vào `ViewGroup parent` (**RIGHT NOW**) , khi `attachToRoot = false` thì nó chỉ chuyển đổi xml file(resource) thành View trong java mà không thêm ngay vào `ViewGroup`(**NOT NOW**)

## 2. ViewHolder

---

- ViewHolder là một thuật ngữ nói chung nhằm để tùy chỉnh một View
- ViewHolder cho phép bạn truy cập các thành phần của danh sách. Đặc biệt, nó giúp bạn tránh việc phải thực hiện `findViewById()` và làm cho ứng dụng mượt hơn.

# Trường hợp 1: Không sử dụng ViewHolder

---

- Lần đầu gọi phương thức, convertView là null. Chúng ta phải inflate layout cho item của ListView và tìm TextView với findViewById().
- Lần gọi thứ hai, convertView không còn null. Chúng ta không cần phải inflate layout, nhưng phải findViewById() cho TextView một lần nữa.
- Những lần gọi tiếp theo, convertView không null. Nhưng chúng ta vẫn phải findViewById(), nếu việc này lặp lại nhiều lần sẽ làm giảm hiệu suất cho ứng dụng.



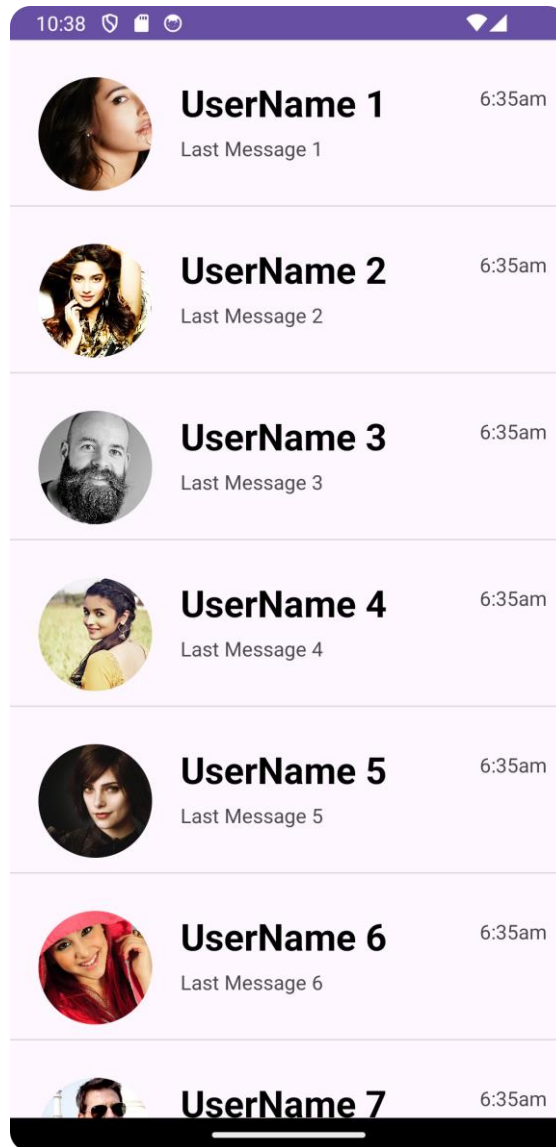
# Trường hợp 2: Sử dụng ViewHolder

---

- Lần đầu gọi phương thức, convertView là null. Chúng ta phải inflate layout cho item của ListView, nó chính là ViewHolder, tìm TextView với findViewById() và gán nó vào ViewHolder sau đó thiết lập ViewHolder như tag của convertView.
- Lần gọi thứ hai, convertView không còn null. Chúng ta không cần phải inflate layout cũng không cần phải findViewById() bởi vì bây giờ chúng ta có thể sử dụng TextView như một thành phần của ViewHolder.
- Những lần gọi tiếp theo, convertView không null và findViewById() không bao giờ được gọi nữa. Như vậy sẽ làm ứng dụng của chúng ta mượt hơn.

# Demo xây dựng ứng dụng

---



# 3. RecyclerView

---

- RecyclerView là phiên bản ListView nâng cao và linh hoạt hơn
- Trong mô hình RecyclerView, một số thành phần khác nhau làm việc cùng nhau để hiển thị dữ liệu
- RecyclerView lấp đầy chính nó bằng các chế độ xem do trình quản lý layout cung cấp.(chẳng hạn như LinearLayoutManager, GridLayoutManager)
- Các view trong list được đại diện bởi các đối tượng ViewHolder. Đối tượng này là một thể hiện của một lớp mà bạn định nghĩa bằng cách extend RecyclerView.ViewHolder

# 3. RecyclerView

---

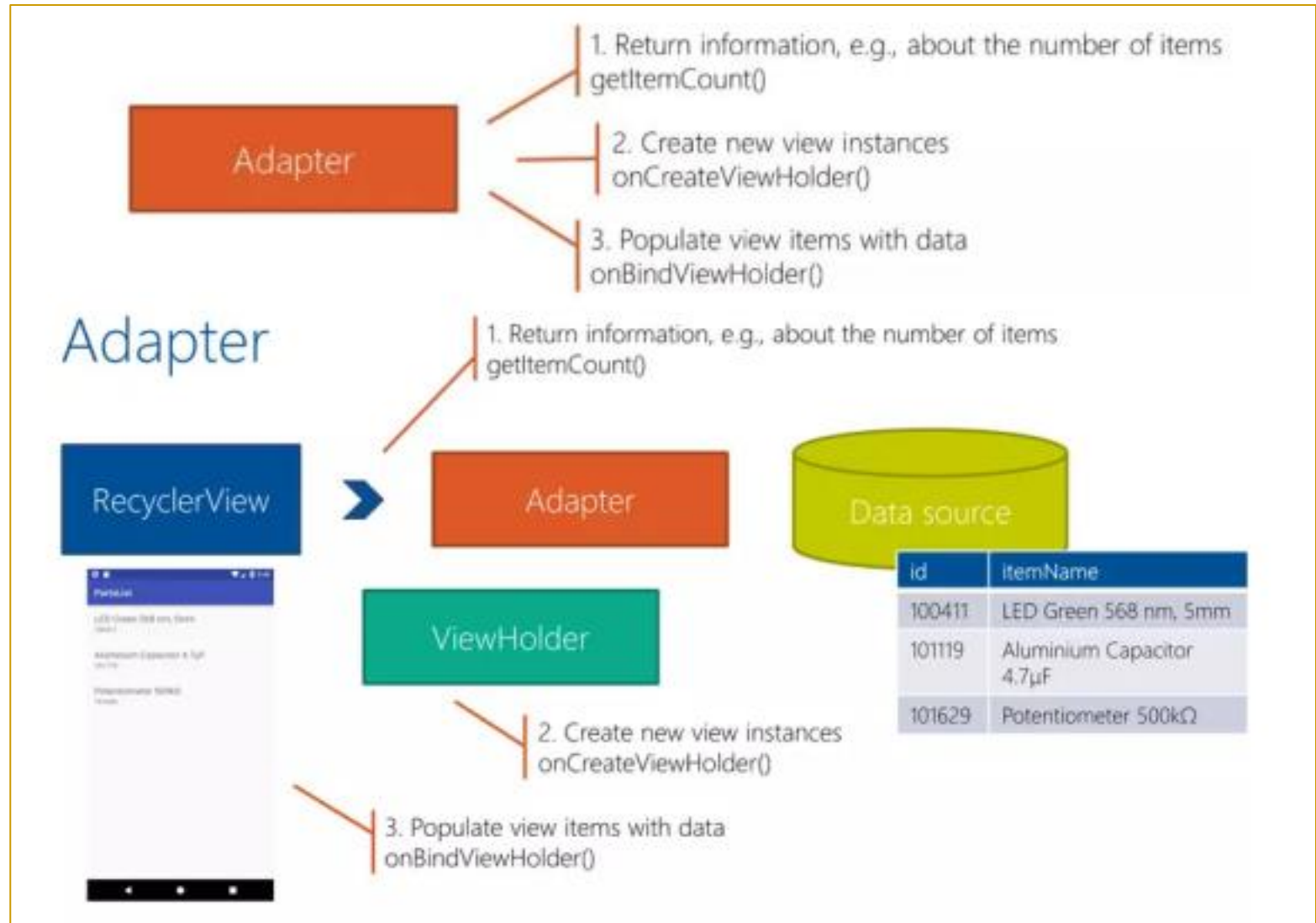
- Mỗi ViewHolder chịu trách nhiệm hiển thị một single item trong một view.
- RecyclerView chỉ tạo ra nhiều ViewHolder khi cần để hiển thị phần trên màn hình của nội dung động, cộng thêm một vài phần bổ sung
- Khi người dùng cuộn qua danh sách, recyclerview lấy các view nằm ngoài màn hình và kết nối chúng với dữ liệu đang cuộn lên màn hình

# 3. RecyclerView

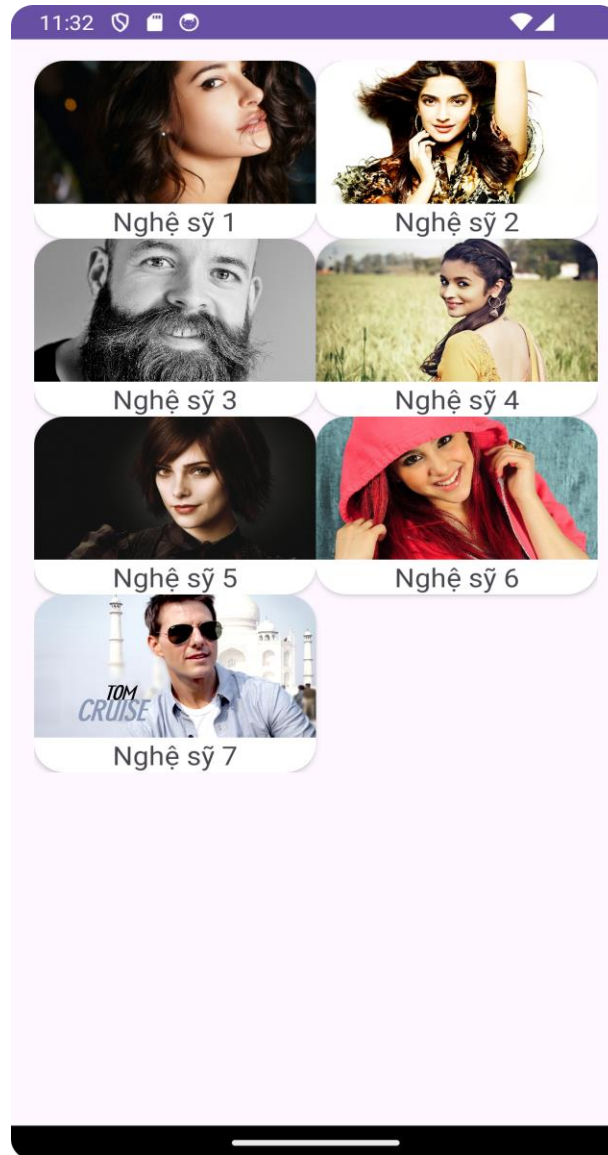
---

- Các đối tượng ViewHolder được quản lý bởi Adapter mà bạn tạo ra bằng cách kế thừa RecyclerView.Adapter.
- Adapter tạo ra các view khi cần thiết. Adapter cũng liên kết ViewHolder với dữ liệu của chúng. Nó thực hiện điều này bằng cách chỉ ViewHolder cho một vị trí và gọi phương thức onBindViewHolder () của adapter.
- Phương thức đó sử dụng vị trí của ViewHolder để xác định nội dung nên dựa trên vị trí danh sách của nó.
- Mô hình RecyclerView này thực hiện rất nhiều công việc tối ưu hóa. Khi danh sách được điền lần đầu tiên, nó sẽ tạo và liên kết một số ViewHolder ở hai bên của danh sách.

# 3. RecyclerView



# Demo xây dựng ứng dụng



---

**THANK YOU**  
**for**  
**YOUR ATTENTION**

