Recycler View

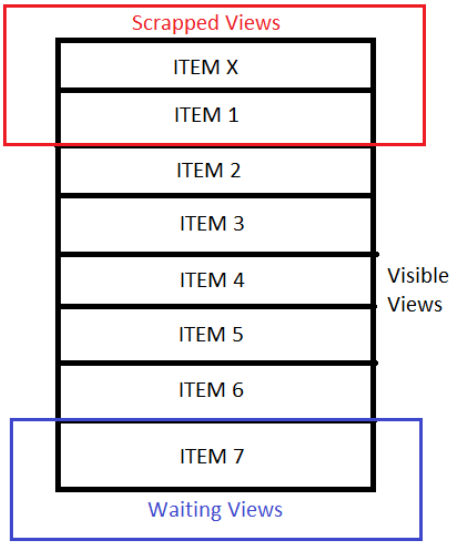
- **Introduction**
- **Working of recycler view**
- **Implementing recycler view**
- **Optimizing recycler view**

Introduction

1. Recycler view is the View group which hold list of data to be displayed on the screen
2. There are 3 major components of recycler view
   a. Adapter
   b. View holder
   c. Layout Manager
3. Adapter is the class which extends **RecyclerView.Adapter** and holds list of data to be displayed on each item of Recycler view
4. View Group is the class which extends **RecyclerView.ViewHolder** and helps to draw UI for each item
5. Layout manager is used to display items in linear view or grid view

Working of recycler view

1. Consider we have Item X to Item 7 in the recycler view and initially item x to item 4 are visible to the user are called as visible view
2. In Step 2 if user scrolls up now item 1 to item 5 are visible to the user and item x moved to scrapped views
3. Scrapped views are collection of views once visible to the user
4. In step 3 user scrolls one more item up now item 2 to 6 are visible to the user and item 7 is waiting to be displayed is called waiting views
5. In step 4 when user tries to load item 7 this item make use of scrapped views are called as dirty views



➢ Implementing recycler view

1. Create a model class which store the data of each item and make a list of type model class
2. Create a View holder class which extends **RecyclerView.ViewHolder** and define all the view for each item in this class
3. Create an adapter class which extends **RecyclerView.Adapter<VH>** of type view holder and override
   a. onCreateViewHolder()
   b. onBindViewHolder()
   c. getItemCount()
4. onCreateViewHolder() method is used to inflate item layout using LayoutinFlater.inflate()
5. onVindViewHolder() method is used to set views to the view holder class
6. getItemCount() returns size of the item list
7. create the object of adapter class by passing list of items and set this adapter object to recycler view using setAdapter() method
8. In the last set the layout manager to the recycler view layout manager can be LinearLayoutManager or GridLayoutManager

```
override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): RecyclerViewHolder {
    val layoutInflater = LayoutInflater.from(context)
    val view = layoutInflater.inflate(R.layout.item_recycler_view,parent, attachToRoot: false)

    return RecyclerViewHolder(view)
}

override fun onBindViewHolder(holder: RecyclerViewHolder, position: Int) {
    val programmingLanguageData = list?.get(position)

    holder.iconIV?.setImageResource(programmingLanguageData!!.icon)
    holder.titleTV?.setText(programmingLanguageData!!.title)
    holder.subTitle?.setText(programmingLanguageData!!.subTitle)
}

override fun getItemCount(): Int {
    return list?.size!!
}
```

Adapter class

```
val recyclerView = findViewById<RecyclerView>(R.id.recycler_view)
val adapter = RecyclerAdapter( context: this, list!!)
recyclerView.adapter = adapter
recyclerView.layoutManager = LinearLayoutManager( context: this)
```

Set adapter in activity

➢ Optimizing recycler view

1. Use the image loading libraries like glide or Picasso to avoid un responsive UI
2. Set fixed image height and width to avoid flickering.
3. Do less work on onBindViewHolder method
4. Use notify item api
    a. notifyItemRemoved(position)
    b. notifyItemChanged
    c. notifyItemInserted
    d. notifyItemRangeInserted(from, to)
5. Avoid using of nested view
6. Use setItemViewCacheSized(size) to retain views which are just screlled