

Recycle View Extension

```
// ----- Recycleview Extension -----
new *
fun <T> RecyclerView.bindData(
    data: List<T>,
    layoutRes: Int,
    bindFunc: (View, T) -> Unit,
    clickListener: ((T) -> Unit)? = null
) {
    adapter = object : RecyclerView.Adapter<RecyclerView.ViewHolder>() {
        override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {
            val view = LayoutInflater.from(parent.context).inflate(layoutRes, parent, attachToRoot: false)
            return ViewHolder(view)
        }

        override fun onBindViewHolder(holder: RecyclerView.ViewHolder, position: Int) {
            val item = data[position]
            bindFunc(holder.itemView, item)
            clickListener?.let { listener ->
                holder.itemView.setOnClickListener { listener(item) }
            }
        }

        override fun getItemCount() = data.size
    }

    new *
    inner class ViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView)
}
}
```

Call Extension

```
viewBinding.recycleList.bindData(
    data = listOf(Student( fname: "Chirag"), Student( fname: "Solanki")),
    layoutRes = R.layout.row_layout,
    bindFunc = { view, item -> view.findViewById<Button>(R.id.devButton).text = item.fname },
    clickListener = { item -> onItemClick(item.fname) }
)
```