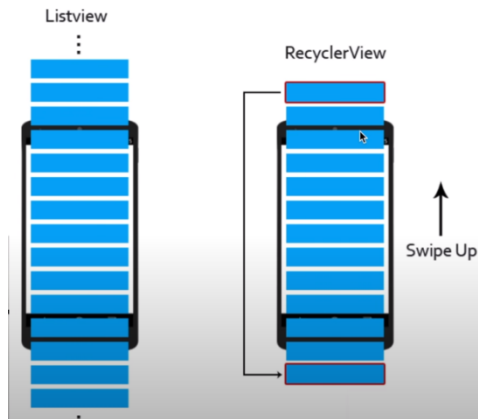


News Park (Android Application)

Overview:-

We are Using here Recycler view to show News in the interface.

But what is Recycler view and why we are not using recycler view?



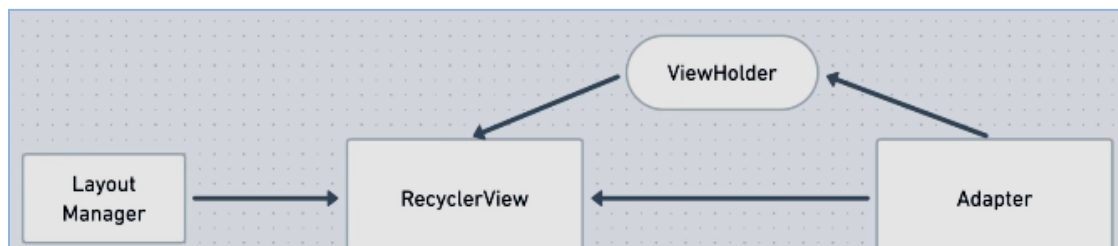
The list view has Unlimited list of all the data it caches as soon as it reaches internet which may cause more data usage in compared to Recycler view as Recycler only creates the view that can be visible in single screen else all the scrolled view recycled when it crosses the screen top line, and given back to the next of bottom most view, Which saves memory of the view creation.

Recycler has better performance than list view.

All the major apps like Instagram, facebook, Youtube App All of the storing apps uses Recycler view to save the memory and efficiently.

Recycler view also gives an option to put views horizontally, vertically and staggered and grid too. But List only had option to Horizontal.

What are things needed to create Recycler view?



Layout Manager- Holds Management of view(grid/horizontal...)

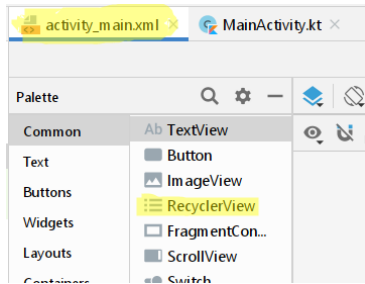
Adapter- provides data in recycler view. So Adapter only deals with backend and provides data from that to the RecyclerView.

Adapter takes help of view hold as Adapter receives xml files then it clubs to the recycler view with help of view hold.

Starting with Android Studio:-

- ✚ Create an Empty project
- ✚ Name- NewsFresh

Go to Activity_main.xml >>delete all and then



pick and drag to the frame

If ask for project dependency click yes.

then the dependencies on the Sdk will come to this project. (You can see the dependency on build.gradle(app))

But before moving further add

```
plugins {  
    id 'com.android.application'  
    id 'kotlin-android'  
    id 'kotlin-android-extensions'  
}
```

in the same(top).

Do changes in Activity_main.xml

```
<androidx.recyclerview.widget.RecyclerView  
    android:layout_width="0dp"  
    android:layout_height="0dp"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintLeft_toLeftOf="parent"  
    app:layout_constraintRight_toRightOf="parent"  
    app:layout_constraintBottom_toBottomOf="parent"  
/>
```

Now we need the components of recycler view

In MainActivity.kt

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
    }  
}
```

```

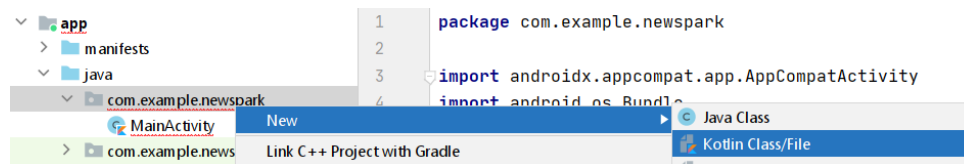
        recyclerView.layoutManager=LinearLayoutManager(this)
    }
}

```

Linear Layout added to the Layout Manager. It can be grid or others.

Now we need to add Adapter

Create a new java class



Name it NewsListAdapter

Now we need to extend this class as with recycler view

```

import android.view.View
import androidx.recyclerview.widget.RecyclerView

```

```

class NewsListAdapter: RecyclerView.Adapter<> {

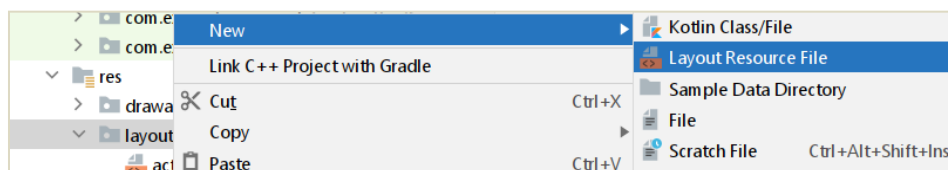
```

```

    class NewsViewHolder(itemView: View): RecyclerView.ViewHolder(itemView) {

```

Now we need to make small items in the RecyclerView what it should be
res>>Layout>>



Just name it item_views

And get ready for a new layout.

Fir now just add a text view it item_views.xml

```

<?xml version="1.0" encoding="utf-8" ?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="16dp"
    xmlns:app="http://schemas.android.com/apk/res-auto">

```

```

<TextView
    android:id="@+id/titlee"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="TextView"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    android:textSize="18sp"
    android:textColor="@color/teal_200"
/>
</androidx.constraintlayout.widget.ConstraintLayout>

```

Now need to call the id of this text view into Adapter

Go to newsListAdapter.kt

We need to implement the members

Alt+enter > implement all 3 members on class newsListAdapter

3 override functions will come

```

class NewsListAdapter: RecyclerView.Adapter<> {
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ??? {
        TODO("Not yet implemented")
    }

    override fun onBindViewHolder(holder: ???, position: Int) {
        TODO("Not yet implemented")
    }

    override fun getItemCount(): Int {
        TODO("Not yet implemented")
    }
}

```

On onCreateViewHolder it stores the view area of screen

Getitemcount tells number of count

GetItemCount binds data with it,

Adapter needs a data(covered from database)

For now let's say

ArrayList of strings passed

Now lets do Adapter Ready

do all the changes so that adapter takes the data and puts to the view.

all explanations in the comments of code

NewsListAdapter.kt

```
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView

class NewsListAdapter(val items:ArrayList<String>):
RecyclerView.Adapter<NewsViewHolder>() {

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): NewsViewHolder
    {
        //Layout inflater added
        val
view=LayoutInflater.from(parent.context).inflate(R.layout.item_news,parent,false)
        //added item_news.xml as a view to be used
        //which is passed below
        return NewsViewHolder(view)
        //Needed instance of NewsViewHolder
    }

    override fun onBindViewHolder(holder:NewsViewHolder, position: Int) {
        //binds data
        //position to bind
        val currentItem=items[position]
        holder.titleview.text=currentItem
    }

    override fun getItemCount(): Int {
        return items.size
    }
}

class NewsViewHolder(itemView: View): RecyclerView.ViewHolder(itemView){

    val titleview: TextView = itemView.findViewById(R.id.titlee)
}
```

Let's Link recycler view with Adapter.

MainActivity.kt

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    recyclerView.layoutManager=LinearLayoutManager(this)
    val items=fetchData()
    val adapter:NewsListAdapter= NewsListAdapter(items)
    //Now needed to link
    recyclerView.adapter=adapter
}
```

```

        private fun fetchData():ArrayList<String>{
            val list=ArrayList<String>()
            for(i in 0 until 100){//added 100 items//which says total 100 list
will be there
                list.add("Item $i")
            }
            return list
        }
    }
}

```

Now Run the Application and check for the list of 100 items

Working Fine

Now we add the onclick listener to the items

```

override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): NewsViewHolder {
    //Layout inflater added
    val
    view=LayoutInflater.from(parent.context).inflate(R.layout.item_news,parent,false)
    //added item_news.xml as a view to be used
    //which is passed below.
    //adding onclick
    val viewHolder=NewsViewHolder(view)
    view.setOnClickListener {
        listener.onItemClick(items[viewHolder.adapterPosition])
    }

    return viewHolder
    //Needed instance of NewsViewHolder
}

```

Now go to MainActivity.kt

```

class MainActivity : AppCompatActivity(), NewsItemsClicked {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        recyclerView.layoutManager=LinearLayoutManager(this)
        val items=fetchData()
        val adapter= NewsListAdapter(items,this)

        //Now needed to link
        recyclerView.adapter=adapter
    }

    private fun fetchData():ArrayList<String>{
        val list=ArrayList<String>()
        for(i in 0 until 100){//added 100 items//which says total 100 list
will be there
            list.add("Item $i")
        }
        return list
    }
}

```

```

    }

    override fun onItemClick(item: String) {
        TODO("Not yet implemented")
    }
}

```

To get yellow one needs to implement onItemClick to add fun in MainActivity

```

    override fun onItemClick(item: String) {
        Toast.makeText(this, "clicked item is $item", Toast.LENGTH_LONG).show()
    }
}

```

Now Run the App you can see the Toast is also working fine with saying item clicked.

The First phase of project completed and now its time for loading data n adhere the magic of API is needed to load the images.

Load data from API and show in RecyclerView

For that we will use [Newsapi.org](https://newsapi.org)

- It has varieties of source like [Endpoints](#)
 - [Everything](#)
 - [Top headlines](#)
 - [Sources](#)

Request Parameters are:-

- COUNTRY
- Category
- Sources
- pagesize -The number of results to return per page (request). 20 is the default, 100 is the maximum.
- page - Use this to page through the results if the total results found is greater than the page size.
- Query

Login to the website and get the API key

For me its 32c647cb10f74ab18ebc263c8a910e4c

Top business headlines in the US right now

<https://newsapi.org/v2/top-headlines?country=us&category=business&apiKey=32c647cb10f74ab18ebc263c8a910e4c>

Now we need to update the fetch data function

But before that we need to add volley Library(Singleton work)

<https://developer.android.com/training/volley>

Gradle Scripts>> build.gradle(Module:MemeShareApp.app)

```
dependencies {  
  
    implementation "org.jetbrains.kotlin:kotlin-stdlib:$kotlin_version"  
    implementation 'androidx.core:core-ktx:1.3.1'  
    implementation 'androidx.appcompat:appcompat:1.2.0'  
    implementation 'com.google.android.material:material:1.2.1'  
    implementation 'androidx.constraintlayout:constraintlayout:2.0.1'  
    testImplementation 'junit:junit:4.+'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'  
  
    implementation("com.android.volley:volley:1.1.1")  
  
}
```

Now needed Glide Android Library for Android images.

<https://github.com/bumptech/glide>

Gradle Scripts>>build.gradle(project)

```
allprojects {  
    repositories {  
        google()  
        mavenCentral()  
        jcenter() // Warning: this repository is going to shut down soon  
    }  
}
```

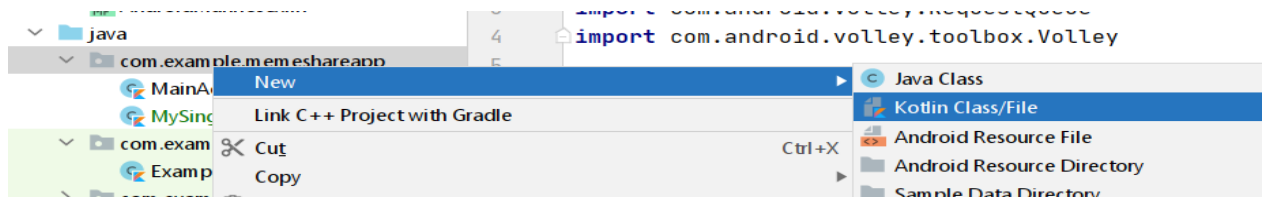
Gradle Scripts>>build.gradle(App)

```
dependencies {  
    implementation 'com.github.bumptech.glide:glide:4.12.0'  
    annotationProcessor 'com.github.bumptech.glide:compiler:4.12.0'  
}
```

Sync now

Singleton work

Put it in a separate class of kotlin



Name it as MySingleton

And after importing All it will look like

MySingleTon.kt

```
import android.content.Context
import com.android.volley.Request
import com.android.volley.RequestQueue
import com.android.volley.toolbox.Volley

class MySingleton constructor(context: Context) {
    companion object {
        @Volatile
        private var INSTANCE: MySingleton? = null
        fun getInstance(context: Context) =
            INSTANCE ?: synchronized(this) {
                INSTANCE ?: MySingleton(context).also {
                    INSTANCE = it
                }
            }
    }

    private val requestQueue: RequestQueue by lazy {
        // applicationContext is key, it keeps you from leaking the
        // Activity or BroadcastReceiver if someone passes one in.
        Volley.newRequestQueue(context.applicationContext)
    }

    fun <T> addToRequestQueue(req: Request<T>) {
        requestQueue.add(req)
    }
}
```

Update the MainActivity.kt

```
private fun fetchData():{
    val url= "https://newsapi.org/v2/top-
headlines?country=in&category=business&apiKey=32c647cb10f74ab18ebc263c8a910e4
c"
```

```

        val jsonObjectRequest = JsonObjectRequest(
            Request.Method.GET,
            url,
            null,
            Response.Listener{

            },
            Response.ErrorListener {

            }
        )
    }
}

```

Now running the website directly

It shows some of the following JSON Objects.

/ 20210529114315

// https://newsapi.org/v2/top-headlines?country=in&category=business&apiKey=32c647cb10f74ab18ebc263c8a910e4c

```

{
  "status": "ok",
  "totalResults": 70,
  "articles": [
    {
      "source": {
        "id": null,
        "name": "Livemint"
      },
      "author": "Staff Writer",
      "title": "Petrol price crosses ₹100 mark in Mumbai as fuel prices hiked again today. Check rates in your city - Mint",
      "description": "In Mumbai, petrol was retailing at ₹100.19 whereas diesel was priced at ₹92.17 per litre.",
      "url": "https://www.livemint.com/news/india/petrol-price-crosses-rs-100-mark-in-mumbai-as-fuel-prices-touch-record-highs-11622257788667.html",
      "urlToImage": "https://images.livemint.com/img/2021/05/29/600x338/petrol_diesel_price-koBB--621x414@LiveMint_1622257815373.JPG",
      "publishedAt": "2021-05-29T03:26:51Z",
      "content": "Fuel prices were hiked again on Saturday taking petrol price in Mumbai over 100 per litre for the first time. Petrol price was hiked up to 26 paise while the price of diesel went up around 30 paise a... [+54 chars]"
    }
  ]
}

```

Only these three things are needed for us to be showed in my news APP

To load this JSON object.

Create a new class

Name it News

And update it as

```
package com.example.newspark
```

```
data class News {  
    val title: String,  
    val author: String,  
    val url: String,  
    val imageUrl: String  
}
```

```
{  
  "status": "ok",  
  "totalResults": 70,  
  "articles": [  
    {  
      "source": {  
        "id": null,  
        "name": "Livemint"  
      },  
      "author": "Staff Writer",  
      "title": "Petrol price cross",  
      "description": "In Mumbai, p",  
      "url": "https://www.livemint.com/11622257788667.html",  
      "urlToImage": "https://image"
```

All these are in Array of articles so call array of articles

go to MainActivity.kt

```
private fun fetchData(): {  
    val url= "https://newsapi.org/v2/top-headlines?country=in&category=business&apiKey=32c647cb10f74ab18ebc263c8a910e4c"  
    val jsonObjectRequest = JsonObjectRequest(  
        Request.Method.GET,  
        url,  
        null,  
        Response.Listener {  
            val newsJsonArray = it.getJSONArray("articles")  
            val NewsArray=ArrayList<News>()  
        })  
}
```

```

        ///to iterate one by one
        for(i in 0 until newsJsonArray.length()) {
            val newsJsonObject = newsJsonArray.getJSONObject(i)
            val news=News(
                newsJsonObject.getString("title"),
                newsJsonObject.getString("author"),
                newsJsonObject.getString("url"),
                newsJsonObject.getString("urlToimage")
            )
            newsArray.add(news)
        }
    },
    Response.ErrorListener {
    }
)

```

In news List Adapter.kt

```

class NewsListAdapter(private val listener:NewsItemsClicked):
RecyclerView.Adapter<NewsViewHolder>() {

    private val items:ArrayList<News> = ArrayList()

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
NewsViewHolder {
        //Layout inflater added
        val
view=LayoutInflater.from(parent.context).inflate(R.layout.item_news,parent,false)

        //added item_news.xml as a view to be used
        //which is passed below.
        //adding onclick
        val viewHolder=NewsViewHolder(view)
        view.setOnClickListener {
            listener.onItemClick(items[viewHolder.adapterPosition])
        }

        return viewHolder
        //Needed instance of NewsViewHolder
    }

    override fun onBindViewHolder(holder:NewsViewHolder, position: Int) {
        //binds data
        //position to bind
        val currentItem= items[position]
        holder.titleview.text=currentItem.title
    }

    override fun getItemCount(): Int {
        return items.size
    }
}

```

```

    }

    fun updateNews(updatedNews: ArrayList<News>){
        items.clear()
        items.addAll(updatedNews)

        notifyDataSetChanged()
    }
}

```

when notify called all above 3 functions are called.

Finally, My News.kt looks like

```
package com.example.newspark
```

```
data class News (
    val title: String,
    val author: String,
    val url: String,
    val urlToImage: String
)
```

and MainActivity looks like

```
package com.example.newspark
```

```
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
//import android.widget.Toast
import androidx.recyclerview.widget.LinearLayoutManager
import com.android.volley.Request
import com.android.volley.Response
import com.android.volley.toolbox.JsonObjectRequest
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity(), NewsItemsClicked {

    private lateinit var mAdapter:NewsListAdapter

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        recyclerView.layoutManager=LinearLayoutManager(this)
        fetchData()
        mAdapter= NewsListAdapter(this)

        //Now needed to link
        recyclerView.adapter=mAdapter
    }

    private fun fetchData() {
        val url =
            "https://newsapi.org/v2/top-
            headlines?country=in&category=business&apiKey=32c647cb10f74ab18ebc263c8a910e4"
    }
}

```

c"

```
val jsonObjectRequest = JsonObjectRequest(
    Request.Method.GET,
    url,
    null,
    Response.Listener {
        val newsJsonArray = it.getJSONArray("articles")
        val newsArray = ArrayList<News>()
        ///to iterate one by one
        for (i in 0 until newsJsonArray.length()) {
            val newsJsonObject = newsJsonArray.getJSONObject(i)
            val news = News(
                newsJsonObject.getString("title"),
                newsJsonObject.getString("author"),
                newsJsonObject.getString("url"),
                newsJsonObject.getString("urlToImage")
                //data parsing
            )
            newsArray.add(news)
        }
        mAdapter.updateNews(newsArray)
    },
    Response.ErrorListener {
    }
)
MySingleton.getInstance(this).addToRequestQueue(jsonObjectRequest)
}

override fun onItemClick(item: News) {
}
}
```

Decorating the User Interface

Add a image view

In item_news.xml

```
<ImageView
    android:layout_width="0dp"
    android:id="@+id/image"
    android:layout_height="200dp"
    android:scaleType="centerCrop"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"/>
```

```
<TextView
    android:id="@+id/title"
```

```

        android:layout_width="0dp"
        android:layout_height="wrap_content"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/image"
        android:textSize="18sp"
        android:maxLines="2"
        android:ellipsize="end"
        android:layout_marginTop="12dp"
        android:textColor="#212121"/>

```

```

<TextView
    android:id="@+id/author"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/title"
    android:layout_marginTop="6dp" />

```

Now to open chrome tabs

You must run <https://developer.chrome.com/docs/android/custom-tabs/integration-guide/>

Then add dependencies

```

dependencies {
    ...
    implementation "androidx.browser:browser:1.3.0"
}

```

then for custom Chrome browser use this

```

String url = "https://paul.kinlan.me/";
CustomTabsIntent.Builder builder = new CustomTabsIntent.Builder();
CustomTabsIntent customTabsIntent = builder.build();
customTabsIntent.launchUrl(this, Uri.parse(url));

```

In MainActivity.kt

```

    override fun onItemClick(item: News) {
        val builder = CustomTabsIntent.Builder();
        val customTabsIntent = builder.build();
        customTabsIntent.launchUrl(this, Uri.parse(item.url));
    }
}

```

Now our app is ready to be used.

Run the App

<http://api.mediastack.com/v1/news>

? access_key = 6fcacbfd6ec372a76101e6c2398e3c8e

http://api.mediastack.com/v1/news?access_key=6fcacbfd6ec372a76101e6c2398e3c8e