**Main Activity: FragDemo**

public class MainActivity extends AppCompatActivity {  
private AppBarConfiguration;  
private ActivityMainBinding binding;  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
super.onCreate(savedInstanceState);  
binding = ActivityMainBinding.inflate(getLayoutInflater());  
setContentView(binding.getRoot());  
setSupportActionBar(binding.toolbar);  
NavController = Navigation.findNavController(this, R.id.nav\_host\_fragment\_content\_main);  
appBarConfiguration = new AppBarConfiguration.Builder(navController.getGraph()).build();  
NavigationUI.setupActionBarWithNavController(this, navController, appBarConfiguration);  
binding.mainContent.mainLayout.setBackgroundColor(Color.parseColor("#FAFAFA"));  
binding.fab.setBackgroundTintList(  
ColorStateList.valueOf(Color.LTGRAY));  
binding.fab.setOnClickListener(new View.OnClickListener() {  
@Override  
public void onClick(View view) {  
ColorDrawable =(ColorDrawable) binding.mainContent.mainLayout.getBackground();  
int colorId = colorDrawable.getColor();  
if (colorId == Color.LTGRAY) {  
binding.mainContent.mainLayout.setBackgroundColor(Color.parseColor("#FAFAFA"));  
binding.fab.setBackgroundTintList(ColorStateList.valueOf(Color.LTGRAY));  
} else { binding.mainContent.mainLayout.setBackgroundColor(Color.LTGRAY);  
binding.fab.setBackgroundTintList(  
ColorStateList.valueOf(Color.parseColor("#FAFAFA")));  
}

Snackbar.make(view, "Like the new color?", Snackbar.LENGTH\_LONG)  
.setAnchorView(R.id.fab).setAction("Undo", new View.OnClickListener() {  
@Override  
public void onClick(View view) {  
if (colorId == Color.LTGRAY){ binding.mainContent.mainLayout.setBackgroundColor(Color.LTGRAY);  
 binding.fab.setBackgroundTintList( ColorStateList.valueOf(Color.parseColor("#FAFAFA")));  
} else {binding.mainContent.mainLayout .setBackgroundColor(Color.parseColor("#FAFAFA")); binding.fab.setBackgroundTintList(ColorStateList.valueOf(Color.LTGRAY));  
 }}}).show(); } });

List<ColorSpec> ColorSpecs = new ArrayList<>();  
List<String> ColorDescs= new ArrayList<>(Arrays.asList("BLACK","ORANGE","PURPLE"));  
List<Integer> ColorVals = new ArrayList<>(Arrays.asList(R.color.black,  
Color.rgb(255,165,0),Color.parseColor("#800080")));

for (int i = 0; i < ColorDescs.size(); i++){  
ColorSpec eachColor = new ColorSpec(ColorDescs.get(i),ColorVals.get(i));  
ColorSpecs.add(eachColor); }  
ColorSpecViewModel colorSpecViewModel= new ViewModelProvider(this).get(ColorSpecViewModel.class);  
colorSpecViewModel.loadColorList(ColorSpecs);  
}  
  
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
// Inflate the menu; this adds items to the action bar if it is present.  
getMenuInflater().inflate(R.menu.menu\_main, menu); return true; }  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 // Handle action bar item clicks here. The action bar will  
 // automatically handle clicks on the Home/Up button, so long  
 // as you specify a parent activity in AndroidManifest.xml.  
 int id = item.getItemId();  
 //noinspection SimplifiableIfStatement  
 if (id == R.id.action\_settings) {  
 Toast.makeText(this, "Clicked on settings", Toast.LENGTH\_SHORT).show();  
 return true;  
 } else if (id == R.id.action\_search) {  
 Toast.makeText(this, "Clicked on search", Toast.LENGTH\_SHORT).show();  
 return true;  
 } else if (id == R.id.action\_userprofile){  
 Toast.makeText(this, "Clicked on userprofile", Toast.LENGTH\_SHORT).show();  
 return true;  
 }  
 return super.onOptionsItemSelected(item);  
 }  
 @Override  
 public boolean onSupportNavigateUp() {  
NavController = Navigation.findNavController(this, R.id.nav\_host\_fragment\_content\_main);  
 return NavigationUI.navigateUp(navController, appBarConfiguration)  
 || super.onSupportNavigateUp();  
 }  
}

**Color Spec: FragDemo**

*public class ColorSpec {  
 private String ColorDesc;  
 private int ColorVal;  
  
 public String getColorDesc() {  
 return ColorDesc;  
 }  
  
 public void setColorDesc(String colorDesc) {  
 ColorDesc = colorDesc;  
 }  
  
 public int getColorVal() {  
 return ColorVal;  
 }  
  
 public void setColorVal(int colorVal) {  
 ColorVal = colorVal;  
 }  
  
 public ColorSpec(String colorDesc, int colorVal) {  
 ColorDesc = colorDesc;  
 ColorVal = colorVal;  
 }  
}*

**ColorSpecAdapter: FragDemo**

public class ColorSpecAdapter extends BaseAdapter {  
List<ColorSpec> ColorList;  
LayoutColoritemBinding coloritemBinding;  
public ColorSpecAdapter(List<ColorSpec> colorList) {  
ColorList = colorList;  
}  
@Override  
public int getCount() {  
return ColorList.size();  
}  
  
@Override  
public Object getItem(int i) {  
return ColorList.get(i);  
}  
  
@Override  
public long getItemId(int i) {  
return i;  
}  
  
@Override  
public View getView(int i, View, ViewGroup viewGroup) {  
if (view == null){  
coloritemBinding = LayoutColoritemBinding .inflate(LayoutInflater.from(viewGroup.getContext()),  
 viewGroup,false);  
 } coloritemBinding.txtViewColorItem.setText(ColorList.get(i).getColorDesc()); coloritemBinding.txtViewColorItem.setTextColor(ColorList.get(i).getColorVal());  
return coloritemBinding.getRoot();  
 }  
}

**ColorSpecViewModel: FragDemo**

public class ColorSpecViewModel extends ViewModel {  
 MutableLiveData<List<ColorSpec>> ColorList = new MutableLiveData<>();  
 public LiveData<List<ColorSpec>> getColorList(){  
 if (ColorList == null){  
 ColorList = new MutableLiveData<>();  
 }  
 return ColorList;  
 }  
 public void loadColorList(List<ColorSpec> colorList){  
 ColorList.setValue(colorList);  
 /\*Handler = new Handler();  
 handler.postDelayed(new Runnable() {  
 @Override  
 public void run() {  
 ColorList.setValue(colorList);  
 }  
 }, 1000);\*/  
 }  
}

**FirstFragment: FragDemo**

public class FirstFragment extends Fragment {  
List<ColorSpec> FragColors = new ArrayList<>();  
ColorSpecViewModel;  
ColorSpecAdapter;

private FragmentFirstBinding binding;  
@Override  
public View onCreateView(LayoutInflater inflater, ViewGroup container,  
Bundle savedInstanceState) {  
binding = FragmentFirstBinding.inflate(inflater, container, false);  
return binding.getRoot();  
}  
public void onViewCreated(@NonNull View, Bundle savedInstanceState) {  
super.onViewCreated(view, savedInstanceState);  
colorSpecViewModel = new ViewModelProvider(requireActivity())  
.get(ColorSpecViewModel.class);  
colorSpecViewModel.getColorList().observe(getViewLifecycleOwner(),  
new Observer<List<ColorSpec>>() {  
@Override  
public void onChanged(List<ColorSpec> colorSpecs) {  
FragColors = colorSpecs;  
colorSpecAdapter = new ColorSpecAdapter(FragColors);  
binding.spinnerColors.setAdapter(colorSpecAdapter);  
}});  
binding.spinnerColors.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {  
@Override  
public void onItemSelected(AdapterView<?> adapterView, View, int i, long l) {  
Toast.makeText(requireActivity(), FragColors.get(i).getColorDesc(), Toast.LENGTH\_SHORT).show();  
 }  
@Override  
public void onNothingSelected(AdapterView<?> adapterView) {  
}});  
binding.buttonFirst.setOnClickListener(new View.OnClickListener() {  
@Override  
public void onClick(View view) {

Bundle = new Bundle();  
bundle.putInt("COLORVAL",FragColors.get(binding.spinnerColors.getSelectedItemPosition()).getColorVal());  
NavHostFragment.findNavController(FirstFragment.this) .navigate(R.id.action\_FirstFragment\_to\_SecondFragment, bundle);  
}}); }  
 @Override  
public void onDestroyView() {  
super.onDestroyView();  
binding = null;  
 }  
 }

**ListViewFragment:FragDemo**

public class ListViewFragment extends Fragment {  
  
 FragmentListViewBinding binding;  
 List<ColorSpec> FragColors;  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 // Inflate the layout for this fragment  
 //return inflater.inflate(R.layout.fragment\_list\_view, container, false);  
binding = FragmentListViewBinding.inflate(inflater,container,false);  
return binding.getRoot();  
}  
@Override  
public void onViewCreated(@NonNull View, @Nullable Bundle savedInstanceState) {  
super.onViewCreated(view, savedInstanceState);  
ColorSpecViewModel = new ViewModelProvider(requireActivity())  
.get(ColorSpecViewModel.class);  
colorSpecViewModel.getColorList().observe(getViewLifecycleOwner(), new Observer<List<ColorSpec>>() {  
@Override  
public void onChanged(List<ColorSpec> colorSpecList) {  
FragColors = colorSpecList;  
binding.listViewColors.setAdapter(new ColorSpecAdapter(FragColors));  
 } });  
binding.listViewColors.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
@Override  
public void onItemClick(AdapterView<?> adapterView, View, int i, long l) {  
Bundle = new Bundle();  
bundle.putInt("COLORVAL", FragColors.get(i).getColorVal());  
//navigate with bundle  
NavHostFragment.findNavController(ListViewFragment.this).navigate(R.id.action\_listViewFragment\_to\_SecondFragment2, bundle);  
} });  
}  
@Override  
public void onDestroyView() {  
super.onDestroyView();  
 } }

**SecondFragment: FragDemo**

public class SecondFragment extends Fragment {  
private FragmentSecondBinding binding;  
@Override  
public View onCreateView(  
LayoutInflater inflater, ViewGroup container,  
Bundle savedInstanceState ) {  
binding = FragmentSecondBinding.inflate(inflater, container, false);  
return binding.getRoot();}  
public void onViewCreated(@NonNull View view, Bundle savedInstanceState) {  
super.onViewCreated(view, savedInstanceState);  
 if (getArguments() != null){  
int colorVal = getArguments().getInt("COLORVAL", Color.BLUE);  
binding.textviewSecond.setTextColor(colorVal);  
}  
binding.buttonSecond.setOnClickListener(new View.OnClickListener() {  
@Override  
public void onClick(View view) {  
NavHostFragment.findNavController(SecondFragment.this).navigate(R.id.action\_SecondFragment\_to\_FirstFragment);  
 } });  
binding.buttonThird.setOnClickListener(new View.OnClickListener() {  
@Override  
public void onClick(View view) {  
NavHostFragment.findNavController(SecondFragment.this).navigate(R.id.action\_SecondFragment\_to\_listViewFragment2);  
} });

}  
  
 @Override  
 public void onDestroyView() {  
 super.onDestroyView();  
 binding = null;  
 }

}

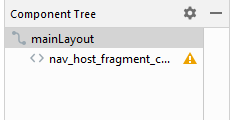
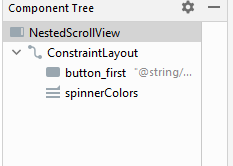
**Layout Image’s**

**Menu layout:**

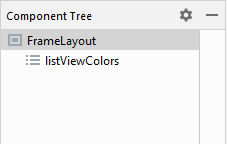
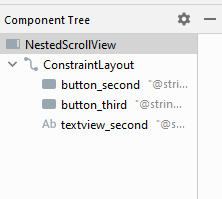
<item  
 android:id="@+id/action\_search"  
 android:orderInCategory="90"  
 android:title="@string/action\_search"  
 android:icon = "@drawable/search"  
 app:showAsAction="always" />  
<item  
 android:id="@+id/action\_settings"  
 android:orderInCategory="100"  
 android:title="@string/action\_settings"  
 android:icon = "@drawable/settings"  
 app:showAsAction="never" />  
<item  
 android:id="@+id/action\_userprofile"  
 android:orderInCategory="90"  
 android:title="@string/action\_userprofile"  
 android:icon = "@drawable/user"  
 app:showAsAction="ifRoom" />

**Navigation Layout:**

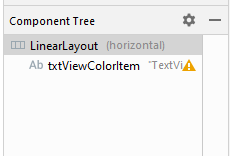
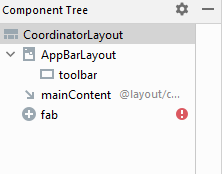
<fragment  
 android:id="@+id/FirstFragment"  
 android:name="com.example.fragdemo.FirstFragment"  
 android:label="@string/first\_fragment\_label"  
 tools:layout="@layout/fragment\_first">  
 <action  
 android:id="@+id/action\_FirstFragment\_to\_SecondFragment"  
 app:destination="@id/SecondFragment" />  
</fragment>  
<fragment  
 android:id="@+id/SecondFragment"  
 android:name="com.example.fragdemo.SecondFragment"  
 android:label="@string/second\_fragment\_label"  
 tools:layout="@layout/fragment\_second">  
 <action  
 android:id="@+id/action\_SecondFragment\_to\_FirstFragment"  
 app:destination="@id/FirstFragment" />  
 <action  
 android:id="@+id/action\_SecondFragment\_to\_listViewFragment2"  
 app:destination="@id/listViewFragment" />  
</fragment>  
<fragment  
 android:id="@+id/listViewFragment"  
 android:name="com.example.fragdemo.ListViewFragment"  
 android:label="fragment\_list\_view"  
 tools:layout="@layout/fragment\_list\_view" >  
 <action  
 android:id="@+id/action\_listViewFragment\_to\_SecondFragment2"  
 app:destination="@id/SecondFragment" />  
</fragment>

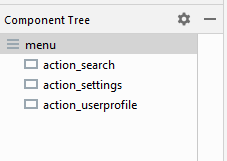
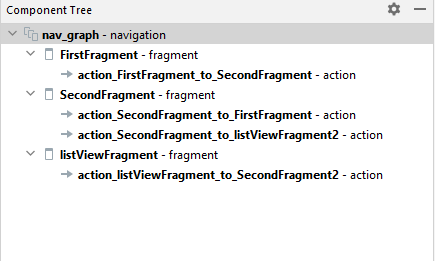
Content\_Main Fragment\_First

Fragment\_List\_View Fragment\_Second

Layout\_Colour\_Item Main\_Activity

Menu\_Item Nav\_graph

**Main Activity : Date Demo**

public class Dog {  
 private int id;  
 private String dogBreed;  
 private String dogName;  
 private int dogPicDrawable; //this is not the resource name but the drawable int value  
private LocalDate dogDob;  
public LocalDate getDogDob() {  
 return dogDob;  
 }  
public void setDogDob(LocalDate dogDob) {  
this.dogDob = dogDob;  
 }  
public int getDogPicDrawable() {  
 return dogPicDrawable;  
 }  
 public void setDogPicDrawable(int dogPicDrawable) {  
 this.dogPicDrawable = dogPicDrawable;  
 }  
 public int getId() {  
 return id;  
 }  
 public void setId(int id) {  
 this.id = id;  
 }  
 public String getDogBreed() {  
 return dogBreed;  
 }  
 public void setDogBreed(String dogBreed) {  
 this.dogBreed = dogBreed;  
 }  
 public String getDogName() {  
 return dogName;  
 }  
 public void setDogName(String dogName) {  
 this.dogName = dogName;  
 }  
 public Dog(int id, String dogBreed, String dogName, int dogPicDrawable) {  
 this.id = id;  
 this.dogName = dogName;  
 this.dogBreed = dogBreed;  
 this.dogPicDrawable = dogPicDrawable;  
 }  
  
public Dog(int id, String dogBreed, String dogName, int dogPicDrawable, LocalDate dogDob) {  
 this.id = id;  
 this.dogBreed = dogBreed;  
 this.dogName = dogName;  
 this.dogPicDrawable = dogPicDrawable;  
 this.dogDob = dogDob;  
 }  
}

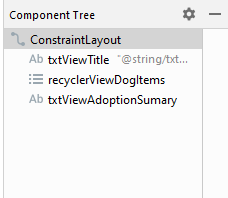
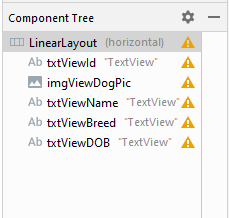
**DogAdapter: Date Demo**

public class DogAdapter extends RecyclerView.Adapter<DogAdapter.DogViewHolder> {  
 List<Dog> AdapterDogList;  
 OnItemClickListener;  
 public DogAdapter(List<Dog> adapterDogList, OnItemClickListener onItemClickListener) {  
 AdapterDogList = adapterDogList;  
 this.onItemClickListener = onItemClickListener;  
 }  
 public DogAdapter(List<Dog> adapterDogList) {  
 AdapterDogList = adapterDogList;  
 }  
 @NonNull  
 @Override  
 public DogViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 //return null;  
 //Inflate the binding  
 //using binding create holder object  
 //return holder object  
 LayoutDogitemBinding binding = LayoutDogitemBinding  
 .inflate(LayoutInflater  
 .from(parent.getContext()),  
 parent, false);  
 DogViewHolder holder = new DogViewHolder(binding.getRoot(), binding);  
 return holder;  
 }  
 @RequiresApi(api = Build.VERSION\_CODES.O)  
 @Override  
 public void onBindViewHolder(@NonNull DogViewHolder holder, int position) {  
 holder.binding.txtViewId  
 .setText(Integer.toString(AdapterDogList.get(position).getId()));  
 holder.binding.txtViewName.setText(AdapterDogList.get(position).getDogName()); holder.binding.txtViewBreed.setText(AdapterDogList.get(position).getDogBreed());  
 holder.binding.imgViewDogPic  
 .setImageResource(AdapterDogList.get(position).getDogPicDrawable());  
 DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd MM yyyy");  
 String dobStr = formatter.format(AdapterDogList.get(position).getDogDob());  
 holder.binding.txtViewDOB.setText(dobStr);  
 }  
 @Override  
 public int getItemCount() {  
  
 return AdapterDogList.size();  
 }  
  
 public class DogViewHolder extends RecyclerView.ViewHolder {  
 LayoutDogitemBinding binding;  
  
 public DogViewHolder(@NonNull View itemView) {  
 super(itemView);  
 }  
  
 public DogViewHolder(@NonNull View itemView, LayoutDogitemBinding binding) {  
 super(itemView);  
 this.binding = binding;  
 this.binding.getRoot()  
 .setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 onItemClickListener.onItemClick(getAdapterPosition());  
 }  
 });  
 }  
 }  
 public interface OnItemClickListener {  
 public void onItemClick(int i);  
 }  
}

**MainActivity: Date Demo**

public class MainActivity extends AppCompatActivity {  
 List<Dog> DogList;  
 @RequiresApi(api = Build.VERSION\_CODES.O)  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 //setContentView(R.layout.activity\_main);  
 ActivityMainBinding binding = ActivityMainBinding.inflate(getLayoutInflater());  
 setContentView(binding.getRoot());  
 ReadDogData();  
 Log.d("DATEDEMO",DogList.size() + " dogs in the list");  
 // DogAdapter dogAdapter = new DogAdapter(DogList);  
 DogAdapter dogAdapter = new DogAdapter(DogList,  
 new DogAdapter.OnItemClickListener() {  
 @Override  
 public void onItemClick(int i) {  
 binding.txtViewAdoptionSumary  
 .setText("Thank you for taking "  
 + DogList.get(i).getDogName() +  
 " to their forever home");  
 }  
 });  
 binding.recyclerViewDogItems.setAdapter(dogAdapter);  
 binding.recyclerViewDogItems  
 .setLayoutManager(new LinearLayoutManager(this));  
  
 }  
 @RequiresApi(api = Build.VERSION\_CODES.O)  
 private void ReadDogData(){  
 DogList = new ArrayList<>();  
 InputStream inputStream  
 = getResources().openRawResource(R.raw.doginfo);  
 BufferedReader reader  
 = new BufferedReader(new InputStreamReader(inputStream));  
 try {  
 String csvLine;  
 //no header in input file  
 //if header line is present use if to read it out  
 /\*if ((csvLine = reader.readLine()) != null){  
 //csvLine contains header  
 }\*/  
 //data lines  
 while((csvLine = reader.readLine()) != null){  
 String[] eachDogFields = csvLine.split(",");  
 int dogId = Integer.parseInt(eachDogFields[0]);  
 String resourceName = eachDogFields[1];  
 String breedName = eachDogFields[2];  
 String dogName = eachDogFields[3];  
 String dogDobStr = eachDogFields[4];  
 int dogDrawable = getResources()  
 .getIdentifier(resourceName,"drawable",getPackageName());  
  
 //d - one or more digits for date e.g., 8, 16, 31  
 //dd - two digits for date e.g., 08, 16, 31  
 //MMM - three letter month code e.g., SEP, OCT, MAY  
 //MM - two digit month number e.g., 01, 12, 10  
 //yyyy - four digit year number e.g., 1638, 2023, 1949  
 //yy - two digit year number e.g., 38, 23, 49  
  
 DateTimeFormatter formatter =  
 DateTimeFormatter.ofPattern("d-MMM-yyyy");  
 LocalDate dob = LocalDate.parse(dogDobStr,formatter);  
  
 Dog eachDog = new Dog(dogId,breedName,dogName,dogDrawable,dob);  
 DogList.add(eachDog);  
 }  
 } catch (Exception ex){  
 ex.printStackTrace();  
 }  
 }  
  
}

**Layout**

** **

Main Activity Layout\_DogItem

How to put recycler view inside the Fragment

**Create a RecyclerView in XML layout:**

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".YourFragment">

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/recyclerView"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:padding="16dp"

android:clipToPadding="false" />

</RelativeLayout>

**Fragment Class**

public class YourFragment extends Fragment {

private RecyclerView recyclerView;

private YourAdapter adapter;

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

View view = inflater.inflate(R.layout.fragment\_your, container, false);

recyclerView = view.findViewById(R.id.recyclerView);

recyclerView.setLayoutManager(new LinearLayoutManager(getContext()));

adapter = new YourAdapter(/\* pass data if needed \*/);

recyclerView.setAdapter(adapter);

return view;

}

}

**Adapter class**

public class YourAdapter extends RecyclerView.Adapter<YourViewHolder> {

// Your data goes here

@NonNull

@Override

public YourViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

View = LayoutInflater.from(parent.getContext()).inflate(R.layout.your\_item\_layout, parent, false); return new YourViewHolder(view); }

@Override

public void onBindViewHolder(@NonNull YourViewHolder holder, int position) {

// Bind your data to the ViewHolder here

}

@Override

public int getItemCount() {

// Return the size of your data return /\* size \*/;

}}

**ViewHolder Class**

public class YourViewHolder extends RecyclerView.ViewHolder {

// Declare your views here

public YourViewHolder(@NonNull View itemView) {

super(itemView);

// Initialize your views here

}

}