There are 2 examples in project,

so at first you get only text and at second you also add image

In grade scripts add to the dependencies in build.grade module: app:

implementation 'com.android.support:recyclerview-v7:28.0.0'

Then as in file example (code). In main xml write:

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.ConstraintLayout initialization… ">  
<android.support.v7.widget.RecyclerView - important type of block  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:listitem="@android:layout/list\_content"  
 android:id="@+id/recycleview"  
 android:scrollbars="vertical"> - add scrollbar   
  
</android.support.v7.widget.RecyclerView>  
</android.support.constraint.ConstraintLayout>

Then create new fragment of type that you need for example TextView without any other heading tags, block is main:

<?xml version="1.0" encoding="utf-8"?>  
<TextView initialization… > </TextView>

<ImageView … />

Then create class which will be ruling this data and extend it from RecyclerView.Adapter<your inside interface>

Other as in file, create list of your variables and inside class, which extends RecyclerView.ViewHolder. In this class create variable of type of your main block in xml and initialize it in constructor (change common parameter on the TextView for example)

Then fill functions as in example and in main as well as in example

Adapter.notifyDataSetChanged(); - update data

Error with changing each 12 element

@Override  
public long getItemId(int position) {  
 return position;  
}  
  
@Override  
public int getItemViewType(int position) {  
 return position;  
}

Get needed view

View view = recyclerView.findViewHolderForAdapterPosition(getAdapterPosition()).itemView;

Creating filter for search

Implement Filterable

@Override  
public Filter getFilter() {  
 return new Filter() {  
 @Override  
 protected FilterResults performFiltering(CharSequence charSequence) {  
 String charString = charSequence.toString(); - get sting from method  
 if (charString.isEmpty()) { - if it is empty display all  
 arrayListFiltered = arrayList;  
 } else {  
 ArrayList<String> filteredList = new ArrayList<>(); - create new array  
 for (String row : arrayList) { - run trough all variables  
  
 if (row.toLowerCase().contains(charString.toLowerCase()) || row.contains(charSequence)) { - if row equals to entered string  
 filteredList.add(row);

}  
 }  
  
 arrayListFiltered = filteredList; - set new filter  
 }  
  
 FilterResults filterResults = new FilterResults();  
 filterResults.values = arrayListFiltered;  
 return filterResults;  
 }  
  
 @Override  
 protected void publishResults(CharSequence charSequence, FilterResults filterResults) {  
 arrayListFiltered = (ArrayList<String>) filterResults.values;  
  
 // refresh the list with filtered data  
 notifyDataSetChanged(); - update data  
 }  
 };  
}

Then use in search view

searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {  
 @Override  
 public boolean onQueryTextSubmit(String query) {  
 // filter recycler view when query submitted  
 adapter.getFilter().filter(query);  
 return false;  
 }  
  
 @Override  
 public boolean onQueryTextChange(String query) {  
 // filter recycler view when text is changed  
 adapter.getFilter().filter(query);  
 return false;  
 }  
});