



实现水平滚动和瀑布流布局

1. 实训目的

- (1) 掌握适配器 ArrayAdapter 的用法;
- (2) 掌握在 build.gradle 中添加依赖的方法;
- (3) 掌握使用 RecyclerView 实现滚动布局和瀑布流布局的方法。

2. 实训要求

- (1) 创建项目并选择合适的布局类型;
- (2) 创建布局和自定义子布局、创建实体类、自定义适配器;
- (3) 完成实验报告。

3. 实训指导

任务一 RecyclerView 基本用法

- (1) 在 build.gradle 中添加依赖

由于该控件并不在 Andorid SDK 中的, 而是在 support v7 包中, 因此我们要手动添加该控件。在 app/build.gradle 文件中添加如下依赖:

```
dependencies {  
    ...  
    compile 'com.android.support:appcompat-v7:23.1.1' //Toolbar  
    compile 'com.android.support:recyclerview-v7:23.1.1' //RecyclerView  
}
```

- (2) Fruit 类和 fruit_item.xml 和 ListView 中的例子一样。

- (3) 创建 FruitAdapter 类

FruitAdapter.java

```
public class FruitAdapter extends RecyclerView.Adapter  
<FruitAdapter.ViewHolder>{  
    private List<Fruit> mFruitList;  
    static class ViewHolder extends RecyclerView.ViewHolder{  
        ImageView fruitImage;  
        TextView fruitName;  
        public ViewHolder(View itemView) {
```



```
        super(itemView);

        fruitImage=(ImageView)
itemView.findViewById(R.id. fruit_image);

        fruitName=(TextView)
itemView.findViewById(R.id. fruit_name);
    }
}

public FruitAdapter(List<Fruit> fruitList) {
    mFruitList=fruitList;
}

@Override

    public ViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
        View view=
LayoutInflater.from(parent.getContext()).inflate(R.layout. fruit_item,
parent, false);

        RecyclerView.ViewHolder holder=new ViewHolder(view);
        return (ViewHolder) holder;
    }

@Override

    public void onBindViewHolder(ViewHolder holder, int position) {
        Fruit fruit=mFruitList.get(position);

        holder.fruitImage.setImageResource(fruit.getImageId());
        holder.fruitName.setText(fruit.getName());
    }

@Override

    public int getItemCount() {
        return mFruitList.size();
    }
}
```



```
}  
}
```

(4) 修改 MainActivity.java 代码

MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
    private List<Fruit> fruitList=new ArrayList<>();  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        initFruits();  
        RecyclerView  
recyclerView=(RecyclerView)findViewById(R.id.recycler_view);  
        LinearLayoutManager layoutManager=new  
LinearLayoutManager(this);  
        recyclerView.setLayoutManager(layoutManager);  
        FruitAdapter adapter=new FruitAdapter(fruitList);  
        recyclerView.setAdapter(adapter);  
    }  
    private void initFruits() {  
        for(int i=0;i<2;i++) {  
            Fruit banana=new Fruit("Banana",R.drawable.banana1);  
            fruitList.add(banana);  
            Fruit cherry=new Fruit("Cherry",R.drawable.cherry1);  
            fruitList.add(cherry);  
            Fruit grape=new Fruit("Grape",R.drawable.grape1);  
            fruitList.add(grape);  
            Fruit mango=new Fruit("Mango",R.drawable.mango1);  
            fruitList.add(mango);  
        }  
    }  
}
```



```
Fruit orange=new Fruit("Orange",R.drawable. orange1);  
fruitList.add(orange);  
Fruit pear=new Fruit("Pear",R.drawable. pear1);  
fruitList.add(pear);  
Fruit pineapple=new  
Fruit("Pineapple",R.drawable. pineapple1);  
fruitList.add(pineapple);  
Fruit tomato=new Fruit("Tomato",R.drawable. tomato1);  
fruitList.add(tomato);  
}  
}
```

任务二 实现横向滚动和瀑布流布局

(1) 实现横向滚动效果主要代码:

```
LinearLayoutManager layoutManager=new LinearLayoutManager(this);  
layoutManager.setOrientation(LinearLayoutManager.HORIZONTAL);  
recyclerView.setLayoutManager(layoutManager);
```

(2) 瀑布流布局效果主要代码:

```
StaggeredGridLayoutManager layoutManager = new  
StaggeredGridLayoutManager(3, StaggeredGridLayoutManager.VERTICAL);
```

任务三 RecyclerView 的点击事件

修改 FruitAdapter.java 代码

```
static class ViewHolder extends RecyclerView.ViewHolder {  
    View fruitView;//点击事件新加  
    ImageView fruitImage;  
    TextView fruitName;  
    public ViewHolder(View itemView) {  
        super(itemView);  
        fruitView=itemView;//点击事件新加  
        fruitImage=(ImageView)
```



```
itemView.findViewById(R.id.fruit_image);
        fruitName=(TextView) itemView.findViewById(R.id.fruit_name);
    }
}

public FruitAdapter(List<Fruit> fruitList) {
    mFruitList=fruitList;
}

@Override
public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType)
{
    View view=
    LayoutInflater.from(parent.getContext()).inflate(R.layout.fruit_item,
    parent,false);

    final ViewHolder holder=new ViewHolder(view);
    holder.fruitView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            int position=holder.getAdapterPosition();
            Fruit fruit=mFruitList.get(position);
            Toast.makeText(view.getContext(), "you clicked
            view"+fruit.getName(), Toast.LENGTH_LONG).show();
        }
    });
    holder.fruitImage.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            int position=holder.getAdapterPosition();
            Fruit fruit=mFruitList.get(position);
            Toast.makeText(view.getContext(), "you clicked
```

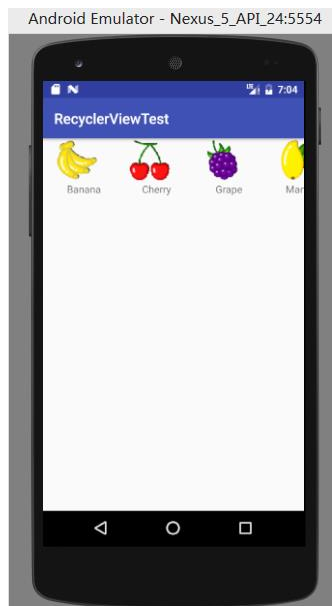
```
view"+fruit.getName(), Toast.LENGTH_LONG).show();

    }

});

return holder;

}
```



例 2 效果图



例 3 效果图



例 4 效果图