RecyclerView 的使用

> *顾名思义，RecyclerView 所做的事情就是回收再利用，循环往复。*
> *RecyclerView 套餐中，Adapter 从模型层获取数据，创建 ViewHolder，提供给 RecyclerView 显示。*
> *ViewHolder 懒惰成性，为 itemView 而生，只做一件事：容纳 View 视图*

1.为 app 添加依赖

在 Add Library Dependency 界面的搜索栏中，键入 **androidx.recyclerview**

2.RecyclerView.ViewHolder 类

```
// 在此处实现点击接口，即可实现 recyclerView 中 item 的点击事件
private class CrimeHolder extends RecyclerView.ViewHolder implements View.OnClickLi
stener{
    private TextView mTitleTextView;
    private TextView mDateTextView;
    private ImageView mSolvedImageView;
    private Crime mCrime;

    // 自定义构造方法，注意参数
    public CrimeHolder(LayoutInflater inflater, ViewGroup parent) {
        super(inflater.inflate(R.layout.list_item_crime, parent, false));

        // 设置点击监听器
        itemView.setOnClickListener(this);
        mTitleTextView = itemView.findViewById(R.id.crime_title);
        mDateTextView = itemView.findViewById(R.id.crime_date);
        mSolvedImageView = itemView.findViewById(R.id.crime_solved);
    }

    // 绑定方法，实现视图绑定
    // 任何时候，只要有可能，都要确保这个方法轻巧、高效
    public void bind(Crime crime) {
        mCrime = crime;
        mTitleTextView.setText(crime.getTitle());
        mDateTextView.setText(crime.getDate().toString());
        mSolvedImageView.setVisibility(crime.isSolved() ? View.VISIBLE : View.GONE)
;
    }
```

```
    @Override
    public void onClick(View v) {
        Toast.makeText(getActivity(), mCrime.getTitle() + " clicked", Toast.LENGTH_
SHORT).show();
        Intent intent = CrimePagerActivity.newIntent(getActivity(), mCrime.getId())
;
        startActivity(intent);
    }
}
```

3.RecyclerView.Adapter 类

```
private class CrimeAdapter extends RecyclerView.Adapter<CrimeHolder> {
    private List<Crime> mCrimes;

    public CrimeAdapter(List<Crime> crimes) {
        mCrimes = crimes;
    }

    @NonNull
    @Override
    // 此方法的调用并不频繁，一旦有了足够的 ViewHolder，RecyclerView 便会停止调用此方法，转而
回收利用旧的 ViewHolder 以节约时间和内存
    public CrimeHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType)
{
        LayoutInflater layoutInflater = LayoutInflater.from(getActivity());
        return new CrimeHolder(layoutInflater, parent);
    }

    @Override
    public void onBindViewHolder(@NonNull CrimeHolder holder, int position) {
        Crime crime = mCrimes.get(position);
        holder.bind(crime);
    }

    @Override
    public int getItemCount() {
        return mCrimes.size();
    }
}
```

4.初始化 recyclerview 数据、配置布局管理器、适配器

```
// 配置布局管理器
mCrimeRecyclerView = view.findViewById(R.id.crime_recycler_view);
mCrimeRecyclerView.setLayoutManager(new LinearLayoutManager(getActivity()));
```

```
// 模型层的数据若有变化，应通知 RecyclerView 的 Adapter
// 以便其获取最新数据并刷新显示列表项
// 一般来说，要保证 fragment 视图得到刷新，
// 在 onResume() 方法内更新代码是最安全的选择
if (mAdapter == null) {
    mAdapter = new CrimeAdapter(crimes);
    mCrimeRecyclerView.setAdapter(mAdapter);
} else {
    mAdapter.notifyDataSetChanged();
}
```