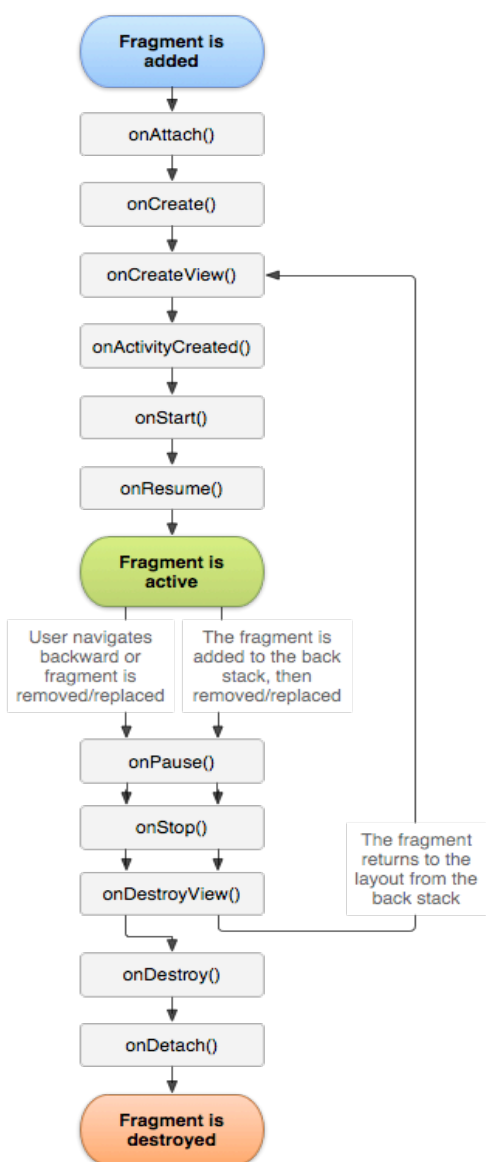


## Fragment 生命周期

### 知识解析

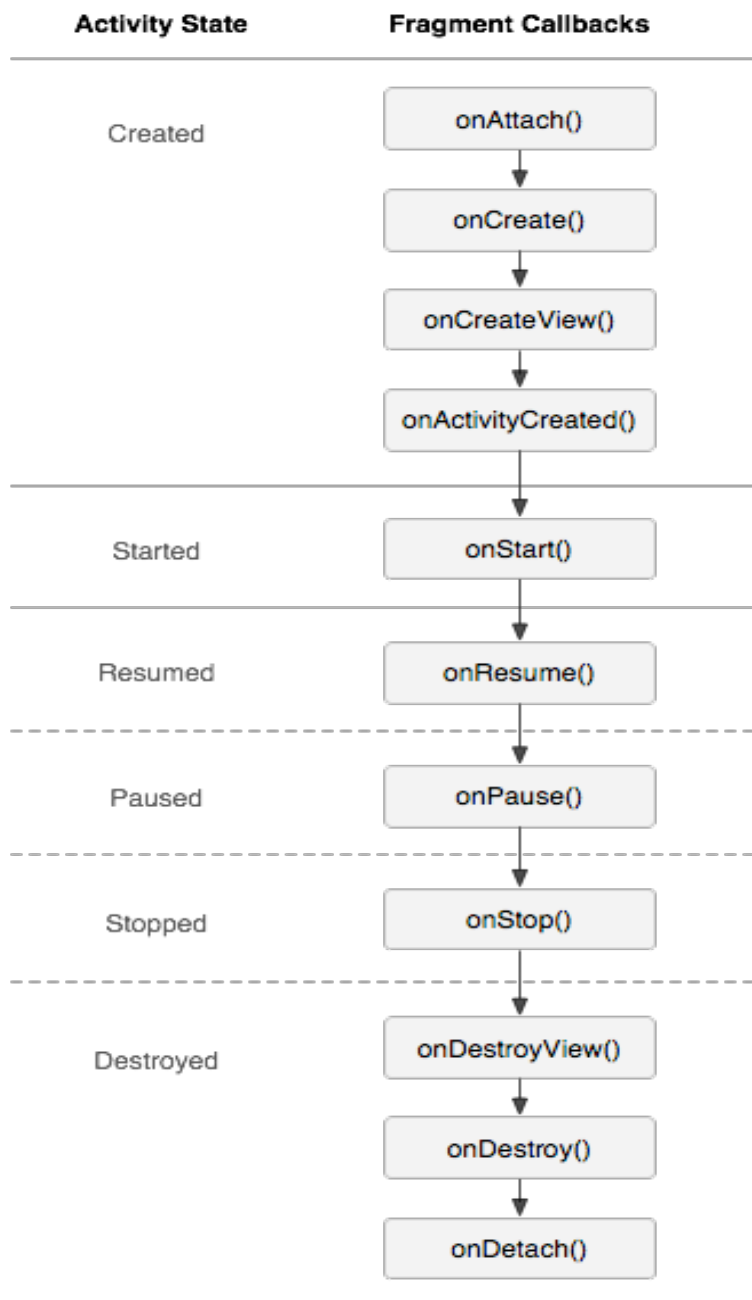
Fragment 的生命周期可以通过下面的图来了解。



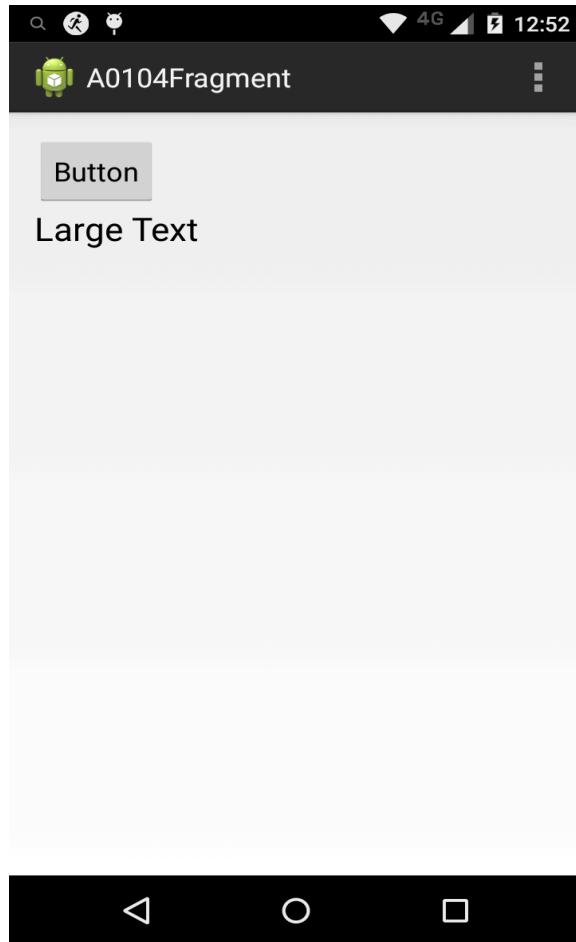
Fragment 的生命周期方法的用途如下表所述。

方法	描述
onAttach(Activity)	当前 Fragment 与 Activity 关联时调用
onCreate()	完成 fragment 的初始化创建
onCreateView()	创建并返回与当前 fragment 相关联的层次视图 view
onActivityCreated()	主 activity 的 onCreate() 执行完后，该方法才执行
onStart()	fragment 可见，当主 activity 处于 started 状态后执行
onResume()	fragment 能与用户交互，当主 activity 处于 resumed 状态后执行
onPause()	fragment 不再与用户交互，可能在宿主 activity 将要处于 paused 前执行，可能该 fragment 被修改
onStop()	fragment 不再可见，可能在宿主 activity 将要处于 stopped 前执行，可能该 fragment 被修改
onDestroyView()	允许该 fragment 清理视图相关资源
onDestroy()	清理掉视图 state 信息
onDetach()	该 fragment 不再与 activity 关联

它的生命周期方法和宿主 activity 状态之间的关系可以参考下图：



## 功能演示



## 实战操作

```
public class MyFragment extends Fragment {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        Log.w("Fragment", "onCreate");
        super.onCreate(savedInstanceState);
    }
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
```

```
container, Bundle savedInstanceState) {  
    Log.w("Fragment", "onCreateView");  
    View view = inflater.inflate(R.layout.fragment, container);  
    return view;  
}  
@Override  
public void onActivityCreated(Bundle savedInstanceState) {  
    Log.w("Fragment", "onActivityCreated");  
    super.onActivityCreated(savedInstanceState);  
}  
@Override  
public void onAttach(Activity context) {  
    Log.w("Fragment", "onAttach");  
    super.onAttach(context);  
}  
@Override  
public void onDestroy() {  
    Log.w("Fragment", "onDestroy");  
    super.onDestroy();  
}  
@Override  
public void onDestroyView() {  
    Log.w("Fragment", "onDestroyView");  
    super.onDestroyView();  
}  
@Override  
public void onDetach() {  
    Log.w("Fragment", "onDetach");  
    super.onDetach();  
}  
@Override  
public void onPause() {  
    Log.w("Fragment", "onPause");  
    super.onPause();  
}  
@Override  
public void onResume() {
```

```
        Log.w("Fragment", "onResume");
        super.onResume();
    }
    @Override
    public void onStart() {
        Log.w("Fragment", "onStart");
        super.onStart();
    }
    @Override
    public void onStop() {
        Log.w("Fragment", "onStop");
        super.onStop();
    }
}
```

## 职业素质

每一个 fragments 都有自己的一套生命周期回调方法和处理自己的用户输入事件。想要更好的使用 **Fragment**，熟练掌握 **Fragment** 的生命周期是必不可少的。在不同的生命周期里，我们可以做不同的事情，处理不同的应用程序状态。