

帧布局实现切换界面效果

1 在需要实现界面切换效果的地方添加布局代码

```
<FrameLayout
    android:layout_weight="1"
    android:id="@+id/fl_all_content"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
</FrameLayout>
```

2 在java代码中 添加Fragment管理器。

```
fragmentManager = getFragmentManager();
//开启事务
FragmentTransaction mTransaction = fragmentManager.beginTransaction();
maininterfaceActivity = new MsgfaceFragment();
mTransaction.replace(R.id.fl_all_content, maininterfaceActivity).commit();
```

并设置不同的点击效果，实现进入不同的界面

```
@Override
public void onClick(View v) {
    FragmentTransaction ft = fragmentManager.beginTransaction();
    switch (v.getId()) {
        case R.id.it_msg:
            ft.replace(R.id.fl_content, maininterfaceActivity);
            break;
        case R.id.it_lxr:
            break;
        case R.id.it_dt:
            break;
    }
}
```

3 让显示的Activity 继承 Fragment 父类

```
public class MsgfaceFragment extends Fragment
```

在onCreateView 方法中实现对需要显示的布局的映射

```
public View onCreateView(
    LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {

    v = inflater.inflate(R.layout.fragment_msg_face,null);
    initUI();
    return v;
}

private void initUI() {
    tv_msg = (TextView) v.findViewById(R.id.tv_msg);
    tv_phone = (TextView) v.findViewById(R.id.tv_phone);
    msgFragment = new ItemMsgFragment();
    phoneFragment = new ItemPhoneFragment();

    tv_msg.setOnClickListener(this);
    tv_phone.setOnClickListener(this);

    //获取到fragment的管理者
    fragmentManager = getFragmentManager();
    //开启事务
```

```
FragmentTransaction mTransaction = fragmentManager.beginTransaction();
//将消息界面代替初始界面
mTransaction.replace(R.id.fl_content, msgFragment).commit();
}
```