

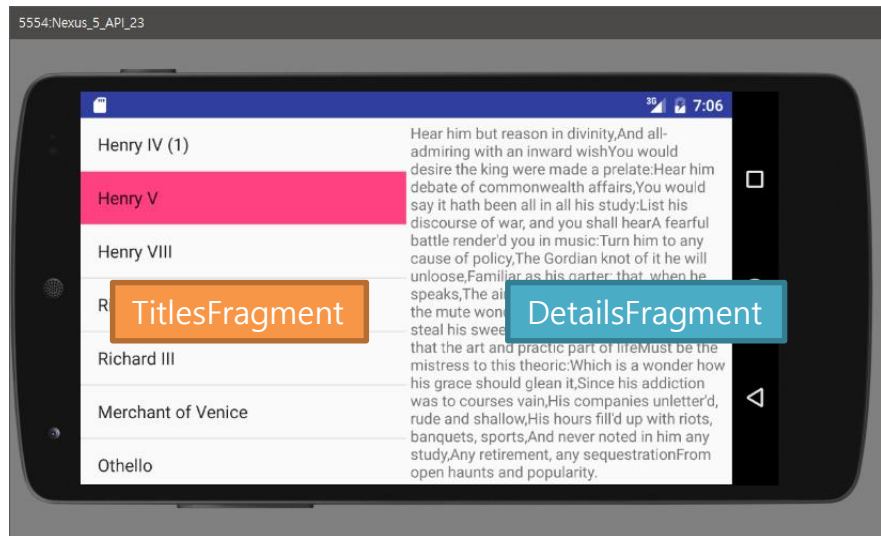
04

화면 방향과 프래그먼트 활용

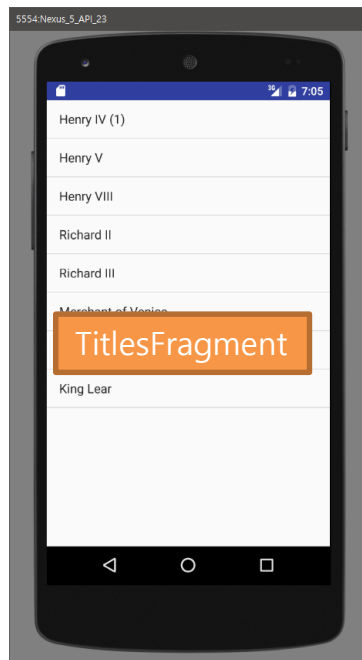
3 화면 방향과 프래그먼트 활용



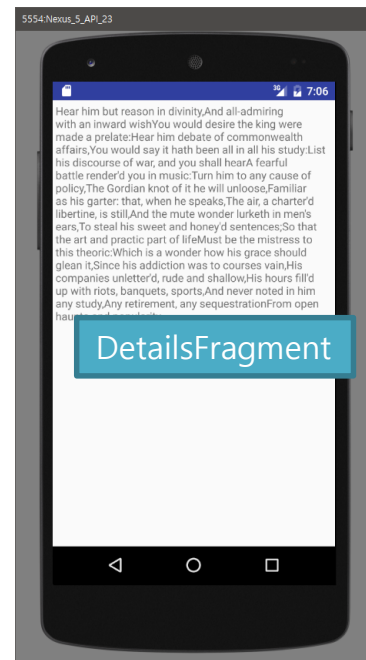
가. TitlesFragment 와 DetailsFragment의 재사용



MainActivity



MainActivity



DetailActivity

예제 프로젝트 파일:

<https://github.com/kwanulee/Android/tree/master/examples/FragmentExample>



나. 액티비티 레이아웃

- ◆ 액티비티에 가로보기(Landscape) 레이아웃을 추가
 - res/layout 에서 마우스 오른쪽 클릭
 - New > Layout resource file
 - Qualifiers에서 Orientation - Landscape 선택
 - 파일 이름은 activity_main.xml

주의! 화면 방향을 바꾸면 액티비티가 새로 만들어짐.
당연히 프래그먼트도 다시 만들어짐



나. 액티비티 레이아웃

◆ Landscape 용 레이아웃 layout-land/activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <fragment
        android:name="com.example.kwanwoo.fragmentexample.TitlesFragment"
        android:id="@+id/titles"
        android:layout_weight="1"
        android:layout_width="0px"
        android:layout_height="match_parent" />
    <FrameLayout
        android:id="@+id/details"
        android:layout_weight="1"
        android:layout_width="0px"
        android:layout_height="match_parent" />
</LinearLayout>
```

<- DetailsFragment를 동적으로 추가할 위치



나. 액티비티 레이아웃

◆ Portrait 용 레이아웃 layout/activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

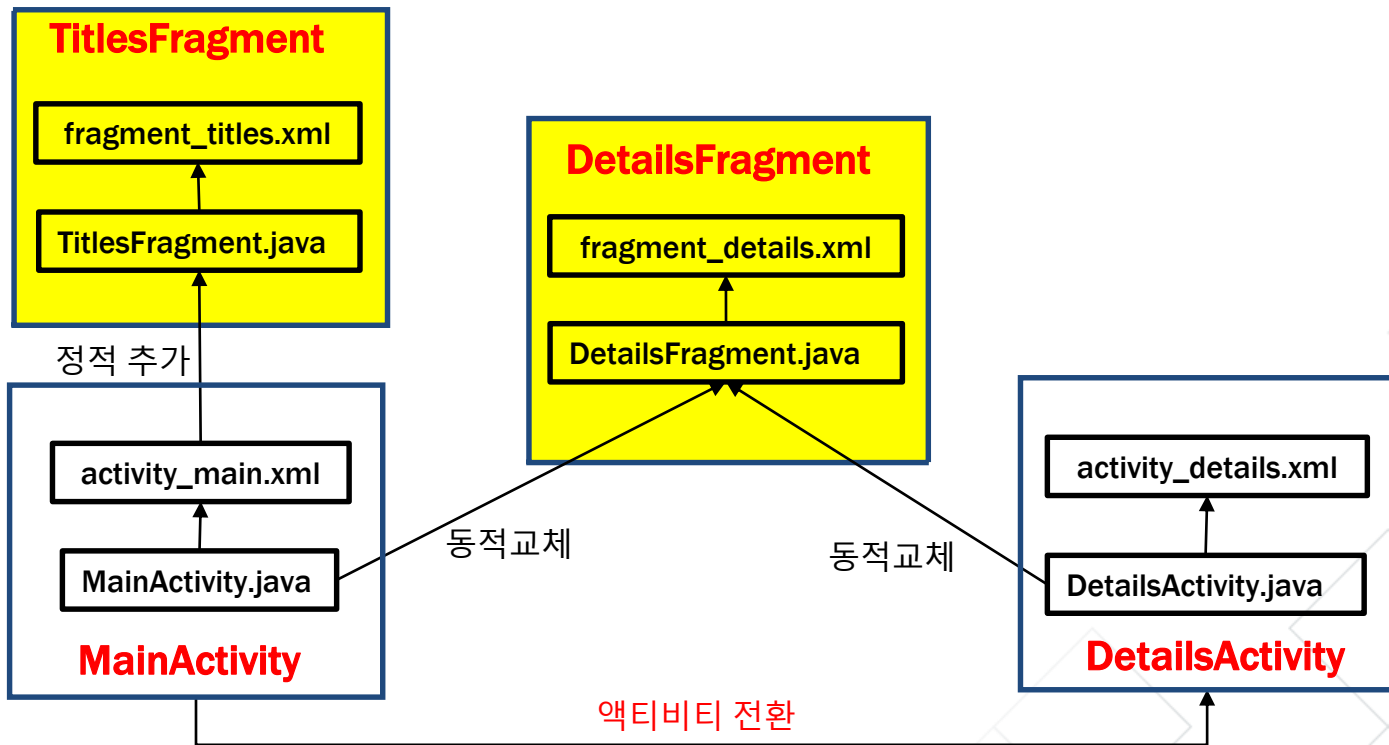
    <fragment
        android:name="com.example.kwanwoo.fragmentexample.TitlesFragment"
        android:id="@+id/titles"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```

3 화면 방향과 프래그먼트 활용



다. 애플리케이션 구조





라. TitlesFragment 클래스

```
public class TitlesFragment extends Fragment {  
    ...  
    public interface OnTitleSelectedListener { // 액티비티로 전달할 메시지 인터페이스  
        public void onTitleSelected(int i);  
    }  
  
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
        View rootView = (View)inflater.inflate(R.layout.fragment_titles, container, false);  
  
        ListView lv = (ListView)rootView.findViewById(R.id.listview);  
  
        ...  
        lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
            @Override // 리스트에서 항목을 선택했을 때 호출  
            public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {  
                ...  
                ((OnTitleSelectedListener)getActivity()).onTitleSelected(i);  
            }  
        });  
        lv.setChoiceMode(ListView.CHOICE_MODE_SINGLE);  
        return rootView;  
    }  
}
```

TitlesFragment와
연결된 액티비티



마. MainActivity 클래스

```
public class MainActivity extends AppCompatActivity implements TitlesFragment.OnTitleSelectedListener {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void onTitleSelected(int i) {
        if (getResources().getConfiguration().orientation
            == Configuration.ORIENTATION_LANDSCAPE) { // 가로보기용 레이아웃 사용중인 경우
            // DetailsFragment를 선택한 항목에 맞는 것으로 바꿈
            DetailsFragment detailsFragment = new DetailsFragment();
            detailsFragment.setSelection(i);
            getSupportFragmentManager().beginTransaction().replace(R.id.details, detailsFragment).commit();
        } else { // 세로보기용 레이아웃
            Intent intent = new Intent(this, DetailsActivity.class);
            intent.putExtra("index", i);
            startActivity(intent); // DetailActivity 를 시작함
        }
    }
}
```




바. DetailsActivity 클래스

```
public class DetailsActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_details);

        if (getResources().getConfiguration().orientation
            == Configuration.ORIENTATION_LANDSCAPE) {
            finish();
            return;
        }

        DetailsFragment details = new DetailsFragment();

        // MainActivity에서 인텐트로 넘겨 받은 리스트 뷰 항목 선택 번호를 DetailsFragment 객체로 전달
        details.setSelection(getIntent().getIntExtra("index", -1));

        // details를 기존 프래그먼트와 교체
        getSupportFragmentManager().beginTransaction().replace(R.id.details, details).commit();
    }
}
```



사. DetailsFragment 클래스

```
public class DetailsFragment extends Fragment {  
  
    int index=-1;  
  
    public DetailsFragment() {  
        // Required empty public constructor  
    }  
  
    public void setSelection(int i) { index = i; }  
  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)  
    {  
        // Inflate the layout for this fragment  
        View view = inflater.inflate(R.layout.fragment_details, container, false);  
        TextView tv = (TextView)view.findViewById(R.id.textview);  
  
        if (index >= 0)  
            tv.setText(Shakespeare.DIALOGUE[index]);  
  
        return view;  
    }  
}
```



아. TitlesFragment 클래스 (Revisited)

- ◆ 프래그먼트가 다시 시작될 때 저장된 선택 항목의 위치 설정

```
public class TitlesFragment extends Fragment {
    int mCurCheckPosition = -1;    // 선택한 항목의 위치
    ...

    @Override
    public void onViewStateRestored(Bundle savedInstanceState) {
        ...
        if (savedInstanceState != null) { // 프래그먼트가 다시 시작될 때 저장된 선택 항목의 위치를 가져옴
            mCurCheckPosition = savedInstanceState.getInt("curChoice", -1);
            if (mCurCheckPosition >= 0) // 저장된 위치가 0이상이면 onTitleSelected 메소드 호출
                ((OnTitleSelectedListener)getActivity()).onTitleSelected(mCurCheckPosition);
            ...
        }
    }

    @Override
    public void onSaveInstanceState(Bundle outState) { // 프래그먼트가 사라질 때 선택 항목의 위치를 저장
        super.onSaveInstanceState(outState);
        outState.putInt("curChoice", mCurCheckPosition);
    }
}
```