



# 任务四 手动更新天气

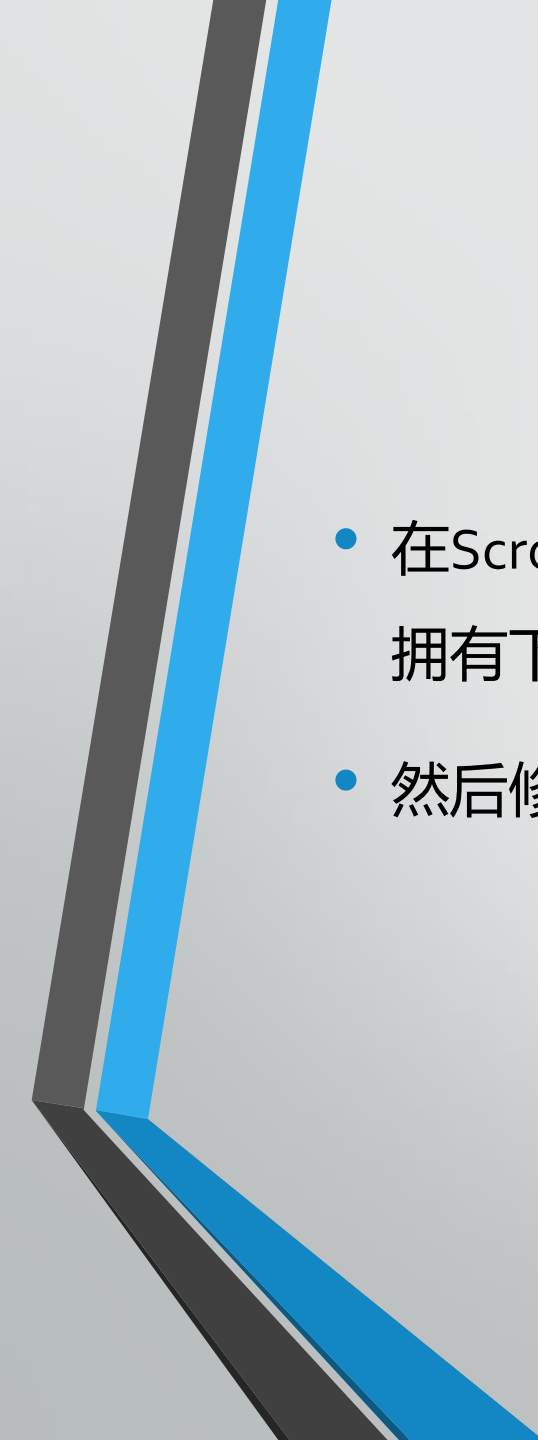
主讲教师：张晓诺

- 虽然酷欧天气主体功能完成了，但是还存在一个比较严重的bug，当选中某个城市后，就没法再去查看其它城市的天气了，即使退出程序下次进来时还是直接跳转到WeatherActivity。

- 前面我们实现了对天气信息进行缓存，每次展示的是缓存中的数据，因此需要能够让用户手动更新天气信息。采用下拉刷新的方式来实现手动更新天气信息。
- 修改activity\_weather.xml中的代码，如下所示：

- `<FrameLayout`  
    `xmlns:android="http://schemas.android.com/apk/res/android"`  
    `android:layout_width="match_parent"`  
    `android:layout_height="match_parent"`  
    `android:background="@color/colorPrimary">`  
    .....  
    `<android.support.v4.widget.DrawerLayout`  
        `android:id="@+id/drawer_layout"`  
        `android:layout_width="match_parent"`  
        `android:layout_height="match_parent">`  
  
    `<android.support.v4.widget.SwipeRefreshLayout`  
        `android:id="@+id/swipe_refresh"`  
        `android:layout_width="match_parent"`  
        `android:layout_height="match_parent">`

- .....

- 
- 在ScrollView的外层嵌套一层SwipeRefreshLayout，这样ScrollView就可以自动拥有下拉刷新功能。
  - 然后修改WeatherActivity中的代码，加入更新天气逻辑，如下所示：

- `public class WeatherActivity extends AppCompatActivity {`
- `.....`
- `public SwipeRefreshLayout swipeRefresh;`
- `.....`
- `@Override`  
`protected void onCreate(Bundle savedInstanceState) {`
- `swipeRefresh = (SwipeRefreshLayout) findViewById(R.id.swipe_refresh);`  
`swipeRefresh.setColorSchemeResources(R.color.colorPrimary);`  
`drawerLayout = (DrawerLayout) findViewById(R.id.drawer_layout);`  
`navButton = (Button) findViewById(R.id.nav_button);`  
`SharedPreferences prefs = PreferenceManager.getDefaultSharedPreferences(this);`  
`String weatherString = prefs.getString("weather", null);`  
`if (weatherString != null) {`  
    // 有缓存时直接解析天气数据  
`Weather weather = Utility.handleWeatherResponse(weatherString);`  
`mWeatherId = weather.basic.weatherId;`  
`showWeatherInfo(weather);`  
`} else {`  
    // 无缓存时去服务器查询天气  
`mWeatherId = getIntent().getStringExtra("weather_id");`  
`weatherLayout.setVisibility(View.INVISIBLE);`  
`requestWeather(mWeatherId);`  
`}`
- `swipeRefresh.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener() {`  
    `@Override`  
    `public void onRefresh() {`  
        `requestWeather(mWeatherId);`  
    `}`  
`});`

- ```
/**
 * 根据天气id请求城市天气信息。
 */
public void requestWeather(final String weatherId) {
    String weatherUrl = "http://guolin.tech/api/weather?cityid=" + weatherId +
"&key=bc0418b57b2d4918819d3974ac1285d9";
    HttpUtil.sendOkHttpRequest(weatherUrl, new Callback() {
        @Override
        public void onResponse(Call call, Response response) throws IOException {
            final String responseText = response.body().string();
            final Weather weather = Utility.handleWeatherResponse(responseText);
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    if (weather != null && "ok".equals(weather.status)) {
                        SharedPreferences.Editor editor =
PreferenceManager.getDefaultSharedPreferences(WeatherActivity.this).edit();
                        editor.putString("weather", responseText);
                        editor.apply();
                        mWeatherId = weather.basic.weatherId;
                        showWeatherInfo(weather);
                    } else {
                        Toast.makeText(WeatherActivity.this, "获取天气信息失败", Toast.LENGTH_SHORT).show();
                    }
                }
            });
            swipeRefresh.setRefreshing(false);
        }
    });
}
```

- 在onCreate () 方法中获取到SwipeRefreshLayout的实例，然后调用setColorSchemeResourecs () 方法来设置下拉刷新进度条的颜色，这里使用主题中的colorPrimary作为进度条的颜色。接着定义一个weatherId变量，用于记录城市的天气id，然后调用setOnRefreshListener () 方法来设置一个下拉刷新的监听器，当触发了下拉刷新操作的时候，就会回调这个监听器的onRefresh () 方法，在这里调用requestWeather () 方法请求天气信息就可以了。
- 当请求结束后，需要调用SwipeRefreshLayout的setRefreshing () 方法并传入false，用于表示刷新事件结束，并隐藏刷新进度条。
- 运行程序





谢 谢