# Animation：

| **java类名** | **xml关键字** | **描述信息** |
| --- | --- | --- |
| AlphaAnimation | <alpha> 放置在res/anim/目录下 | 渐变透明度动画效果 |
| RotateAnimation | <rotate> 放置在res/anim/目录下 | 画面转移旋转动画效果 |
| ScaleAnimation | <scale> 放置在res/anim/目录下 | 渐变尺寸伸缩动画效果 |
| TranslateAnimation | <translate> 放置在res/anim/目录下 | 画面转换位置移动动画效果 |
| AnimationSet | <set> 放置在res/anim/目录下 | 一个持有其它动画元素  alpha、scale、translate、rotate  或者其它set元素的容器 |

动画的属性：

# 定时任务：

定时统计上报任务：

## Handler：延时发消息，handler自己发消息；（一般在 Application onCreate（）方法触发）

private Handler handler = new Handler() {  
 @Override  
 public void handleMessage(Message msg) {  
 super.handleMessage(msg);  
 Log.*i*("session", "定时任务");//对应的逻辑处理方法  
 handler.sendEmptyMessageDelayed(MSG\_REPEAT\_TIME, 3000);  
 }  
};

## 用广播

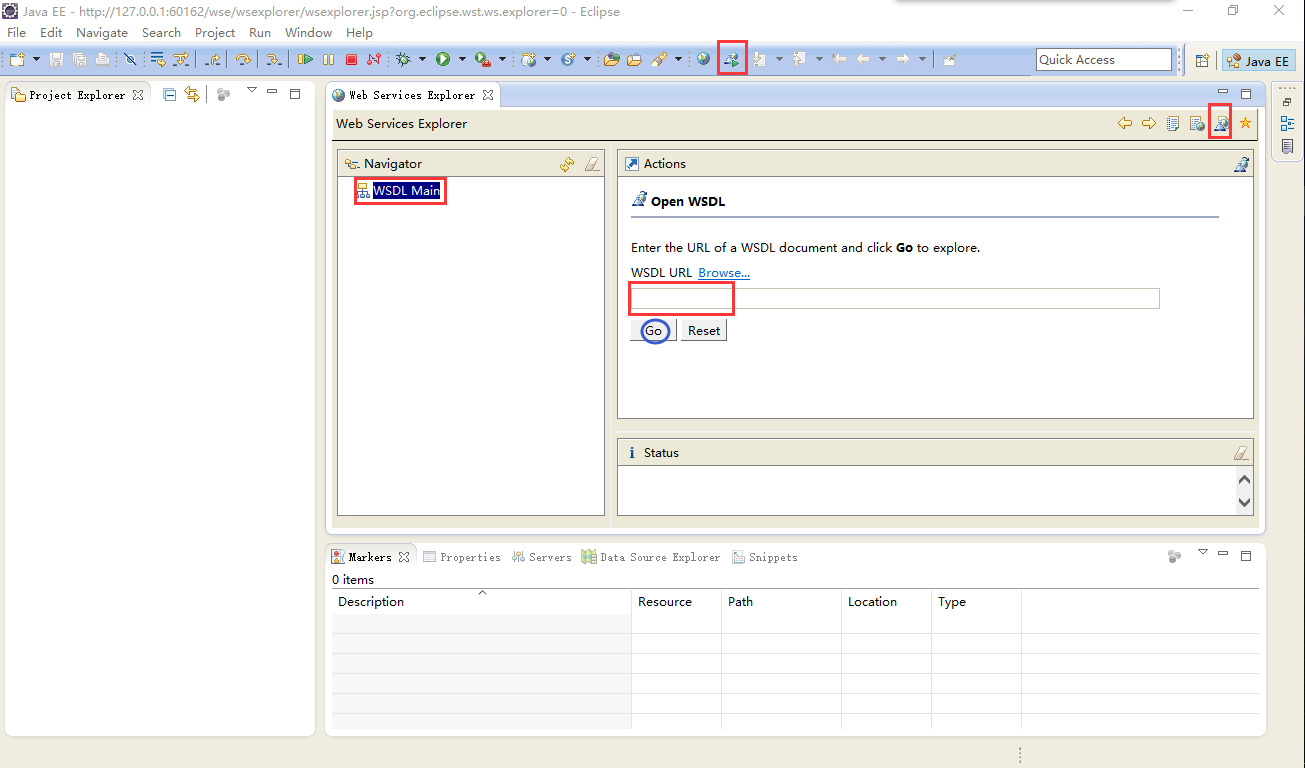
（注意有可能广播会被阻塞）

public class RepeatBroadcast extends BroadcastReceiver {  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 if (intent.getAction().equals(SdkUtil.*REPEAT\_ACTION*)) {  
 Logcat.*loge*("RepeatBroadcast ==>> onReceive: " + System.*currentTimeMillis*());  
 int ret = intent.getIntExtra("repeat", 60);  
  
 //上报本地数据  
 try {  
 HttpUtil.*getInstance*().sendLocalDataToServer();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 //上报缓存  
 SessionTime.*getInstance*(context).getUploadJson(true, false);  
  
 reportDataInTime(ret, context);  
 }  
 }  
  
  
  
 private void reportDataInTime(int second, Context mContext) {  
 AlarmManager alarmManager = (AlarmManager) mContext.getSystemService(Context.*ALARM\_SERVICE*);  
  
 Intent intent = new Intent(mContext, RepeatBroadcast.class);  
 intent.setAction(SdkUtil.*REPEAT\_ACTION*);  
 intent.putExtra("repeat", second);  
 PendingIntent pendingIntent = PendingIntent.*getBroadcast*(mContext, 1, intent, 0);  
  
 long startTime = SystemClock.*elapsedRealtime*();  
 long repeat = second \* 1000;  
  
 alarmManager.set(AlarmManager.*ELAPSED\_REALTIME\_WAKEUP*, startTime + repeat, pendingIntent);  
 }  
}

# webService网络编程

## webService模拟

材料要求：J2EE Eclipse ;jdk 1.7版本环境配置



## webService移动端网络编程 ：

动画属性：

# 位置问题：

rawX ， rawY 相对于屏幕的坐标  
x，y 相对于当前控件的坐标

# 自定义控件：

# RadioGroup：

我们可以通过在RadioGroup中添加默认的选中项，

通过语句android:checkedButton="@+id/radiobutton\_man"来设置默认选中项。具体代码如下：

# CheckBox：

//如果被选中，则密码显示；如果没有被选中，密码隐藏输入。默认显示是隐藏显示。

if (isChecked) {

mEditTextInputPassword.setTransformationMethod(null);

} else {

mEditTextInputPassword.setTransformationMethod(new PasswordTransformationMethod());

}

}