以下是getSystemService的两个方法：

public final T getSystemService (Class<T> serviceClass)

Added in API level 23

Return the handle to a system-level service by class.

public abstract Object getSystemService (String name)

Added in API level 1

Return the handle to a system-level service by name.

一般我们用的是第二个方法，写法有两种一个是用的Context里定义的字符串常量，当然意思是一样的，由于返回的对象是Object所以要对放回的值进行转型。

举个获取wifi管理器对象的例子：

WifiManager wifiManager = (WifiManager)getSystemService(Context.WIFI\_SERVICE);

WifiManager wifiManager = (WifiManager)getSystemService("wifi");

例子二：

WindowManager windowManager = (WindowManager) getSystemService(Context.WINDOW\_SERVICE);

WindowManager windowManager = (WindowManager) getSystemService("window");

查询API上归纳了所有可获取的类如下：

Currently available classes are:

WindowManager,WINDOW\_SERVICE ("window") ：窗口管理器

LayoutInflater, LAYOUT\_INFLATER\_SERVICE ("layout\_inflater")：布局

ActivityManager, ACTIVITY\_SERVICE ("activity")

PowerManager, POWER\_SERVICE ("power")：电源管理

AlarmManager, ALARM\_SERVICE ("alarm")：闹钟管理

NotificationManager,NOTIFICATION\_SERVICE ("notification")：通知栏管理

KeyguardManager, KEYGUARD\_SERVICE ("keyguard")

LocationManager, LOCATION\_SERVICE ("location")：位置管理

SearchManager, SEARCH\_SERVICE ("search")

Vibrator, VIBRATOR\_SERVICE ("vibrator") 手机震动

ConnectivityManager, CONNECTIVITY\_SERVICE ("connection")

WifiManager, WIFI\_SERVICE ("wifi")：wifi管理

AudioManager, WIFI\_P2P\_SERVICE ("wifip2p")

MediaRouter,Context.MEDIA\_ROUTER\_SERVICE这三个没有名称的方法

TelephonyManager, Context.TELEPHONY\_SERVICE

SubscriptionManager, 这个比较特殊You do not instantiate this class directly; instead, you retrieve a reference to an instance through from(Context).

应该这么写吧：SubscriptionManager ubscriptionManager = SubscriptionManager.from(this);

InputMethodManager, INPUT\_METHOD\_SERVICE ("input\_method")

UiModeManager, UI\_MODE\_SERVICE ("uimode")

DownloadManager, DOWNLOAD\_SERVICE ("download")

BatteryManager, BATTERY\_SERVICE ("batterymanager")：电池管理

JobScheduler,JOB\_SCHEDULER\_SERVICE ("taskmanager")

NetworkStatsManager.NETWORK\_STATS\_SERVICE ("netstats")

挺多的好多基本不认识，所以学会查看API是最基本的东西，这样才能方便学习。

这个个写个总结好像有点没完没了了。

贴上网上大神的总结吧

WindowManager

窗口管理

这个大家应用最多的应该都是用来获取手机屏幕的大小吧

WindowManager windowManager = (WindowManager) getSystemService(Context.WINDOW\_SERVICE);

int h = windowManager.getDefaultDisplay().getHeight();

int h = windowManager.getDefaultDisplay().getWidth();

LayoutInflater

用来获取布局文件的view

看介绍这个好像用到的概率比较小

http://www.cnblogs.com/top5/archive/2012/05/04/2482328.html

ActivityManager

获取系统内存信息以及进程信息

http://www.2cto.com/kf/201111/111395.html

PowerManager

电源管理

http://blog.csdn.net/xieqibao/article/details/6562256

AlarmManager

主要闹钟吧

http://blog.csdn.net/wangxingwu\_314/article/details/8060312

NotificationManager

这个通知栏的应该会用的比较多，基本的app都要用到吧

http://www.cnblogs.com/stay/articles/1898963.html

KeyguardManager

屏幕保护

http://blog.csdn.net/eyu8874521/article/details/8477724

LocationManager

定位服务

http://blog.csdn.net/liuhe688/article/details/6573459

SearchManager

利用此服务可以实现对系统中的应用、联系人、SMS等进行搜索

http://willsunforjava.iteye.com/blog/1674817

Vibrator

手机震动管理

http://www.android-study.com/jichuzhishi/414.html

ConnectivityManager

网络连接，这个应该用的比较多吧，那个app不需要网络，需要就要对网络状态进行判断

http://blog.csdn.net/chenzheng\_java/article/details/6387116/

WifiManager

对wifi的信息进行获取，判断处理

http://blog.csdn.net/hknock/article/details/7871866

AudioManager

手机音频文件的管理操作

http://blog.csdn.net/x605940745/article/details/16940097

MediaRouter

好像比较复杂的样子

http://blog.csdn.net/lilian0118/article/details/22940943

TelephonyManager

用于管理手机通话状态，获取电话信息(设备信息、sim卡信息以及网络信息)，

http://www.2cto.com/kf/201410/347844.html

SubscriptionManager

http://blog.163.com/machao\_163job/blog/static/2087753220121020101316838/

InputMethodManager

软键盘的输入控制

http://www.cnblogs.com/weixing/p/3300908.html

UiModeManager

设置夜间模式和行车模式

http://blog.sina.com.cn/s/blog\_8417aea80100v056.html

DownloadManager

下载管理

http://blog.csdn.net/sir\_zeng/article/details/8983430

JobScheduler

这个东西可以让系统批处理一些不重要的APP 请求

http://blog.csdn.net/cuiran/article/details/42805057

NetworkStatsManager

这个网上找不到说明文章，说明大家应该都没有吧