



군산대학교
KUNSAN NATIONAL UNIVERSITY

전화 통신과 SMS

모바일 시스템 SW

남 광 우

전화 통신을 위한 편의 수단들

❖ Telephony API

- ✓ 현재 통화증인지를 알아보거나, 전화서비스에 대한 정보를 얻거나, 주어진 전화번호가 유효한지를 점검
- ✓ `android.telephony.TelephonyManager` 클래스

❖ 주요 내용

- ✓ 전화기 상태 정보에 접근하기 위한 권한 얻기
- ✓ 통화 상태 조회
- ✓ 이동통신 서비스 정보 조회
- ✓ 전화번호 다루기

Telephony API

❖ 전화기 상태 정보에 대한 접근 권한 얻기

✓ AndroidManifest.xml

```
<uses-permission  
    android:name="android.permission.READ_PHONE_STATE" />  
<uses-permission  
    android:name="android.permission.SEND_SMS" />  
<uses-permission  
    android:name="android.permission.RECEIVE_SMS" />
```

Telephony API

❖ 통화 상태 조회

- ✓ TelephonyManager를 통해 전화기의 상태를 알수 있으며, 단말기의 전화번호와 같은 통화 서비스 자체에 대한 정보도 획득 가능
- ✓ TelephonyManager 객체 획득

```
TelephonyManager telManager = (TelephonyManager)  
    getSystemService( Context.TELEPHONY_SERVICE) ;
```

Telephony API

❖ 통화 상태 조회

- ✓ 현재의 통화 상태 파악
 - `getCallState()`

```
int callStatus = telManager.getCallState();

String callStateString = null;
switch (callState) {
    case TelephonyManager.CALL_STATE_IDLE:
        callStateString = "Phone is idle.";
        break;
    case TelephonyManager.CALL_STATE_OFFHOOK:
        callStateString = "Phone is in use.";
        break;
    case TelephonyManager.CALL_STATE_RINGING:
        callStateString = "Phone is ringing!";
        break;
}
```

Telephony API

❖ telephonyManager.listen() 의 사용

```
TelephonyManager telephonyManager = (TelephonyManager) getSystemService(Context.TELEPHONY_SERVICE);  
// 리스너로 phone의 상태가 다른 컴포넌트로 알려진다.  
//  
    PhoneStateListener phoneStateListener = new PhoneStateListener() {  
        public void onCallForwardingIndicatorChanged(boolean cfi) {} // 통화 전달.  
        public void onCallStateChanged(int state, String incomingNumber) {} // 통화상태(벨, 끊기)  
        public void onCellLocationChanged(CellLocation location) {} // 셀위치변경,  
        public void onDataActivity(int direction) {}  
        public void onDataConnectionStateChanged(int state) {} // 데이터 연결  
        public void onMessageWaitingIndicatorChanged(boolean mwi) {} // 대기  
        public void onServiceStateChanged(ServiceState serviceState) {} // 폰서비스변경,  
        public void onSignalStrengthChanged(int asu) {} // 모바일신호세기  
    };  
// 모니터링 할 이벤트를 리스너에 등록한다.  
    telephonyManager.listen(phoneStateListener,  
        PhoneStateListener.LISTEN_CALL_FORWARDING_INDICATOR |  
        PhoneStateListener.LISTEN_CALL_STATE |  
        PhoneStateListener.LISTEN_CELL_LOCATION |  
        PhoneStateListener.LISTEN_DATA_ACTIVITY |  
        PhoneStateListener.LISTEN_DATA_CONNECTION_STATE |  
        PhoneStateListener.LISTEN_MESSAGE_WAITING_INDICATOR |  
        PhoneStateListener.LISTEN_SERVICE_STATE |  
        PhoneStateListener.LISTEN_SIGNAL_STRENGTH);  
// 리스너를 해제  
    telephonyManager.listen(phoneStateListener, PhoneStateListener.LISTEN_NONE);
```

Telephony API

❖ Telephone 상태변화 감지 API

- ✓ `onCallForwardingIndicatorChanged()` : 호전환(Call Forwarding) 상태 변화
- ✓ `onCellLocationChanged()` : 단말기의 Cell위치 변화
- ✓ `onDataActivity()` : Data 활성화 상태 변화
- ✓ `onDataConnectionStateChanged()` : Data 연결상태 변화
- ✓ `onMessageWaitingIndicatorChanged()` : 메시지 대기상태 변화
- ✓ `onServiceStateChanged()` : 단말기의 서비스 상태 변화
- ✓ `onSignalStrengthsChanged()` : 망의 신호강도 변화

Telephony API

❖ 예 : 통화 상태의 변경시 수행

- ✓ listen() 메소드를 이용하여 현재 신호오는 전화번호 획득

```
telManager.listen(new PhoneStateListener() {  
    public void onCallStateChanged(int state, String incomingNumber) {  
        switch (state) {  
            case TelephonyManager.CALL_STATE_IDLE : break; // 폰이 울리거나 통화중이 아님.  
            case TelephonyManager.CALL_STATE_RINGING :  
                Log.i("telephony", newState + " number = " + incomingNumber);  
                break; // 폰이 울린다.  
            case TelephonyManager.CALL_STATE_OFFHOOK : break; // 폰이 현재 통화 중.  
            default: break; } else {  
                Log.i("telephony", newState);  
            }  
        }  
    }  
}, PhoneStateListener.LISTEN_CALL_STATE);
```

- onCallStateChanged() 메소드는 전화기의 벨이 울리거나, 사용자가 전화를 걸거나, 사용자가 전화를 받거나, 전화가 끊어지면 호출됨

Telephony API

❖ 예 : 모바일 데이터 전송 및 연결 모니터링

```
PhoneStateListener dataStateListener = new PhoneStateListener() {  
    public void onDataActivity(int direction) {  
        switch (direction) {  
            case TelephonyManager.DATA_ACTIVITY_IN : break;  
            case TelephonyManager.DATA_ACTIVITY_OUT : break;  
            case TelephonyManager.DATA_ACTIVITY_INOUT : break;  
            case TelephonyManager.DATA_ACTIVITY_NONE : break;  
        }  
    }  
    public void onDataConnectionStateChanged(int state) {  
        switch (state) {  
            case TelephonyManager.DATA_CONNECTED : break;  
            case TelephonyManager.DATA_CONNECTING : break;  
            case TelephonyManager.DATA_DISCONNECTED : break;  
            case TelephonyManager.DATA_SUSPENDED : break;  
        }  
    }  
};  
telephonyManager.listen(dataStateListener,  
    PhoneStateListener.LISTEN_DATA_ACTIVITY |  
    PhoneStateListener.LISTEN_DATA_CONNECTION_STATE);
```

Telephony API

❖ 통화 서비스 자체의 상태 정보 파악

```
ServiceState serviceState = new ServiceState();
int serviceStatus = serviceState.getState();
String serviceStateString = getServiceStateString(serviceStatus);

private String getServiceStateString(int serviceState) {
    String serviceStateString = null;
    switch (serviceState) {
        case ServiceState.STATE_EMERGENCY_ONLY:
            serviceStateString = "Emergency calls only";
            break;
        case ServiceState.STATE_IN_SERVICE: // 서비스 망 내
            serviceStateString = "Normal service";
            break;
        case ServiceState.STATE_OUT_OF_SERVICE: // 서비스망 밖
            serviceStateString = "No service available";
            break;
        case ServiceState.STATE_POWER_OFF: //전원 off
            serviceStateString = "Telephony radio is off";
            break;
    }

    return serviceStateString;
}
```

Telephony API

❖ 이동통신 서비스 조회

```
String opName = telManager.getNetworkOperatorName();  
Log.i("telephony", "operator name = " + opName);  
  
String phoneNumber = telManager.getLine1Number();  
Log.i("telephony", "phone number = " + phoneNumber);  
  
String providerName = telManager.getSimOperatorName();  
Log.i("telephony", "provider name = " + providerName);
```

Telephony API

❖ 로밍 중인지 파악하기

```
boolean isRoaming = serviceState.getRoaming();  
Log.i("telephony", "Roaming state is " + isRoaming);
```

❖ 전화번호 포맷 맞추기

```
String formattedNumber = PhoneNumberUtils.formatNumber("9995551212");  
Log.i("telephony", formattedNumber);
```

Telephony API

❖ 전화걸기

- ✓ Intent.ACTION_CALL
 - 자동으로 전화 걸고, 통화중 애플 표시
 - 네이티브 다이얼러를 대체하는 경우만 이용해야 함.
 - CALL_PHONE 권한을 부여 받아야 함
- ✓ Intent.ACTION_DIAL
 - 바로 전화를 걸지 않고, 다이얼러 시작 - 지정된 번호 전달.

```
Intent intent = new Intent(Intent.ACTION_CALL, Uri.parse("tel:1234567"));
startActivity(intent);
```

```
<uses-permission
android:name="android.permission.CALL_PHONE" />
```

SMS API

❖ SMS 전송 수신을 위한 권한 얻기

✓ AndroidManifest.xml;

```
<uses-permission  
    android:name="android.permission.READ_PHONE_STATE" />  
<uses-permission  
    android:name="android.permission.SEND_SMS" />  
<uses-permission  
    android:name="android.permission.RECEIVE_SMS" />
```

SMS API

❖ SMS 보내기

```
final SmsManager sms = SmsManager.getDefault();  
  
sms.sendTextMessage("0164028040", null, "바쁘삼???", null, null );
```

✓ Multipart 메시지 보내기

- divideMessage() 메소드 이용하여, SMS 최대 크기로 메시지 절단
- sendMultipartTextMessage() 메소드를 호출하여 전송

```
public void sendTextMessage (String destinationAddress, String scAddress,  
String text, PendingIntent sentIntent, PendingIntent deliveryIntent)
```

```
public void sendMultipartTextMessage (String destinationAddress, String scAddress,  
ArrayList parts, ArrayList sentIntents, ArrayList deliveryIntents)
```

SMS API

❖ SMS 전송 성공 여부의 확인

```
Intent msgSent = new Intent("ACTION_MSG_SENT");

final PendingIntent pendingMsgSent = PendingIntent.getBroadcast(this, 0, msgSent, 0);

rcvMsgSent = new BroadcastReceiver() {

    @Override
    public void onReceive(Context context, Intent intent) {
        int result = getResultCode();
        if (result != Activity.RESULT_OK) {
            Log.e("telephony", "SMS send failed code = " + result);
            pendingMsgReceipt.cancel();
        } else {
            Log.i("telephony", "SMS message sent!");
            messageEntry.setText("");
        }
    }
};

registerReceiver(rcvMsgSent, new IntentFilter("ACTION_MSG_SENT"));
```


SMS API

❖ Broadcast Receiver를 이용하여 SMS 받기

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.smsreceiver"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="4" />
    <uses-permission android:name="android.permission.RECEIVE_SMS" />
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity
            android:name=".SmsLoggerActivity"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <receiver android:name="com.example.YourSMSReceiver" android:enabled="true">
            <intent-filter>
                <action android:name="android.provider.Telephony.SMS_RECEIVED" />
            </intent-filter>
        </receiver>
    </application>
</manifest>
```

SMS API

❖ Broadcast Receiver를 이용하여 SMS 받기

```
final TextView receivedMessage = (TextView)findViewById(R.id.received_message);
rcvIncoming = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent.getAction().equals("android.provider.Telephony.SMS_RECEIVED")) {
            Log.i("telephony", "SMS received");
            Bundle data = intent.getExtras();
            if (data != null) {
                // SMS uses a data format known as a PDU
                Object pdus[] = (Object[]) data.get("pdus");
                String message = "New message:\n";
                String sender = null;
                for (Object pdu : pdus) {
                    SmsMessage part = SmsMessage.createFromPdu((byte[])pdu);
                    message += part.getDisplayMessageBody();
                    if (sender == null) {
                        sender = part.getDisplayOriginatingAddress();
                    }
                }
                receivedMessage.setText(message+"\nFrom: "+sender);
                numberEntry.setText(sender);
            }
        }
    }
};

registerReceiver(rcvIncoming, new IntentFilter("android.provider.Telephony.SMS_RECEIVED"));
```

multipart 메시지의 처리

전화걸기

❖ 전화걸기

```
Button call = (Button) findViewById(R.id.call_button);
call.setOnClickListener(new View.OnClickListener() {

    public void onClick(View v) {
        Uri number = Uri.parse("tel:" + numberEntry.getText().toString());
        Intent dial = new Intent(Intent.ACTION_DIAL, number);
        startActivity(dial);
    }

});
```

idea? and report

❖ 선거 SMS 자동 삭제 프로그램

- ✓ 목적 : 특정 문자/전화번호 발신 SMS 스팸처리하기
- ✓ 이메일의 스팸처리기능을 SMS에 적용

문구	<input type="text"/>
전화번호	<input type="text"/>
and <input type="radio"/> or <input type="radio"/>	
처리방법	
Spam 저장함으로 이동	<input type="radio"/>
자동삭제	<input type="radio"/>

idea? and report

❖ Spam 전화번호 Google 자동 검색 프로그램

- ✓ IncomingCall을 intercept 하기

```
public class IncommingCallInterceptor extends BroadcastReceiver {
    @override
    public void onReceive( Context context, Intent intent )
    {
        String state = intent.getStringExtra( TelephonyManager.EXTRA_STATE);
        String msg = "Phone state changed to " + state ;

        if (telephonyManager.EXTRA_STATE_RINGING.equals(state))
        {
            String number = intent.getStringExtra(TelephonyManager.EXTRA_INCOMING_NUMBER);
            msg = " Incoming phone number is " + number;
        }

        Toast.makeText( context, msg, Toast.LENGTH_LONG).show
    }
}
```

<http://blog.naver.com/PostView.nhn?blogId=onefinedays&logNo=50141672280>

idea? and report

❖ Spam 전화번호 Google 자동 검색 프로그램

✓ manifest.xml

```
<application android:icon="@drawable/icon" android:label="@string/app_name">
    <activity android:name=".PhoneCall2"
        android:label="@string/app_name">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>

    </activity>
    <receiver android:name=".PhoneCallReceiver">
        <intent-filter android:priority="100" >
            <action android:name="android.intent.action.PHONE_STATE" />
        </intent-filter>
    </receiver>

</application>
<uses-sdk android:minSdkVersion="3" />
<uses-permission android:name="android.permission.MODIFY_PHONE_STATE" />
<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />

</manifest>
```

idea? and report

❖ Spam 전화번호 Google 자동 검색 프로그램

- ✓ Google Custom-search API
 - JSON으로 결과를 반환

https://developers.google.com/custom-search/v1/using_rest

- ✓ Google API Key를 발급 받음
 - <https://code.google.com/apis/console/?api=customsearch>

- ✓ Google CX 발급 받음
 - <http://www.google.com/cse>

- ✓ REST API
 - HTTP GET
 - https://www.googleapis.com/customsearch/v1?key=INSERT-YOUR-KEY&cx=013036536707430787589:_pqjad5hr1a&q=flowers&alt=json

idea? and report

❖ Spam 전화번호 Google 지

Google 맞춤검색

kwangwoo@gmail.com | 내계정 | 도움말 | 로그인

1. 검색 엔진 설정 > 2. 사용해 보기 > 3. 코드 가져오기

- 웹사이트를 하나 이상 입력해야 합니다.

검색엔진 설명

이름:
예: 실제 기후 검색엔진

설명:
예: 기후 과학자의 기후 과학

언어:

검색엔진 정의

검색할 사이트:

한 줄에 하나의 URL만 나열하세요.

URL 서식 지정에 대해 [자세히 알아보기](#)

URL 서식 지정

- 개별 페이지: `www.mysite.com/mypage.html`을 지정하면 `www.mysite.com`의 `mypage.html` 페이지만 포함됩니다.
- 전체 사이트: `www.mysite.com/*`을 지정하면 `www.mysite.com`의 모든 페이지가 포함됩니다.
- 사이트 일부: 사이트의 특정 부분만 포함하도록 와일드카드 패턴을 사용할 수 있습니다. 예를 들어 `www.mysite.com/about*`은 `www.mysite.com`에서 이름에 `about`이 있는 파일만 포함합니다.
- 전체 도메인: `*.mydomain.com`을 사용하여 전체 도메인을 지정할 수도 있습니다. `mydomain.com`을 지정하면 자동으로 `*.mydomain.com/*`으로 변환됩니다. 변환된 도메인을 원하지 않을 경우 제어판에서 다시 변경할 수 있습니다.

버전 선택

- 버전:
- ☒ 표준형: 무료이며 검색결과 페이지에 광고가 게재됩니다.
 - ☐ 사이트 검색: 최저 요금은 연간 미화 100달러이며 검색결과 페이지에 광고 게재는 선택사항입니다.

각 버전에 대해 [자세히 알아보기](#)

☐ 다음을 읽었으며 동의: [서비스 약관](#)

idea? and report

❖ Spam 전화번호 Google 자동 검색 프로그램

```
https://www.googleapis.com/customsearch/v1?q={searchTerms}&num={count?}&start={startIndex?}&lr={language?}&safe={safe?}&cx={cx?}&cref={cref?}&sort={sort?}&filter={filter?}&gl={gl?}&cr={cr?}&googlehost={googleHost?}&cc={cc?}&coff={disableOnTranslation?}&hq={hq?}&hl={hl?}&siteSearch={siteSearch?}&siteSearchFilter={siteSearchFilter?}&exactTerms={exactTerms?}&excludeTerms={excludeTerms?}&linkSite={linkSite?}&orTerms={orTerms?}&relatedSite={relatedSite?}&dateRestrict={dateRestrict?}&lowRange={lowRange?}&highRange={highRange?}&searchType={searchType?}&fileType={fileType?}&rights={rights?}&imgSize={imgSize?}&imgType={imgType?}&imgColorType={imgColorType?}&imgDominantColor={imgDominantColor?}&alt=json"
},
"queries": {
  "request": [
    {
      "title": "Google Custom Search - flowers",
      "totalResults": "2",
      "searchTerms": "flowers",
      "count": 2,
      "startIndex": 1,
      "inputEncoding": "utf8",
      "outputEncoding": "utf8",
      "safe": "off",
      "cx": "004800638323602830096:qyudrzjzxs8"
    }
  ]
},
"context": {
  "title": "phone number search"
},
"searchInformation": {
  "searchTime": 0.19439,
  "formattedSearchTime": "0.19",
  "totalResults": "2",
  "formattedTotalResults": "2"
},
"items": [
  {
    "kind": "customsearch#result",
    "title": "01094958947 - 우리가 만드는 전화번호부 - 더콜",
    "htmlTitle": "01094958947 - 우리가 만드는 전화번호부 - 더콜",
    "link": "http://www.thecall.co.kr/bbs/board.php?bo_table=phone&wr_id=138330",
    "displayLink": "www.thecall.co.kr",
    "snippet": "2012년 9월 4일 ... 로즈flower 밤 날씨가 많이 쌀쌀해졌습니다 건강 유념하시고 항상 행복하세요~^^. 다음에 해당되는 글이 있다면, 더욱 정확한 '더콜'을 위하여 [삭제 ...",
    "htmlSnippet": "2012년 9월 4일 #u003c#u003e...#u003c#u003e 로즈flower#u003c#u003e 밤 날씨가 많이 쌀쌀해졌습니다 건강 유념하시고 항상 행복하세요~^^. #u003c#u003e 다음에 해당되는 글이 있다면, 더욱 정확한 &#39;더콜&#39;을 위하여 [삭제 #u003c#u003e...#u003c#u003e",
    "cacheId": "pShA2PVTgFsj",
    "formattedUrl": "www.thecall.co.kr/bbs/board.php?bo_table=phone&wr...",
    "htmlFormattedUrl": "www.thecall.co.kr/bbs/board.php?bo_table=phone&wr...",
    "pageMap": {
      "cse_image": [
        {
          "src": "http://www.thecall.co.kr/img/thecall_square.png"
        }
      ]
    },
    "metatags": [
      {
        "og:title": "우리가 만드는 전화번호부 - 더콜",
        "og:type": "website",
        "og:url": "http://www.thecall.co.kr/bbs/board.php?bo_table=phone",
        "og:image": "http://www.thecall.co.kr/img/thecall_square.png",
        "og:site_name": "더콜"
```