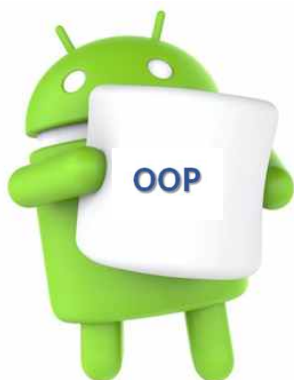


바코드리더기를 활용한 냉장고 관리



소 속	컴퓨터정보계열
과 목 명	안드로이드
담당교수	백건효
팀 명	NoError
팀 장	한승일
팀 원	박선재, 우성욱
제 출 일	2018년 06월 26일

목차

1. 개요 -----	3
- 프로젝트 주제 및 목표	
- 프로젝트 개발 일정 및 개발환경	
2. 안드로이드 앱 개발 -----	4
- 개요	
- 안드로이드 UI 및 설계 구현	
- 주요소스 코드	
- 카카오톡 로그인 연동하기	
3. 참고자료 및 출처 -----	36

1. 개요

1. 프로젝트 주제 및 목표

- 주제 : 바코드를 활용한 냉장고 관리 시스템
- 주제 선정 이유 : 냉장고를 열어보면 유통기한이 지나버린 음식이 가득했던 경우가 누구나 한번쯤 있을 것입니다. 사다놓은 재료를 깜빡 잊고 또 사거나, 냉장고 깊숙이 넣어둔 음식이 상해서 버리는 경우도 많아 돈은 돈대로, 식재료는 식재료대로 낭비되어 골치가 아프다. 이러한 문제점을 조금이라도 해결하고 간편하고 체계적으로 냉장고를 관리할 수 있는 앱을 만들고자 합니다.

2. 프로젝트 개발 일정 및 개발 환경

- 개발일정

시기 단계별 내용	5월			6월			
	3주	4주	5주	1주	2주	3주	4주
주제 선정							
자료 수집 및 업무정의서							
UseCase Diagram							
Class Diagram							
Sequence Diagram							
Activity Diagram & ERD							
UI 제작							
시스템 구현							
보고서 작성							
발표 자료 작성							

- 개발환경

1. 안드로이드 스튜디오



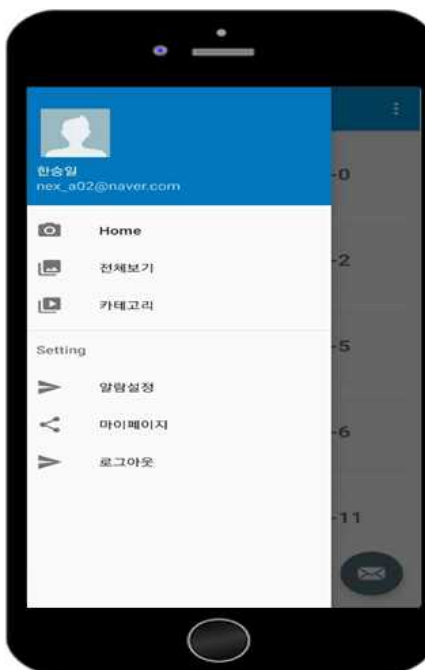
2. 닷홈



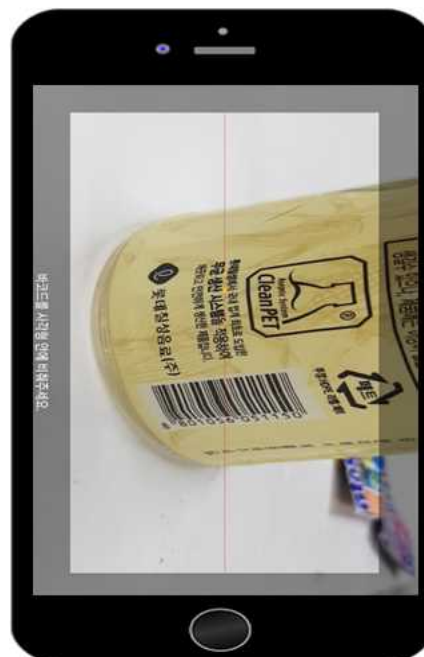
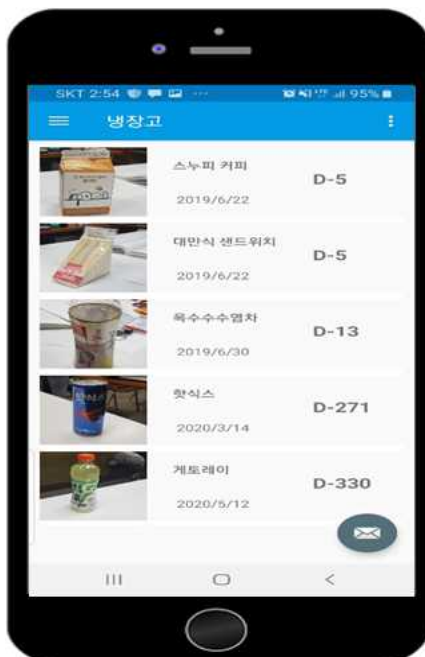
3. 카카오



2. 안드로이드 UI 설계 및 구현



2. 안드로이드 UI 설계 및 구현



3. 주요 소스 코드

1. ZXing라이브러리를 이용하여 바코드 구현하기

< build.gradle(Module:app) >

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

< AndroidManifest.xml >

```
<activity android:name=".qr.CaptureActivityAnyOrientation"
    android:screenOrientation="fullSensor"
    android:stateNotNeeded="true"
    android:theme="@style/zxing_CaptureTheme"
    android:windowSoftInputMode="stateAlwaysHidden"
/>
```

-zxing을 사용하기 위해 라이브러리를 추가한다.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setTitle("냉장고");

    setContentView(R.layout.activity_main);
    Toolbar toolbar = findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
    FloatingActionButton fab = findViewById(R.id.fab);

    fab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

            Intent intent = new Intent(getApplicationContext(), ScanResult.class);
            startActivity(intent);
        }
    });
}
```

- fab버튼 클릭 시 Intent로 ScanResult.class로 이동이된다.

3. 주요 소스 코드

```

/*****권한 요청*****/
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
    /**
     * 사용자 단말기의 권한 중 "카메라" 권한이 허용되어 있는지 확인한다.
     * Android는 C언어 기반으로 만들어졌기 때문에 Boolean 타입보다 Int 타입을 사용한다.
     */
    int permissionResult = checkSelfPermission(Manifest.permission.CAMERA);

    /** * 패키지는 안드로이드 어플리케이션의 아이디이다. *
     * 현재 어플리케이션이 카메라에 대해 거부되어있는지 확인한다. */
    if (permissionResult == PackageManager.PERMISSION_DENIED) {

        /** * 사용자가 CALL_PHONE 권한을 거부한 적이 있는지 확인한다. *
         * 거부한적이 있으면 True를 리턴하고 *
         * 거부한적이 없으면 False를 리턴한다. */
        if (shouldShowRequestPermissionRationale(Manifest.permission.CAMERA)) {
            AlertDialog.Builder dialog = new AlertDialog.Builder(MainActivity.this);
            dialog.setTitle("권한이 필요합니다.").setMessage("이 기능을 사용하기 위해서는 단말기의 W"카메라W" 권한이 필요합니다. 계속 하시겠습니까?");
            dialog.setPositiveButton("네", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    /** * 새로운 인스턴스(onClickListener)를 생성했기 때문에 *
                     * 버전체크를 다시 해준다. */
                    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                        // CALL_PHONE 권한을 Android OS에 요청한다.
                        requestPermissions(new String[]{Manifest.permission.CAMERA}, 1000);
                    }
                }
            })
            dialog.setNegativeButton("아니요", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    Toast.makeText(MainActivity.this, "기능을 취소했습니다", Toast.LENGTH_SHORT).show();
                }
            });
            dialog.create().show();
        }
        // 최초로 권한을 요청할 때
        else {
            // CALL_PHONE 권한을 Android OS에 요청한다.
            requestPermissions(new String[]{Manifest.permission.CAMERA}, 1000);
        }
    }
    // CALL_PHONE의 권한이 있을 때
    else {
    }
}
/*****권한요청 끝*****/

}

public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
    if (requestCode == 1000) {
        // 요청한 권한을 사용자가 "허용" 했다면...
        if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            // 이곳에 허용했을때 실행할 코드를 넣는다
            // 근데 난 안넣음
        } else {
            // 거부했을때 띄워줄
            Toast.makeText(MainActivity.this, "권한요청을 거부했습니다.", Toast.LENGTH_SHORT).show();
        }
    }
}
}

```

- 바코드스캐너를 사용하기 위해서는 단말기의 '카메라' 권한이 필요하기 때문에 사용자에게 권한을 요청하는 코드이다.

3. 주요 소스 코드

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

    IntentIntegrator integrator = new IntentIntegrator(this);
    integrator.initiateScan(); //바코드 스캔 카메라 실행
```

-integrator.initiateScan()가 실행되면 바코드 카메라가 실행된다

```
protected void onActivityResult(int requestCode,int resultCode, Intent data){
    super.onActivityResult(requestCode, resultCode, data);

    // QR코드/ 바코드를 스캔한 결과
    IntentResult result = IntentIntegrator.parseActivityResult(requestCode, resultCode, data);
    // result.getFormatName() : 바코드 종류 result.getContents() : 바코드 값
    final String BARCODE = result.getContents().toString(); //바코드 결과값 변수에 저장

    //바코드 결과 값 전역변수로 선언
    GlobalApplication barcode = (GlobalApplication)getApplication();
    barcode.setBarcode(BARCODE);
    Log.e("barcode",barcode.getBarcode().toString());
}
```

- 바코드를 스캔 하였을 때 스캔된 값을 전역변수로 선언하는 코드이다.
- 데이터베이스에 연동하여 사용하기 위해 바코드 결과 값을 전역변수로 선언하였다.

3. 주요 소스 코드

```
class BackgroundTask extends AsyncTask<Void, Void, String> {
    String target;

    protected void onPreExecute() {
        target = "http://qkfldk197.dothome.co.kr/barcode.php";
    }

    @Override
    protected String doInBackground(Void... voids) { //데이터를 받아오는 부분

        try {
            URL url = new URL(target);
            HttpURLConnection httpURLConnection = (HttpURLConnection)
            url.openConnection();
            InputStream inputStream = httpURLConnection.getInputStream(); //넘어오는 결과
            값 저장
            BufferedReader bufferedReader = new BufferedReader(new
            InputStreamReader(inputStream));
            String temp;
            StringBuilder stringBuilder = new StringBuilder();
            while ((temp = bufferedReader.readLine()) != null) {
                stringBuilder.append(temp + "\n");
            }
            bufferedReader.close();
            inputStream.close();
            httpURLConnection.disconnect();
            return stringBuilder.toString().trim();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }
}
```

- 바코드 정보를 검색하기 위해 데이터베이스를 연동하는 코드이다.

```
<?php
——$con=mysqli_connect("localhost", "qkfldk197", "hsi0206197", "qkfldk197");
……—mysqli_set_charset($con, "utf8");
——$result=mysqli_query($con, "select * from barcode");
——$response=array();

——while($row=mysqli_fetch_array($result)){
————array_push($response, array("barcodeNum"=>$row[0], "foodName"=>$row[1], "foodDate"=>$row[2], "
————foodCategory"=>$row[3], "foodImage"=>$row[4]));
——}

——echo json_encode(array("response"=>$response));
——mysqli_close($con)
?>
```

- 데이터베이스 연동에 사용된 barcode.php
- barcode테이블에 있는 모든 바코드 정보를 검색하도록 작성하였다.

3. 주요 소스 코드

```

public void onPostExecute(String result) { //해당결과를 처리(결과값을 result 로 받
아옴)

    ImageView foodImageView = (ImageView)findViewById(R.id.foodImageView);
    TextView bacordeNumtxt = (TextView) findViewById(R.id.scanResult);
    TextView foodNametxt = (TextView) findViewById(R.id.foodNameTxt);
    TextView foodDatetxt = (TextView) findViewById(R.id.foodDate);
    TextView foodCategorytxt = (TextView) findViewById(R.id.category);

    try {
        JSONObject jsonObject = new JSONObject(result);
        JSONArray jsonArray = jsonObject.getJSONArray("response");
        int count = 0;
        String bacordNum, foodName, foodDate, foodCategory, foodImage;

        while (count < jsonArray.length()) {

            GlobalApplication barcode = (GlobalApplication)getApplication();
            Log.e("scanBarcode", barcode.getBarcode().toString());

            JSONObject object = jsonArray.getJSONObject(count);
            bacordNum = object.getString("barcodeNum");
            foodName = object.getString("foodName");
            foodDate = object.getString("foodDate");
            foodCategory = object.getString("foodCategory");
            foodImage = object.getString("foodImage");

            if(bacordNum.equals(barcode.getBarcode())){

                Log.e("dbBarcode", bacordNum);
                bacordeNumtxt.setText(bacordNum);
                foodNametxt.setText(foodName);
                foodDatetxt.setText(foodDate);
                foodCategorytxt.setText(foodCategory);
            }
        }
    }
}

```

- 데이터베이스 값을 json형태로 받아와서 파싱, 데이터베이스에 저장된 바코드 값과 스캐너로 찍은 바코드값을 비교하여 일치하는 값을 불러온다.

3. 주요 소스 코드

- 안드로이드 어플을 구현하는 과정에서 많은 양의 이미지를 보여줘야 하는 경우가 많은 데, 많은 양의 이미지를 모두 어플에 넣는다면 용량이 증가하는 문제가 발생하게 되는 데 이미지url을 사용함으로써 이러한 문제를 해결할 수 있다.

barcodeNum	foodName	foodDate	foodCategory	foodImage
8801056051150	옥수수수염차	2020/5/27	음료	http://img.danawa.com/prod_img/500000/299/148/img/...
8801056290016	레쓰비	2019/9/11	음료	https://www.costco.co.kr/medias/sys_master/h67/h25...

- 하지만 이미지url로는 바로 ImageView에 불러올 수가 없다.

```
Thread mThread = new Thread() {
    @Override
    public void run() {
        try {

            URL url = new URL(foodImage);

            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setDoInput(true);
            conn.connect();

            InputStream is = conn.getInputStream();
            bitmap = BitmapFactory.decodeStream(is); //가져온 이미지 주소를 bitmap으로 변환

        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
};

mThread.start();

try{
    mThread.join();
    foodImageView.setImageBitmap(bitmap);
}catch (InterruptedException e){
    e.printStackTrace();
}
```

- 이미지뷰에 url로 이미지를 지정하는 방법은 이미지를 Bitmap 로딩하여 이미지뷰에 지정할 수 있다.

3. 주요 소스 코드

2. 카메라, 앨범 라이브러리 사용하여 이미지 불러오기

< AndroidManifest.xml >

```
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

- 카메라와 갤러리에 접근할 수 있는 권한을 받아오기 위함.

< build.gradle(Module:app) >

```
implementation "gun0912.ted:tedpermission:2.1.0"
```

```
private void goToAlbum(){
    isCamera=false;

    Intent intent = new Intent(Intent.ACTION_PICK); //갤러리 접근
    intent.setType(MediaStore.Images.Media.CONTENT_TYPE);
    startActivityForResult(intent, PICK_FROM_ALBUM);
}

private void TakePhoto(){
    Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE); //카메라접근
    ...
}
```

- 외부라이브러리를 사용하여 Intent로 손쉽게 갤러리와 카메라에 접근할 수 있다.

3. 주요 소스 코드

```
private File createImageFile() throws IOException {
    // 이미지 파일 이름 (blackJin_{저장 된 시간}_ )
    String timeStamp = new SimpleDateFormat("HHmmss").format(new Date());
    String imageFileName = "blackJin_" + timeStamp + "_";

    // 이미지가 저장될 폴더 이름 ( blackJin )
    File storageDir = new File(Environment.getExternalStorageDirectory() + "/blackJin/");
    if (!storageDir.exists()) storageDir.mkdirs();

    // 빈 파일 생성
    File image = File.createTempFile(imageFileName, ".jpg", storageDir);

    return image;
}
```

- 촬영한 사진이 저장될 폴더를 생성하고 사진파일의 이름을 지정해주는 코드

```
private void takePhoto() {
    isCamera = true;

    Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);

    try {
        tempFile = createImageFile(); //촬영한 사진 저장
    } catch (IOException e) {
        Toast.makeText(this, "이미지 처리 오류! 다시 시도해주세요.", Toast.LENGTH_SHORT).show();
        finish();
        e.printStackTrace();
    }

    if (tempFile != null) { //촬영한 사진이 있다면

        Uri photoUri = Uri.fromFile(tempFile);
        intent.putExtra(MediaStore.EXTRA_OUTPUT, photoUri);
        startActivityForResult(intent, PICK_FROM_CAMERA);
    }
}
}
```

- tempFile에 촬영한 사진이 저장되고 Uri경로를 intent에 추가해 줌으로써 사진이 저장될 경로를 지정해 준다.

3. 주요 소스 코드

```
int mYear, mMonth, mDay, mHour, mMinute; //날짜를 받을 변수
TextView foodDateTxt;

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

    //텍스트뷰 연결
    foodDateTxt = (TextView)findViewById(R.id.foodDateTxt);

    //현재 날짜와 시간을 가져오기 위한 Calendar 인스턴스 선언
    Calendar cal = new GregorianCalendar();
    mYear = cal.get(Calendar.YEAR);
    mMonth = cal.get(Calendar.MONTH);
    mDay = cal.get(Calendar.DAY_OF_MONTH);

    UpdateNow();//화면 텍스트뷰에 업데이트 해줌.
}

void UpdateNow(){
    foodDateTxt.setText(String.format("%d/%d/%d", mYear, mMonth + 1, mDay));
}
}
```

- 안드로이드스튜디오에서 제공하는 달력 라이브러리를 이용해서 선택한 날짜의 값을 받아오는 코드
- 날짜를 받을 변수를 선언하고 Calendar 인스턴스를 사용하여 변수에 년,월,일 값들을 저장한다.

3. 주요 소스 코드

```
//등록하기 버튼
Button foodInsertBtn = (Button)findViewById(R.id.insertBtn);
foodInsertBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        GlobalApplication mail = (GlobalApplication)getApplication(); //전역변수 email 가져오기

        String foodName = foodNameTxt.getText().toString();
        String foodDate = foodDateTxt.getText().toString();
        String category = spinner.getSelectedItem().toString();
        String Email = mail.getEmail().toString();
        Log.e("saveEmail2", mail.getEmail());

        //빈칸이 존재할 때
        if(foodName.equals("") || foodDate.equals("") || category.equals("-선택하세요-") ||
        division.equals("")){

            Dialog dialog;
            AlertDialog.Builder builder = new AlertDialog.Builder(foodInsert.this);
            dialog = builder.setMessage("빈칸이 있습니다.")
                .setNegativeButton("확인",null)
                .create();
            dialog.show();
            return;
        }
    }
};
```

- 등록하기 버튼을 통해서 냉장고에 내 음식을 등록하는 코드이다.
- 버튼을 클릭 했을 때 빈칸이 존재한다면 팝업창이 뜨면서 등록되지 않는다.

```
//빈칸이 존재하지 않으면 계속 진행 (디비연결부분)
Response.Listener<String> responseListener = new Response.Listener<String>() { //특정 요청 이후에
    리스너에서 원하는 결과값을 다룰수 있게 함
    @Override
    public void onResponse(String response) {
        try {
            //자바에서 서버로 데이터를 전달하여 클라이언트와 주고받기 위해 JSON 사용
            JSONObject jsonResponse = new JSONObject(response); //특정 response를 실행 했을 때 결과가
            담길 수 있도록 한다.
            boolean success = jsonResponse.getBoolean("success");
            if(success) { //데이터가 db에 저장됐을 때(회원가입 성공했을 때)

                AlertDialog.Builder builder = new AlertDialog.Builder(foodInsert.this);
                builder.setMessage("등록 되었습니다!").setPositiveButton("확인", new
                DialogInterface.OnClickListener() {

                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        Log.e("test", "test5");
                        Intent intent = new Intent(foodInsert.this, MainActivity.class);
                        foodInsert.this.startActivity(intent);
                    }
                });
                AlertDialog dialog = builder.create();
                dialog.show();
            }else {
                AlertDialog.Builder builder = new AlertDialog.Builder(foodInsert.this);
                builder.setMessage("등록에 실패했습니다.").setNegativeButton("다시 시도",null).create().show();
            }
        } catch (JSONException e){
            e.printStackTrace();
        }
    }
};
```

- 빈칸이 존재하지 않는다면 데이터베이스 연동을 한다.

3. 주요 소스 코드

```
Log.e("test", "test4");
foodInsertRequest foodInsertRequest = new foodInsertRequest(foodName, foodDate, division,
category, foodImage, Email, responseListener); //StringRequest를 받은 객체를 부름
RequestQueue queue = Volley.newRequestQueue(foodInsert.this);
queue.add(foodInsertRequest);
```

```
public class foodInsertRequest extends StringRequest {

    final static private String URL = "http://qkfldk197.dothome.co.kr/foodInsert.php";
    private Map<String, String> parameters;

    public foodInsertRequest(String foodName, String foodDate, String division, String category,
String foodImage, String Email, Response.Listener<String> listener) { //생성자 호출
        super(Method.POST, URL, listener, null); //상속받은 StringRequest 의 생성자를 부른다. (어떤 방법
        으로 어디에, 에러시)
        parameters = new HashMap<>(); //해쉬맵 형태로 값을 받아옴
        parameters.put("foodName", foodName);
        parameters.put("foodDate", foodDate);
        parameters.put("division", division);
        parameters.put("category", category); //("태그", 실제데이터) 형태로 저장
        parameters.put("foodImage", foodImage);
        parameters.put("Email", Email);
    }
}
```

```
<?php
...$con=mysqli_connect("localhost", "qkfldk197", "hsi0206197", "qkfldk197");
...mysqli_set_charset($con, "utf8");

...$foodName=$_POST["foodName"];
...$foodDate=$_POST["foodDate"];
...$division=$_POST["division"];
...$category=$_POST["category"];
...$foodImage=$_POST["foodImage"];
...$Email=$_POST["Email"];

...$statement=mysqli_prepare($con, "insert into Refrigerator values(?,?,?,?,?,?,?)");
...mysqli_stmt_bind_param($statement, "ssssss", $foodName, $foodDate, $division, $category, $foodImage,
    $Email);
...mysqli_stmt_execute($statement);

...$response=array();
...$response["success"]=true;

...echo json_encode($response)
?>
```

- 식품등록에 사용된 FoodInsert.php 파일

3. 주요 소스 코드

```
Log.e("test", "test4");
foodInsertRequest foodInsertRequest = new foodInsertRequest(foodName, foodDate, division,
category, foodImage, Email, responseListener); //StringRequest를 받은 객체를 부름
RequestQueue queue = Volley.newRequestQueue(foodInsert.this);
queue.add(foodInsertRequest);
```

```
public class foodInsertRequest extends StringRequest {

    final static private String URL = "http://qkfldk197.dotheme.co.kr/foodInsert.php";
    private Map<String, String> parameters;

    public foodInsertRequest(String foodName, String foodDate, String division, String category,
String foodImage, String Email, Response.Listener<String> listener) { //생성자 호출
        super(Method.POST, URL, listener, null); //상속받은 StringRequest 의 생성자를 부른다. (어떤 방법
        으로 어디에, 예러시)
        parameters = new HashMap<>(); //해쉬맵 형태로 값을 받아옴
        parameters.put("foodName", foodName);
        parameters.put("foodDate", foodDate);
        parameters.put("division", division);
        parameters.put("category", category); //("태그", 실제데이터) 형태로 저장
        parameters.put("foodImage", foodImage);
        parameters.put("Email", Email);
    }
}
```

```
<?php
...$con=mysqli_connect("localhost", "qkfldk197", "hsi0206197", "qkfldk197");
...mysqli_set_charset($con, "utf8");

...$foodName=$_POST["foodName"];
...$foodDate=$_POST["foodDate"];
...$division=$_POST["division"];
...$category=$_POST["category"];
...$foodImage=$_POST["FoodImage"];
...$Email=$_POST["Email"];

...$statement=mysqli_prepare($con, "insert into Refrigerator values(?,?,?,?,?,?)");
...mysqli_stmt_bind_param($statement, "ssssss", $foodName, $foodDate, $division, $category, $foodImage,
    $Email);
...mysqli_stmt_execute($statement);

...$response=array();
...$response["success"]=true;

...echo json_encode($response)
?>
```

- 식품등록에 사용된 FoodInsert.php 파일

[Android] 카카오 계정으로 로그인하기

1. 프로젝트에 Kakao SDK 추가하기

카카오 로그인을 연동하기 위해서는 카카오에서 제공하는 SDK를 프로젝트에 추가해야 합니다. 카카오 디벨로퍼에서 SDK 라이브러리를 받아 사용해도 무관합니다만, 간단하게 Gradle 설정을 통해 추가해보도록 하겠습니다.

■ build.gradle (Project: -) 파일을 열어 다음과 같이 두 줄을 추가해주세요.

```
allprojects {  
    repositories {  
        google()  
        jcenter()  
        //kakao  
        mavenCentral()  
        maven { url 'http://devrepo.kakao.com:8088/nexus/content/groups/public/' }  
    }  
}
```

■ 다음으로 gradle.properties 파일을 열어 다음 두 줄을 추가해줍니다.

```
gradle.properties  
  
KAKAO_SDK_GROUP=com.kakao.sdk  
KAKAO_SDK_VERSION=1.1.7
```

여기서 SDK 버전은 최신버전을 확인하시고 넣어주시면 됩니다.
최신버전 확인은 카카오 디벨로퍼에서 확인 할 수 있습니다.
(<https://developers.kakao.com/docs/sdk>)

■ 다음으로 build.gradle(Module : app)의 dependencies 에 다음과 같이 추가해주세요.

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'

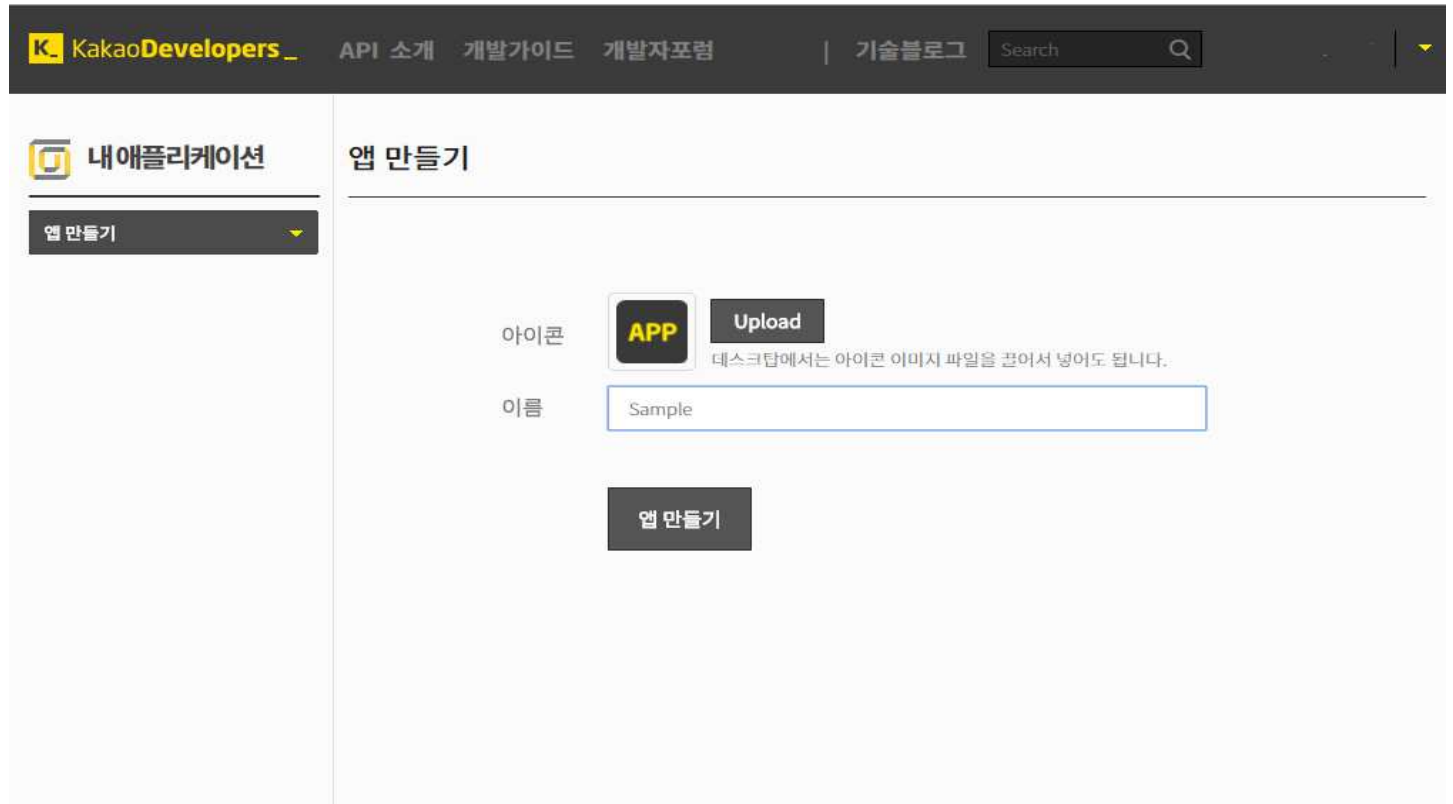
    // 카카오톡 SDK 추가
    compile group: project.KAKAO_SDK_GROUP, name: 'usermgmt', version:
    project.KAKAO_SDK_VERSION
}
```

2. App Key 발급 및 등록하기

연동을 위해서는 SDK외에도 App Key를 발급받아 프로젝트에 등록해주어야 합니다.

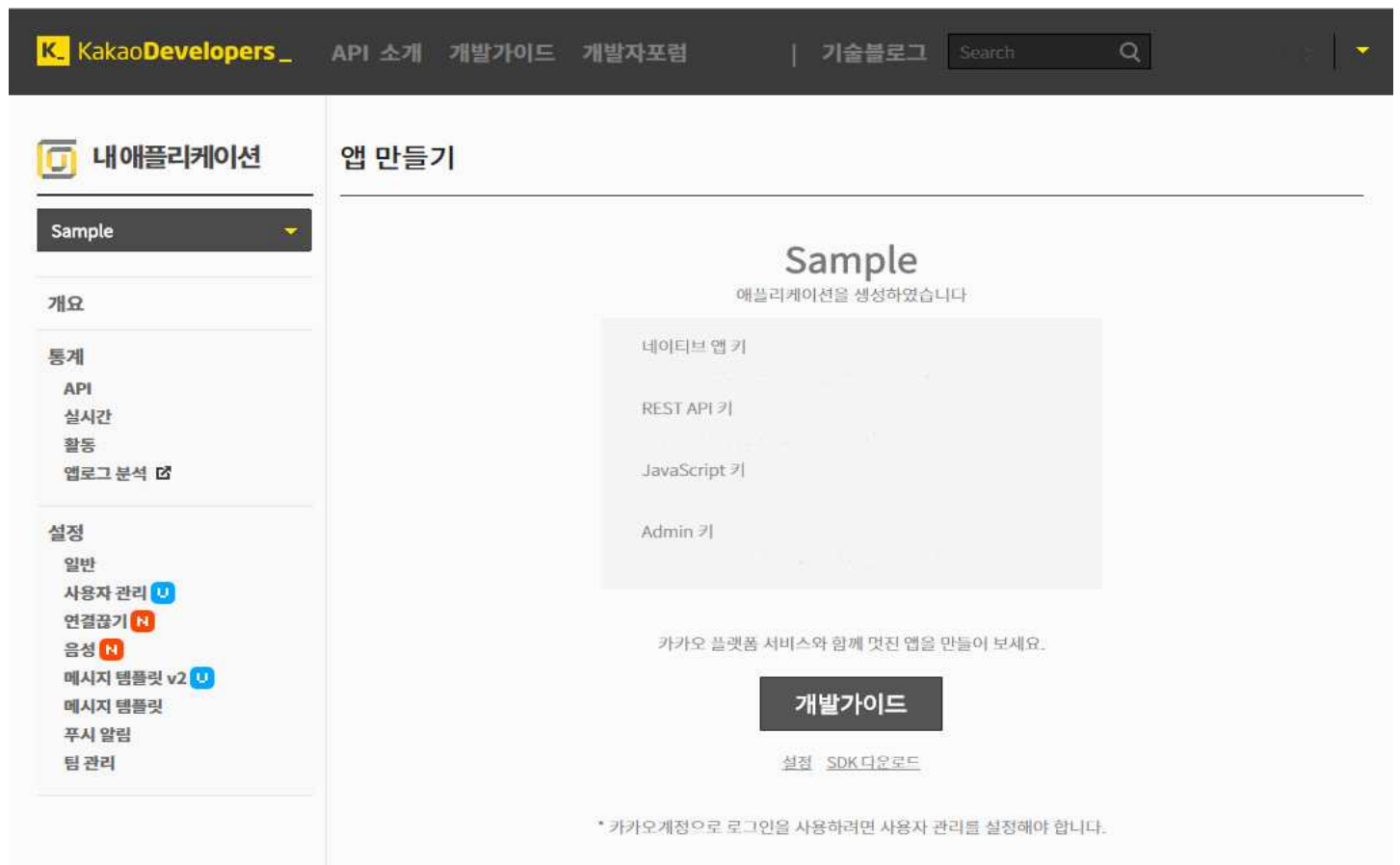
App Key는 카카오 디벨로퍼 사이트 (<https://developers.kakao.com/docs/sdk>)에서 발급 받으실 수 있습니다. 사이트에 접속하신 뒤에, 로그인을 하시고 나면 아래와 같은 화면이 나옵니다. 좌측에 보이는 앱 만들기 버튼을 클릭합니다.

The screenshot shows the Kakao Developers portal. The top navigation bar includes 'KakaoDevelopers', 'API 소개', '개발가이드', '개발자포럼', '기술블로그', and a search bar. The left sidebar has a menu with '내애플리케이션' (My Applications) and a sub-menu with '전체 애플리케이션' (All Applications) and '앱 만들기' (Create App), which is highlighted with a red box. The main content area shows '전체 애플리케이션 (4)' (All Applications (4)) and a '사용량' (Usage) section. The usage section includes '총 API 요청' (Total API Requests) with a value of 303 and '총 쿼터 사용량' (Total Quota Usage) with a value of 303. A bar chart shows usage over time, with a maximum value of 3,000,000. The bottom section shows '전월 사용량' (Previous Month Usage) with a value of 72.



앱 아이콘은 이후에도 변경이 가능하니, 걱정하지 않으셔도 됩니다.

앱 이름을 작성하신 후에, 확인을 눌러 앱을 만들면 다음과 같이 키가 발급됩니다.
(아래 그림의 키 값들은 보이지 않도록 지워져 있습니다.)



발급 받은 키 중에 저희가 사용할 것은 네이티브 앱 키 입니다. 키 값을 일단 복사해서 프로젝트에 붙여두겠습니다.

strings.xml

```
<resources>
<string name="app_name">KaKaoLogin</string>
<string name="kakao_app_key">자신의카카오네이티브앱키</string>
</resources>
```

다시 디벨로퍼 사이트 화면으로 돌아와서, 나머지 설정을 하겠습니다.
좌측 메뉴 중에 개요를 선택해주세요.

The screenshot shows the Kakao Developers portal interface. At the top, there's a navigation bar with the Kakao Developers logo and links for API 소개, 개발가이드, 개발자포럼, 기술블로그, and a search bar. The main content area is titled 'Sample' under the 'APP' category. On the left, there's a sidebar menu with options like '내 애플리케이션', '개요', '통계', 'API', '실시간', '활동', '앱로그 분석', '설정', '일반', '사용자 관리', '연결끊기', '음성', '메시지 템플릿 v2', '메시지 템플릿', '푸시 알림', and '팀 관리'. The '개요' (Overview) section is active, showing details for the 'Sample' app. It includes a '카카오 계정 로그인' link, a note about using Kakao account login, '앱 정보' (App Info) section with links for '앱 키' and '설정된 플랫폼', and a '통계' (Statistics) section showing 'API 요청수' (0) and '쿼터 사용량' (0). At the bottom, there are links for '카카오링크' and '앱로그 분석'.

내 애플리케이션 > 개요

개요 화면을 보시면 제일 상단에 '카카오 계정 로그인' 메뉴가 있습니다.

아래 설명이 잘 되어 있네요. '사용자 관리' 링크를 클릭하셔서 사용자 관리를 활성화 시키도록 하겠습니다.

활성화 시에 나타나는 옵션들은 필요에 따라 추가하시거나 빼주시면 됩니다.

개인정보 보호 항목에서 사용할 항목에 대해서는 수집목적을 반드시 작성하셔야 정상적으로 저장이 가능합니다.

KakaoDevelopers API 소개 개발가이드 개발자포럼 | 기술블로그 Search

내애플리케이션

Sample

개요

통계 API 실시간 활동 앱로그 분석

설정

일반 사용자 관리 연결공기 음성 메시지 템플릿 v2 메시지 템플릿 푸시 알림 팀 관리

APP Sample

사용자 관리 개발가이드: [iOS](#) [Android](#) [JavaScript](#) [REST API](#)

사용 ☒ ON

앱 연동 설정

자동가입 ☒ 로그인시 앱 자동 가입 ?

카카오 계정 연동 ☒ 카카오톡 프로필 1순위 ☒ 카카오톡스토리 프로필 2순위

동의항목 관리

로그인을 통해 해당 앱(서비스) 사용 시 사용자에게 보여주는 동의항목을 설정합니다.
처음 가입 시 최소한의 동의를 받고, 해당 기능 사용 시 추가로 동의받는 것이 가능합니다.
가입 시 보게 되는 동의항목은 필수항목과 선택항목으로 나뉘어집니다.
필수항목은 사용자가 반드시 동의해야만 가입을 완료할 수 있습니다.

개인정보 보호항목	Id	필수	가입시 선택	이용중 선택	사용안함
프로필 정보(닉네임/프로필 사진)	profile	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

앱 연결, 사용자 정보 요청, 카카오톡 프로필 요청, 카카오톡스토리 프로필 요청

수집목적

개요 > 사용자 관리

다시 개요 화면으로 돌아와서 이번에는 '앱 정보' 메뉴를 보겠습니다.

여기서 앱 이름과 아이콘 또는 각 키 값들을 재발급 받을 수 있습니다.

지금은 플랫폼을 추가해야 하기 때문에 아래 그림과 같이 플랫폼 메뉴의 플랫폼 추가 버튼을 눌러주세요.

KakaoDevelopers API 소개 개발가이드 개발자포럼 | 기술블로그 Search

내애플리케이션 Sample

개요

통계
API
실시간
활동
사용자
앱로그 분석

설정
일반
사용자 관리
연결끊기
음성
메시지 템플릿 v2
메시지 템플릿
푸시 알림
팀 관리

APP Sample

기본 정보

앱 키

네이티브 앱 키
REST API 키
JavaScript 키
Admin 키

플랫폼

Client Secret

플랫폼 추가

개요 > 사용자 관리

버튼을 누르면 아래와 같은 다이얼로그가 보이실겁니다.

위에 Android에 체크 해주시고 패키지명에 프로젝트의 패키지명을 작성해주시면 됩니다.

마켓 URL은 패키지명 작성 후에 자동으로 작성되어지니 신경 안쓰셔도 됩니다.

플랫폼 추가

☒ Android ☐ iOS ☐ 웹

패키지명 ?

rebuild.com.kakaotalk_login

마켓 URL ?

market://details?id=rebuild.com.kakaotalk_login

취소

추가

플랫폼 추가 다이얼로그

모두 작성 후에 추가 버튼을 누르시면 아래 그림과 같이 플랫폼이 추가 된 것을 확인 하실 수 있습니다.

The screenshot displays the KakaoDevelopers portal interface. On the left, there's a sidebar with navigation links like '내애플리케이션' (My Application) and '통계' (Statistics). The main content area shows the configuration for an application named 'Sample'. Under the '기본 정보' (Basic Information) section, various API keys are listed. The '플랫폼' (Platform) section shows 'Android' as the selected platform. Below this, there are input fields for '패키지명' (Package Name), '마켓 URL' (Market URL), and '키 해시' (Key Hash). The '키 해시' field is highlighted with a red rectangular box, indicating it is a required field that is currently empty. A red box also highlights the '필수 입력' (Required Input) label next to it. At the bottom, a pink notification box confirms '저장되었습니다.' (Saved).

등록 된 플랫폼을 보시면 키 해시 값이 비어있는 것을 확인 하실 수 있는데요, 키 해시가 등록된 앱에서만 SDK를 이용하여 API를 호출할 수 있습니다. 해시키는 다음과 같은 코드로 쉽게 구할수 있습니다.

☆--해쉬키 얻는 방법--☆

(안드로이드 아무 클래스에 아래 코드를 붙혀넣고 실행 후 LogCat에서 keyHash 검색)

```

import android.content.pm.PackageInfo;
import android.content.pm.PackageManager;
import android.content.pm.Signature;
import android.util.Base64;
import android.util.Log;

import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getHashKey();
    }
    private void getHashKey(){
        PackageInfo packageInfo = null;
        try {
            packageInfo =
getPackageManager().getPackageInfo(getPackageName(),
PackageManager.GET_SIGNATURES);
        } catch (PackageManager.NameNotFoundException e) {
            e.printStackTrace();
        }
        if (packageInfo == null)
            Log.e("KeyHash", "KeyHash:null");

        for (Signature signature : packageInfo.signatures) {
            try {
                MessageDigest md = MessageDigest.getInstance("SHA");
                md.update(signature.toByteArray());
                Log.d("KeyHash", Base64.encodeToString(md.digest(),
Base64.DEFAULT));
            } catch (NoSuchAlgorithmException e) {
                Log.e("KeyHash", "Unable to get MessageDigest.
signature=" + signature, e);
            }
        }
    }
}

```

얻어오신 키 해시 값을 플랫폼 정보의 키 해시 값에 추가 후 저장을 누르면 됩니다.

이것으로 카카오 디벨로퍼에 앱을 등록하는 부분은 끝났습니다.

그럼 아까 저장해 두었던 네이티브 앱 키를 프로젝트에 등록 시키기 위해 manifests.xml 파일을 열어 수정을 하도록 하겠습니다.

manifests.xml

```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="rebuild.com.kakaotalk_login">

    <!-- 인터넷 사용 권한 -->

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <!-- 카카오톡 메타데이터 -->

        <meta-data
            android:name="com.kakao.sdk.AppKey"
            android:value="@string/kakao_app_key" />

        <activity android:name=".LoginActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```


3. 프로젝트와 Kakao SDK 연결하기

Kakao SDK를 프로젝트에서 사용하기 위해서는 SDK와 앱을 연결해 주어야 합니다. 이번에는 그 역할을 해주는 Adapter를 작성하고 초기화 하는 부분을 살펴보겠습니다. Kakao SDK를 사용하기 위해서는 초기화가 필요합니다. 초기화는 GlobalApplication 공유 클래스를 만들어 앱 수준에서 관리하도록 하겠습니다.

```
GlobalApplication.java

import android.app.Activity;
import android.app.Application;

import com.kakao.auth.KakaoSDK;

public class GlobalApplication extends Application {
    private static volatile GlobalApplication obj = null;
    private static volatile Activity currentActivity = null;

    @Override
    public void onCreate() {
        super.onCreate();
        obj = this;
        KakaoSDK.init(new KakaoSDKAdapter());
    }

    public static GlobalApplication getGlobalApplicationContext() {
        return obj;
    }

    public static Activity getCurrentActivity() {
        return currentActivity;
    }

    // Activity가 올라올때마다 Activity의 onCreate에서 호출해줘야한다.
    public static void setCurrentActivity(Activity currentActivity) {
        GlobalApplication.currentActivity = currentActivity;
    }
}
```

GlobalApplication 객체를 공유하기 위해 디자인 패턴 중 싱글 톤 패턴을 이용하여 작성하였습니다. onCreate()에서는 Kakao SDK를 초기화하고 있습니다. KakaoSDKAdapter 클래스는 아직 작성하지 않았기 때문에 에러가 발생하는 것이 정상입니다.

작성을 완료하셨다면, manifests.xml에 추가하셔야 합니다.

manifests.xml

```
<application
    android:name=".GlobalApplication"
    ... >
```

이어서 KakaoSDKAdapter를 작성해 보도록 하겠습니다.

```

import android.app.Activity;
import android.content.Context;
import com.kakao.auth.ApprovalType;
import com.kakao.auth.AuthType;
import com.kakao.auth.IApplicationConfig;
import com.kakao.auth.ISessionConfig;
import com.kakao.auth.KakaoAdapter;

public class KakaoSDKAdapter extends KakaoAdapter {

    /**
     * 로그인을위해Session을생성하기위해필요한옵션을얻기위한abstract class.
     * 기본설정은KakaoAdapter에정의되어있으며, 설정변경이필요한경우상속해서사용할수있다.
     */
    @Override
    public ISessionConfig getSessionConfig() {
        return new ISessionConfig() {
            @Override
            public AuthType[] getAuthTypes() {
                return new AuthType[] {AuthType.KAKAO_LOGIN_ALL};
            }
        }
    }

    // 로그인시인증받을타입을지정한다. 지정하지않을시가능한모든옵션이지정된다.

    //1.KAKAO_TALK : kakaotalk으로login을하고싶을때지정.
    //2.KAKAO_STORY : kakaostory으로login을하고싶을때지정.
    //3.KAKAO_ACCOUNT : 웹뷰Dialog를통해카카오계정연결을제공하고싶을경우지정.
    //4.KAKAO_TALK_EXCLUDE_NATIVE_LOGIN : 카카오톡으로만로그인을유도하고싶으면서계정이없을때계정생성을위한버튼도같이제공을하고싶다면지정.KAKAO_TALK과중복지정불가.
    //5.KAKAO_LOGIN_ALL : 모든로그인방식을사용하고싶을때지정.

    @Override
    public boolean isUsingWebViewTimer() {
        return false;
    }

    // SDK 로그인시사용되는WebView에서pause와resume시에Timer를설정하여CPU소모를절약한다.
    // true 를리턴할경우webview로그인을사용하는화면서모든webview에onPause와onResume시에Timer를설정해주어야한다. 지정하지않을시false로설정된다.

    @Override
    public ApprovalType getApprovalType() {
        return ApprovalType.INDIVIDUAL;
    }

    //일반사용자가아닌Kakao와제휴된앱에서사용되는값으로, 값을채워주지않을경우ApprovalType.INDIVIDUAL값을사용하게된다.

    @Override
    public boolean isSaveFormData() {
        return true;
    }

    // Kakao SDK 에서사용되는WebView에서email 입력폼에서data를save할지여부를결정한다. Default true.
    };

    //Application이가지고있는정보를얻기위한interface.
    @Override
    public IApplicationConfig getApplicationConfig() {
        return new IApplicationConfig() {
            @Override
            public Activity getTopActivity() {
                return GlobalApplication.getCurrentActivity();
            }
        }
    }

    // 현재최상단에위치하고있는Activity. topActivity가아니거나ApplicationContext를넣는다면SDK내에서의Dialog Popup등이동작하지않을수있습니다.
    @Override
    public Context getApplicationContext() {
        return GlobalApplication.getGlobalApplicationContext();
    }
    };
}

```

KakaoSDKAdapter 클래스는 KakaoAdpater를 상속 받아 구현하였습니다. 내부에는 ISessionConfig와 IApplicationConfig를 설정하기 위한 인터페이스로 구성되어있습니다.

ISessionConfig는 Session에 대한 옵션 값들을 설정 할 수 있습니다. 여기서 Session은 로그인 객체를 유지 시키는 객체로 Access Token을 관리하는 역할을 합니다.

IApplicationConfig는 애플리케이션이 가지고 있는 정보를 전달하는 역할을 합니다. 각각의 자세한 설명은 주석을 참고 하시면 될 듯합니다.

로그인 타입의 경우, 각각의 옵션을 설정하신 뒤에 실행시켜 보시고 원하시는 타입을 사용하시면 됩니다.

4. Login 버튼 구현하기

로그인 버튼을 구현할 때, 가장 손쉬운 방법은 Kakao에서 제공하는 버튼을 사용하는 경우입니다.

activity_login.xml

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    tools:context="rebuild.com.kakaotalk_login.Activity.LoginActivity">
    <com.kakao.usermgmt.LoginButton
        android:id="@+id/btn_kakao_login"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
```

이렇게만 버튼을 넣어주면 끝입니다. 아직 로그인 결과에 따른 콜백에 대해 설명은 하지 않았지만, 기본적으로 버튼만 달아주면 알아서 다 해줘서 굉장히 편리합니다.

하지만,

저 버튼은 디자인이 정해져있기 때문에 앱의 디자인 컨셉과 맞지 않을 수도 있습니다. 이런 경우, 원하는 디자인을 사용하면서도 카카오톡 로그인을 할 수 있는 몇 가지 방법이 있습니다.

첫번째로 카카오에서 제공하는 버튼을 안보이게 한 뒤에, 커스텀으로 제작한 버튼을 누를 경우, 클릭 효과를 주는 performClick()메서드를 이용하는 방법입니다.

activity_login.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    tools:context="rebuild.com.kakaotalk_login.Activity.LoginActivity">
    <Button
        android:id="@+id/btn_custom_login"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="카카오 로그인" />
    <com.kakao.usermgmt.LoginButton
        android:id="@+id/btn_kakao_login"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:visibility="gone" />
</LinearLayout>
```

위에서는 간단하게 버튼을 두었습니다. 원하시는 형태에 따라 레이아웃을 구현하시면 되겠습니다. 기존의 카카오버튼은 보이지 않게 숨겼습니다.

그리고 Activity로 와서,

LoginActivity.java

```
import android.content.Context;
import android.content.pm.PackageInfo;
import android.content.pm.PackageManager;
import android.content.pm.Signature;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Base64;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import java.security.MessageDigest;

public class LoginActivity extends AppCompatActivity {
    private Context mContext;

    private Button btn_custom_login;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        mContext = getApplicationContext();

        btn_custom_login = (Button) findViewById(R.id.btn_custom_login);
        btn_custom_login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Session session = Session.getCurrentSession();
                session.addCallback(new SessionCallback());
                session.open(AuthType.KAKAO_LOGIN_ALL, LoginActivity.this);
            }
        });
    }
}
```


위와 같이, 커스텀 버튼의 클릭 리스너에서 Session을 오픈하도록 한 줄만 추가하시면 됩니다.

간단하게 로그인버튼을 구현하는 부분을 살펴보았는데요, 로그인 버튼의 경우, 카카오에서 권장하는 디자인이 있으니 확인 후에 작성하시는 것이 좋습니다.

5. Session Callback 클래스 구현하기

```
import android.util.Log;

import com.kakao.auth.ISessionCallback;

import com.kakao.network.ErrorResult;

import com.kakao.usermgmt.UserManagement;

import com.kakao.usermgmt.callback.MeResponseCallback;

import com.kakao.usermgmt.response.model.UserProfile;

import com.kakao.util.exception.KakaoException;

public class SessionCallback implements ISessionCallback {

    // 로그인에 성공한 상태

    @Override

    public void onSessionOpened() {

        requestMe()

    }

    // 로그인에 실패한 상태

    @Override

    public void onSessionOpenFailed(KakaoException exception) {

        Log.e("SessionCallback :: ", "onSessionOpenFailed : " + exception.getMessage());

    }

}
```

```
// 사용자 정보 요청

public void requestMe() {

    // 사용자정보 요청 결과에 대한 Callback

    UserManagement.requestMe(new MeResponseCallback() {

        // 세션 오픈 실패. 세션이 삭제된 경우,

        @Override

        public void onSessionClosed(ErrorResult errorResult) {

            Log.e("SessionCallback :: ", "onSessionClosed : " +
errorResult.getErrorMessage());

        }

        // 회원이 아닌 경우,

        @Override

        public void onNotSignedUp() {

            Log.e("SessionCallback :: ", "onNotSignedUp");

        }

        // 사용자정보 요청에 성공한 경우,

        @Override

        public void onSuccess(UserProfile userProfile) {

            Log.e("SessionCallback :: ", "onSuccess");

            String nickname = userProfile.getNickname();

            String email = userProfile.getEmail();

            String profileImagePath = userProfile.getProfileImagePath();

            String thumbnailPath = userProfile.getThumbnailImagePath();

            String UUID = userProfile.getUUID();

            long id = userProfile.getId();

            Log.e("Profile : ", nickname + "");

            Log.e("Profile : ", email + "");

            Log.e("Profile : ", profileImagePath + "");

            Log.e("Profile : ", thumbnailPath + "");

            Log.e("Profile : ", UUID + "");

        }

    });
}
```

```

        Log.e("Profile : ", id + "");
    }

// 사용자 정보 요청 실패
@Override
public void onFailure(ErrorResult errorResult) {
    Log.e("SessionCallback :: ", "onFailure : " +
errorResult.getErrorMessage());
}

});
}
}

```

ISessionCallback은 Session의 상태 변경에 따른 Callback을 받도록 구성되어있습니다. onSessionOpend()는 Access Token을 성공적으로 발급 받은 상태로 즉, 로그인에 성공 시에 진입합니다. 이후, 로그인 된 사용자의 정보를 요청하여 제공 받을 수 있습니다. onSessionOpenFailed()는 로그인에 실패한 경우로, 여기서 실패라 함은 네트워크 에러 또는 일반적인 에러 사항에 대한 비정상적인 동작에 의한 에러 등을 말합니다.

6. 로그아웃 하기

로그인을 했으니, 로그아웃 요청에 대해서도 알아보겠습니다.
로그아웃 요청은 매우 간단합니다.

```

UserManagement.requestLogout(new LogoutResponseCallback() {
    @Override
    public void onCompleteLogout() {
        Intent intent = new Intent(MainActivity.this, LoginActivity.class);
        startActivity(intent);
    }
});

```

4. 참고자료 및 출처

- <https://wingsnote.com/149> - 안드로이드스플래시 화면
- <https://namjackson.tistory.com/15> - QR코드, 바코드 리더라이브러리Zxing 1.사용법
- <https://developers.kakao.com/apps/315231/created> - 카카오앱개발자
- <https://re-build.tistory.com/9> - 안드로이드 카카오연동
- <https://kgh940525.tistory.com/entry> -안드로이드-카카오톡Kakaotalk-로그인-연동하기
- <https://coding-factory.tistory.com/126> - 문자열자르기
- <https://mainia.tistory.com/2022> - AlertDialog 선택목록창과 옵션창 띄우기
- <https://inma.tistory.com/64> - ImageView에 이미지주소로 이미지지정하기
- <https://black-jin0427.tistory.com/120> - 카메라와 갤러리에서 이미지가져오기
- <https://superwony.tistory.com/78> - QR코드 리더기만들기, ZXing 라이브러리사용법
- <https://yonoo88.tistory.com/1011> - json파일 파싱하기
- <https://webnautes.tistory.com/1189> - 데이터베이스에데이터저장및json형식파싱