

응용수학 캡스톤디자인 2

오픈 API를 이용하여 가계부 어플 제작하기

응용수학과 2016007701 김가연

응용수학과 2016007838 심유라

응용수학과 2016007874 이나영

응용수학과 2016007974 한혜주

CONTENTS TABLE



1. 오픈 बैं킹 및 앱 소개

API를 이용한
오픈 बैं킹에 대해 알아보자.



2. 앱 디자인 코드

앱 로딩 화면 및 디자인을
삽입해보자.



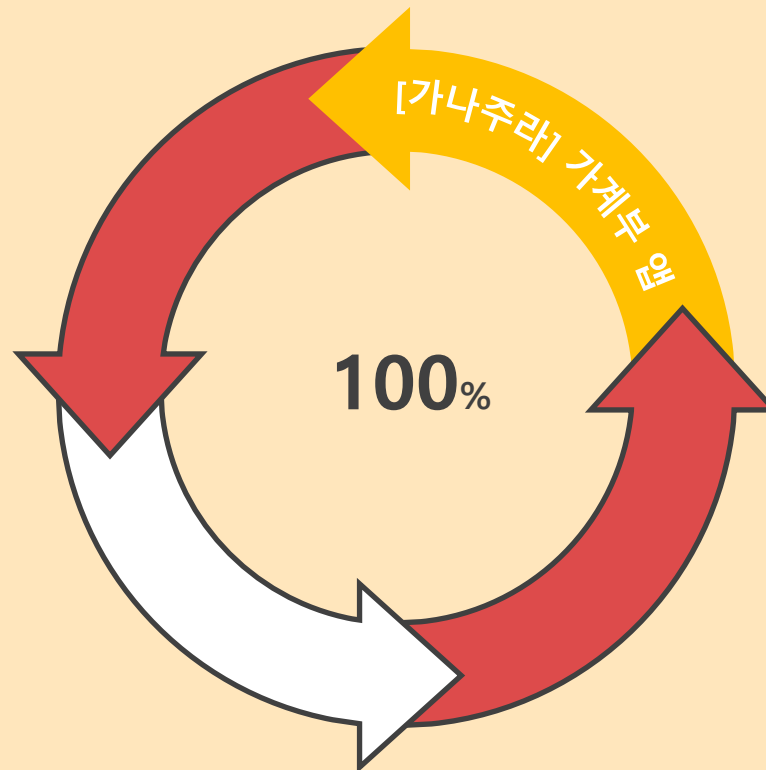
4. 앱 실행

앱이 오류 없이 실행되는지
확인해보자.



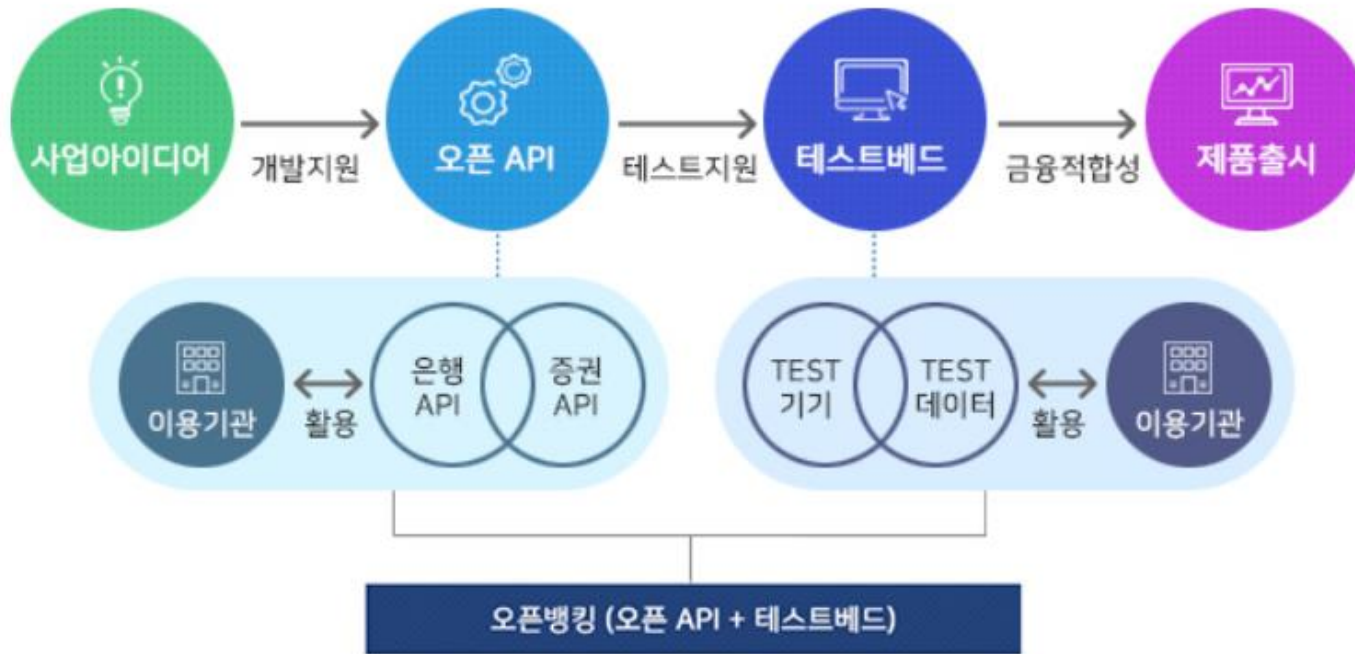
3. 앱 기능 코드

가계부 앱에 필요한 기능을
추가해보자.



1. 오픈 बैं킹 및 앱 소개

오픈 बैं킹이란?



오픈 API

: 핀테크기업이 직접 응용프로그램과 서비스를 개발할 수 있도록 공개된 프로그램 도구

테스트베드

: 개발된 서비스가 금융전산망에서 정상적으로 작동하는지 시험해 볼 수 있는 인프라

1. 오픈 बैं킹 및 앱 소개

오픈 बैं킹이란?



이용기관은 참신한 아이디어와 기술을 가지고 금융 분야에 진출하여 **다양한 핀테크 서비스를 제공할 수 있게 된다.**

참가은행은 이용기관이 제공하는 다양한 서비스를 통하여 신규 고객과 만날 수 있게 됨으로써 **새로운 수익기회**를 얻을 수 있다.

핀테크 서비스를 이용하는 **고객**은 금융에 대한 **접근이 쉬워지고 선택권이 넓어지는 효과**가 생기게 된다.

1. 오픈 बैं킹 및 앱 소개

오픈 बैं킹 공동업무 오픈 API 명세서(v233_1) 첨부파일

오픈 बैं킹 테스트 베드

<https://developers.open-platform.or.kr/>



오픈 API

개발가이드

개발가이드

자료실

시작하기

[최신본] 오픈뱅킹공동업무 오픈API 명세서(v2.3.2_1)

1. 오픈 बैंकिंग 및 앱 소개

01



ANDROID STUDIO

구글이 안드로이드 앱 개발을
위해 만든 통합 개발환경

02



POSTMAN

POSTMAN
API 테스트를 위해 사용

03

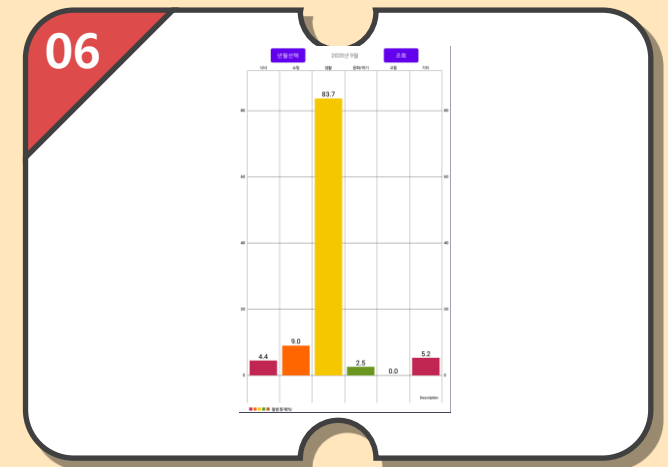
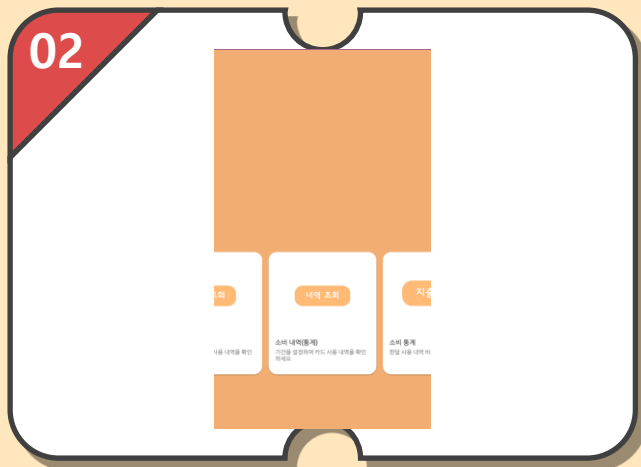
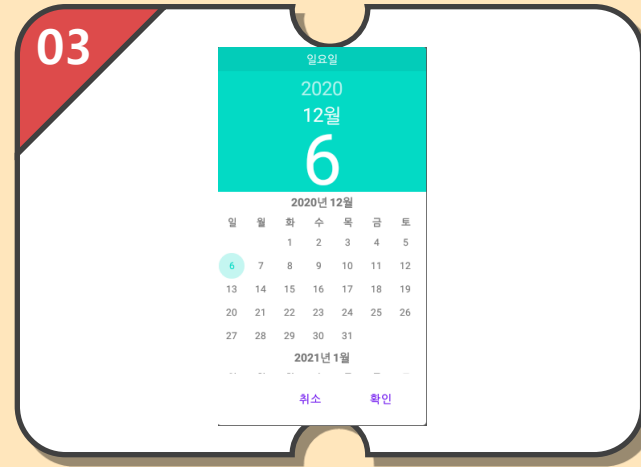
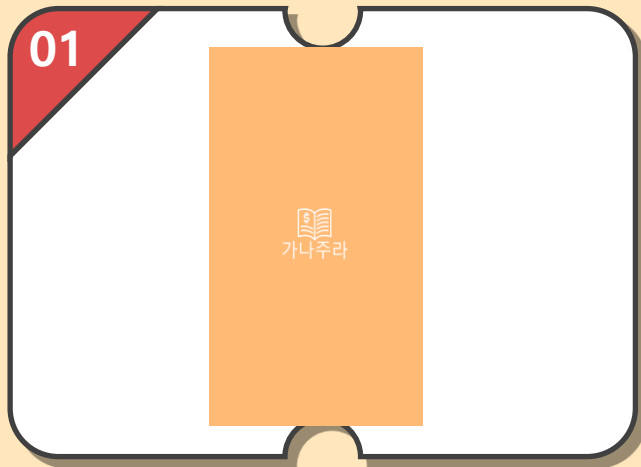


GitHub

GITHUB
코드 공유

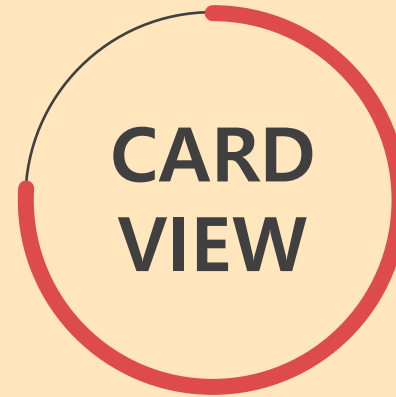
1. 오픈 बैं킹 및 앱 소개

앱 실행 화면 구성



2. 앱 디자인 코드

2. 앱 디자인 코드



2. 앱 디자인 코드

SPLASH
생성

SPLASH ACTIVITY CODE

```
public class SplashActivity extends Activity{
    @Override
    public void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash);
        startLoading();
    }
    private void startLoading(){
        Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() { finish(); }
        }, delayMillis: 2000);
    }
}
```

2. 앱 디자인 코드

SPLASH
생성

SPLASH LAYOUT CODE

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    android:gravity="center"
    android:layout_weight="1"
    android:background="@drawable/logo_2">
</LinearLayout>
```



CARD
VIEW

2. 앱 디자인 코드

BUILD GRADLE CODE

```
dependencies {  
  
    implementation 'androidx.appcompat:appcompat:1.2.0'  
    implementation 'com.google.android.material:material:1.2.1'  
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'  
    testImplementation 'junit:junit:4.+'  
    implementation 'androidx.cardview:cardview:1.0.0'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'  
  
    implementation 'com.github.PhilJay:MPAndroidChart:v2.2.4'  
    implementation 'com.github.bumptech.glide:glide:4.9.0'  
    annotationProcessor 'com.github.bumptech.glide:compiler:4.9.0'
```

LAYOUT 위젯, 형식 가져오기

2. 앱 디자인 코드

MODEL CLASS CODE

```
public class Model {  
  
    private int image;  
    private String title;  
    private String desc;  
  
    public Model(int image, String title, String desc) {  
        this.image = image;  
        this.title = title;  
        this.desc = desc;  
    }  
  
    public int getImage() { return image; }  
  
    public void setImage(int image) { this.image = image; }  
  
    public String getTitle() { return title; }  
  
    public void setTitle(String title) { this.title = title; }  
  
    public String getDesc() { return desc; }  
  
    public void setDesc(String desc) { this.desc = desc; }  
}
```

CARDVIEW에 보여주고자 하는 요소를
CLASS에 정의

2. 앱 디자인 코드

CARD
VIEW

DATA LIST를 입력 받아 각 데이터 항목에
해당하는 VIEW를 생성

ADAPTER CLASS CODE

```
public class Adapter extends PagerAdapter {  
    public static final int sub = 1001;  
    private List<Model> models;  
    private LayoutInflater inflater;  
    private Context context;  
  
    public Adapter(List<Model> models, Context context){  
        this.models = models;  
        this.context = context;  
    }  
}
```

화면에 표시될 VIEW로부터 위젯에 대한 참조 획득

MODELS에서 POSITION에 위치한 데이터
참조 획득

'ITEM' LAYOUT을 INFLATE하여 VIEW참조
획득

ADAPTER CLASS CODE

```
public Object instantiateItem(@NonNull ViewGroup container, final int position){  
    inflater = LayoutInflater.from(context);  
    View view = inflater.inflate(R.layout.item, container, attachToRoot: false);  
  
    ImageView imageView;  
    TextView title, desc;  
  
    imageView = view.findViewById(R.id.image);  
    title = view.findViewById(R.id.title);  
    desc = view.findViewById(R.id.desc);  
  
    imageView.setImageResource(models.get(position).getImage());  
    title.setText(models.get(position).getTitle());  
    desc.setText(models.get(position).getDesc());  
}
```

2. 앱 디자인 코드

ADAPTER CLASS CODE

```
view.setOnClickListener(new View.OnClickListener(){  
    @Override  
    public void onClick(View v){  
        if(position==0){  
            Intent intent = new Intent(context, SubActivity_list.class);  
            context.startActivity(intent);  
        }  
        else if(position==1){  
            Intent intent = new Intent(context, SubActivity.class);  
            context.startActivity(intent);  
        }  
        else if(position==2){  
            Intent intent = new Intent(context, SubActivity2.class);  
            context.startActivity(intent);  
        }  
    }  
});  
container.addView(view, index: 0);  
return view;  
}
```

DATA LIST를 입력 받아 각 DATA항목에
해당하는 VIEW 생성

내역조회, 내역통계, CHART CARD
VIEW(BUTTON)에 맞는 ACTIVITY 연결

2. 앱 디자인 코드

MAIN ACTIVITY CODE

```
public class MainActivity extends AppCompatActivity {

    public static final int sub = 1001;

    ViewPager viewPager;
    Adapter adapter;

    List<Model> models;
    Integer[] colors = null;
    ArgbEvaluator argbEvaluator = new ArgbEvaluator();

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

        ActionBar actionBar = getSupportActionBar();
        actionBar.hide();

        Intent intent = new Intent( packageContext: this, SplashActivity.class);
        startActivity(intent);

        models = new ArrayList<>();
        models.add(new Model(R.drawable.list_image, title: "소비 내역(list)", desc: "기간을 설정하여 카드 사용 내역을 확인하세요"));
        models.add(new Model(R.drawable.list_image, title: "소비 내역(통계)", desc: "기간을 설정하여 카드 사용 내역을 확인하세요"));
        models.add(new Model(R.drawable.statistic_image, title: "소비 통계", desc: "한달 사용 내역 비율을 알아보세요"));
    }
}
```

앱이 처음 실행될 때, 나타나는 PAGE

앱이 실행될 때 SPLASH 화면 출력

ADAPTER CLASS에 참조되는 DATA

2. 앱 디자인 코드

MAIN ACTIVITY CODE

```
viewPager.setOnPageChangeListener(new ViewPager.OnPageChangeListener() {  
    @Override  
    public void onPageScrolled(int position, float positionOffset, int positionOffsetPixels) {  
        if (position < (adapter.getCount() - 1) && position < (colors.length - 1)) {  
            viewPager.setBackgroundColor((Integer) argbEvaluator.evaluate(positionOffset, colors[position], colors[position + 1]))  
        }  
  
        else {  
            viewPager.setBackgroundColor(colors[colors.length - 1]);  
        }  
    }  
  
    @Override  
    public void onPageSelected(int position) {  
    }  
  
    @Override  
    public void onPageScrollStateChanged(int state) {  
    }  
});
```

앱이 처음 실행될 때, 나타나는 PAGE

VIEWPAGER의 이동에 따라 선택한 CARD
VIEW가 나오도록 한다.

2. 앱 디자인 코드

MAIN ACTIVITY LAYOUT CODE

```
<androidx.viewpager.widget.ViewPager
    android:id="@+id/viewPager"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_centerInParent="true"

    android:clipToPadding="false"
    android:overScrollMode="never">

</androidx.viewpager.widget.ViewPager>
```

MAIN ACTIVITY에 CARD VIEW 위젯이 나타나도록
설정

2. 앱 디자인 코드

MAIN ACTIVITY LAYOUT CODE

```
<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:cardCornerRadius="20dp"
    android:layout_margin="8dp">
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="300dp"
    >

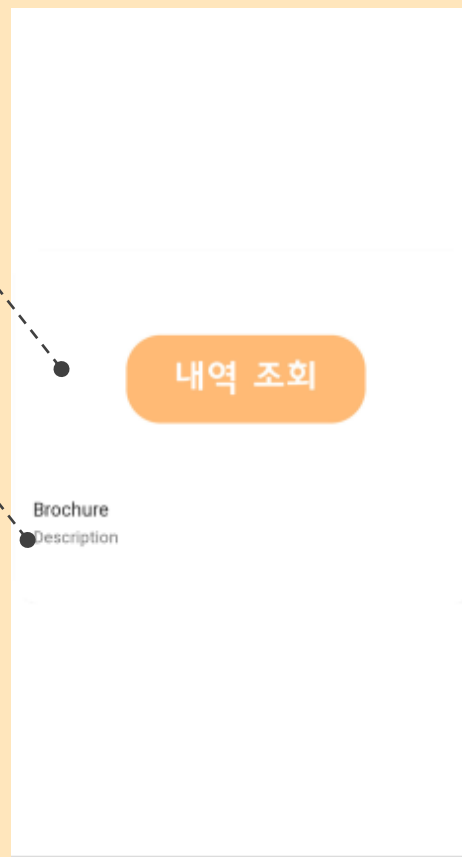
        <ImageView
            android:id="@+id/image"
            android:layout_width="match_parent"
            android:layout_height="200dp"
            android:scaleType="centerCrop"
            android:src="@drawable/list_image" />

        <TextView
            android:id="@+id/title"
            android:textColor="#262626"
            android:layout_below="@id/image"
            android:layout_marginTop="10dp"
            android:layout_marginLeft="16dp"
```

LIST IMAGE로 설정한 IMAGE를 불러와 CARD
IMAGE 구성

내역조회와 같이 CAR VIEW TEXT 구성

ADAPTER CLASS에 참조되는 CARD VIEW LAYOUT
형식



3. 앱 기능 코드

3. 앱 기능 코드

NEW ACTIVITY & NEW ACTIVITY-LIST

날짜 별로 조회하면 내역을 JSON형태로 받아
와 다시 원하는 모양으로 변환하는 코드

CLASS

TASK1 : 날짜 FODD : '식비' 합계

TASK2 : 입/출금 SHOP : '쇼핑' 합계

TASK3 : 구분 LIFE : '생활' 합계

TASK4 : 내역 CULTURE : '문화/여가' 합계

TASK5 : 금액 TRAFFIC : '교통' 합계

OTHER : '기타' 합계

SUBACTIVITY & SUBACTIVITY-LIST

조회한 내역을 화면으로 보여주는 코드

START DATE		END DATE			
2020년 7월 2일		~	2020년 12월 6일		조회
식비	128790	쇼핑	56030	생활	0
문화/여가	24000	교통	0	기타	0
날짜	입/출금	구분	내역	금액	
20200705	출금	식비	저녁식사	49000	
20200719	출금	문화/여가	영화	24000	
20200819	입금	이자	이자	1803	
20200821	출금	식비	저녁식사	43200	
20200907	입금	급여	연구비	27900	
20200919	입금	이자	이자	920	
20200921	출금	식비	편의점	4590	
20201004	출금	쇼핑	쇼핑	56030	
20201007	출금	식비	파스타	32000	

3. 앱 기능 코드

```
public static class Food extends AsyncTask<String, Void, String> {  
    private String str, receiveMsg;  
    String start = ((SubActivity) SubActivity.context).s_start;  
    String end = ((SubActivity) SubActivity.context).s_end;  
  
    @Override  
    protected String doInBackground(String... params) {  
        Random rand = new Random();  
        String numStr = "";  
  
        for (int i = 0; i < 9; i++) {  
            String ran = Integer.toString(rand.nextInt( bound: 9));  
            numStr += ran;  
        }  
        String header = "Bearer" + "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiIxMTAwNzY0MDIyIiwic2NvcGUlOlsiaW5xdWlyeSI6ImxvZ2luIiwiaWF0IjoiMTYwMjE0ODQ0InQ=";  
        URL url = null;  
        try {  
            url = new URL( spec: "https://testapi.openbanking.or.kr/v2.0/account/transaction_list?fin_num?bank_tran_id=T991663280U"  
                + numStr + "&fintech_use_num=199166328057887392605371&inquiry_type=A&inquiry_base=D&from_date=" + start +  
                "&to_date=" + end + "&sort_order=D&tran_dtime=20201006025351");  
  
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();  
            conn.setRequestMethod("GET");  
            conn.setRequestProperty("Authorization", header);
```

DOING BACKGROUND

: url 설정해 json을 string 형태로 반환

3. 앱 기능 코드

NEW ACTIVITY CODE

```
public String listjsonParser(String jsonString) {  
    String tran_date = null;  
    String tran_time = null;  
    String inout_type = null;  
    String tran_type = null;  
    String print_content = null;  
    String tran_amt = null;  
    String after_balance_amt = null;  
    String branch_name = null;  
    String result = "";  
  
    int sum = 0;  
    String[] a = new String[100];  
    String[] b = new String[100];  
    String[] d = new String[100];  
}
```

LISTJSONPARSER

: 조회한 날짜 내에서 '식비' 구분 합계

NEW ACTIVITY CODE

```
try{  
    JSONArray jarray = new JSONObject(jsonString).getJSONArray( name: "res_list");  
    for(int i=0;i<jarray.length();i++){  
        JSONObject jobject = jarray.getJSONObject(i);  
  
        tran_date = jobject.optString( name: "tran_date");  
        tran_time = jobject.optString( name: "tran_time");  
        inout_type = jobject.optString( name: "inout_type");  
        tran_type = jobject.optString( name: "tran_type");  
        print_content = jobject.optString( name: "print_content");  
        tran_amt = jobject.optString( name: "tran_amt");  
        after_balance_amt = jobject.optString( name: "after_balance_amt");  
        branch_name = jobject.optString( name: "branch_name");  
  
        a[i] = tran_type;  
        b[i] = tran_amt;  
        d[i] = tran_date;  
  
        if(Integer.parseInt(start)<=Integer.parseInt(d[i])&&Integer.parseInt(end)>=Integer.parseInt(d[i])){  
            if(a[i].contains("식비")){  
                sum += Integer.parseInt(b[i]);  
            }  
        }  
    }  
}
```

3. 앱 기능 코드

NEW ACTIVITY CODE

```
public String listjsonParser(String jsonString){
    String tran_date = null;
    String tran_time = null;
    String inout_type = null;
    String tran_type = null;
    String print_content = null;
    String tran_amt = null;
    String after_balance_amt = null;
    String branch_name = null;
    String result = "";

    String[] a = new String[100];
    String[] d = new String[100];
```

LISTJSONPARSER

: 조회한 날짜 내에서 '내역'을 STRING
형태로 반환

NEW ACTIVITY CODE

```
String[] arraysum = new String[1];
try{
    JSONArray jarray = new JSONObject(jsonString).getJSONArray( name: "res_list");
    for(int i=0;i<jarray.length();i++){
        HashMap map = new HashMap<>();
        JSONObject jobject = jarray.getJSONObject(i);

        tran_date = jobject.optString( name: "tran_date");
        tran_time = jobject.optString( name: "tran_time");
        inout_type = jobject.optString( name: "inout_type");
        tran_type = jobject.optString( name: "tran_type");
        print_content = jobject.optString( name: "print_content");
        tran_amt = jobject.optString( name: "tran_amt");
        after_balance_amt = jobject.optString( name: "after_balance_amt");
        branch_name = jobject.optString( name: "branch_name");

        a[i]=print_content;
        d[i]=tran_date;

        if(Integer.parseInt(start)<=Integer.parseInt(d[i])&&Integer.parseInt(end)>=Integer.parseInt(d[i])){
            result += a[i] + "\n";
        }
    }
}
```


3. 앱 기능 코드

SUBACTIVITY CODE

```
public void InitializeListener_e() {
    callbackMethod_e = new DatePickerDialog.OnDateSetListener() {
        @Override

        public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {
            textView_Date_e.setText(year + "년 " + (monthOfYear + 1) + "월 " + dayOfMonth + "일");

            e_year = String.valueOf(year);
            e_month = String.valueOf(monthOfYear + 1);
            e_day = String.valueOf(dayOfMonth);
            /*if(length==1){
                e_day = "0"+Integer.toString(dayOfMonth);
            }else{
                e_day= Integer.toString(dayOfMonth);
            }*/
            if (e_day.length() == 1) {
                e_day = "0" + e_day;
            }
            if (e_month.length() == 1) {
                e_month = "0" + e_month;
            }
            s_end = e_year + e_month + e_day;
        }
    };
}
```

START & END DATE 선택 시 그 값을 변수로 저장

START DATE	END DATE
2020년 4월 3일	~ 2020년 12월 5일
<input type="button" value="조회"/>	

3. 앱 기능 코드

SUBACTIVITY CODE

```
@Override
public void onClick(View v) {

    if (Integer.parseInt(s_start) > Integer.parseInt(s_end)) {
        Toast myToast = Toast.makeText(getApplicationContext(), text: "시작 날짜를 마지막 날짜보다 \n전으로 선택하십시오.",
            Toast.LENGTH_SHORT);
        myToast.show();
    } else if ((Integer.parseInt(s_end) > Integer.parseInt(p_start)) || (Integer.parseInt(s_start) > Integer.parseInt(p_start))) {
        Toast myToast = Toast.makeText(getApplicationContext(), text: "오늘 이전의 날짜를 선택하십시오.", Toast.LENGTH_SHORT);
        myToast.show();
    } else {
```

시작 날짜를 마지막 날짜보다
전으로 선택하십시오.

오늘 이전의 날짜를 선택하십시오.

LISTJONEPARSER

: 날짜 선택 시 오류이면 TOAST 띄우기

3. 앱 기능 코드

SUB ACTIVITY CODE

```
try {  
    resultText1 = new NewActivity.Task1().execute().get();  
    resultText2 = new NewActivity.Task2().execute().get();  
    resultText3 = new NewActivity.Task3().execute().get();  
    resultText4 = new NewActivity.Task4().execute().get();  
    resultText5 = new NewActivity.Task5().execute().get();  
  
    resultFoodSum = new NewActivity.Food().execute().get();  
    foodsum = Integer.parseInt(resultFoodSum);  
    resultShopSum = new NewActivity.Shop().execute().get();  
    shopsom = Integer.parseInt(resultShopSum);  
    resultLifeSum = new NewActivity.Life().execute().get();  
    lifesum = Integer.parseInt(resultLifeSum);  
    resultCultureSum = new NewActivity.Culture().execute().get();  
    culturesum = Integer.parseInt(resultCultureSum);  
    resultTrafficSum = new NewActivity.Traffic().execute().get();  
    trafficsum = Integer.parseInt(resultTrafficSum);  
    resultOtherSum = new NewActivity.Other().execute().get();  
    othersum = Integer.parseInt(resultOtherSum);
```

SUB ACTIVITY CODE

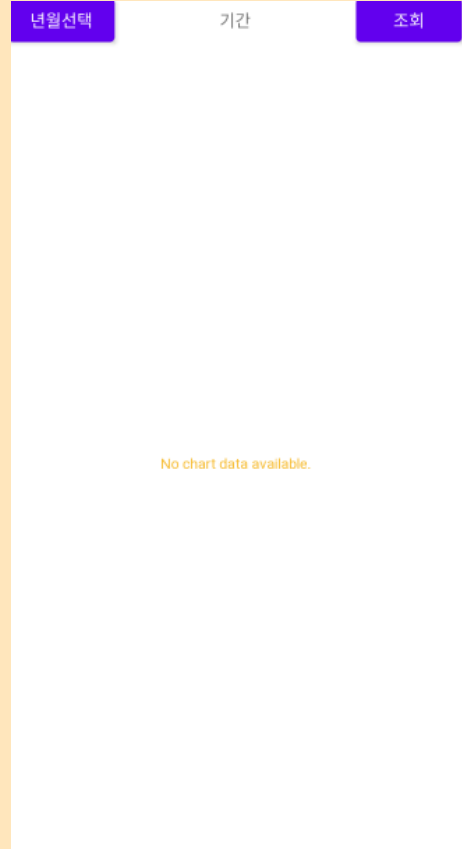
```
list_trandate.setText(resultText1);  
list_inout_type.setText(resultText2);  
list_tran_type.setText(resultText3);  
list_print_content.setText(resultText4);  
list_tran_amt.setText(resultText5);  
food_sum.setText(resultFoodSum);  
shop_sum.setText(resultShopSum);  
life_sum.setText(resultLifeSum);  
culture_sum.setText(resultCultureSum);  
traffic_sum.setText(resultTrafficSum);  
other_sum.setText(resultOtherSum);
```

NEWACTIVITY 에서 받아온 값을
변수로 저장 후 TEXTVIEW에 나
타내기

3. 앱 기능 코드

LAYOUT-FRAG2 CODE

```
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context=".SubActivity2">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="center">
    <Button
        android:id="@+id/btn_year_month_picker"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="년월선택"/>
    <TextView
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:text="기간"
        android:gravity="center"
        android:id="@+id/select_year_month" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="조회"
        android:id="@+id/check" />
</LinearLayout>
<com.github.mikephil.charting.charts.BarChart
    android:id="@+id/barchart"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>
```



화면
구성

2. 앱 기능 코드

년월선택

기간

조회

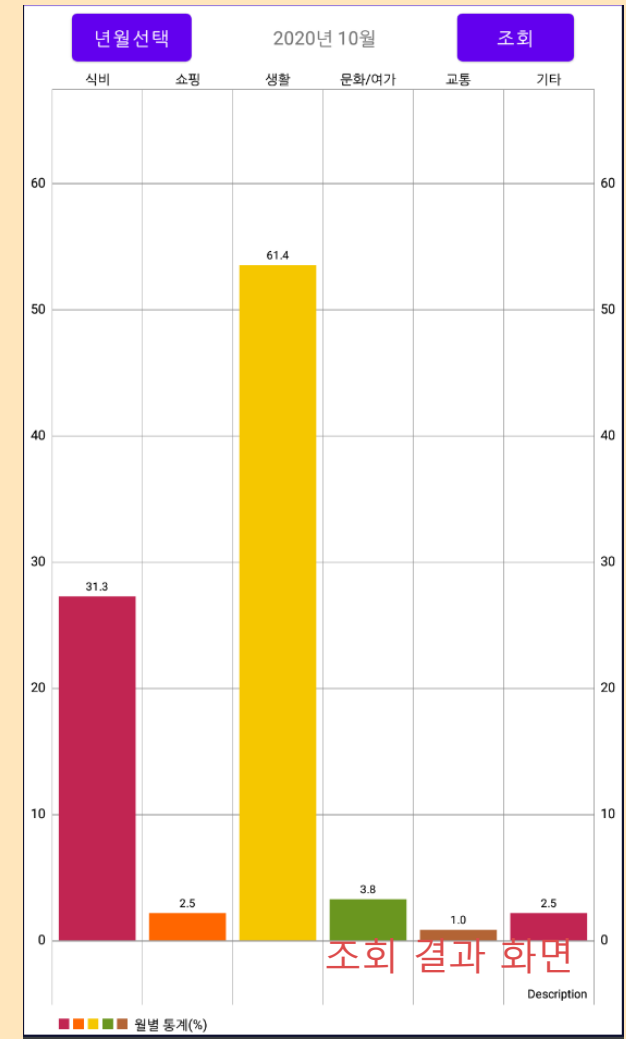
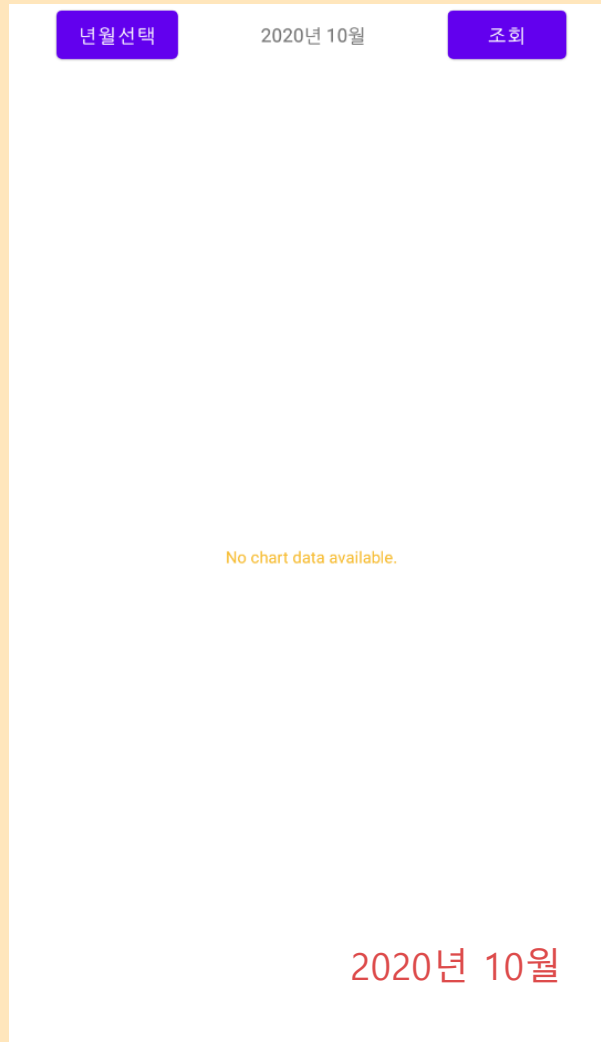
월별 합계

2019	9
2020	10
2021	11

취소

확인

년월 선택



3. 앱 기능 코드

SUBACTIVITY2에서 사용할 변수 선언

선택한 월 정보 저장 변수

소비 내역 별로 금액 합계 저장 변수

NEW ACTIVITY CODE

```
public class SubActivity2 extends AppCompatActivity {  
    public TextView select_year_month;  
    public String select_ym;  
    public String select_year;  
    public String select_month;  
    public static Context context;  
    public float month_food;  
    public float month_shop;  
    public float month_life;  
    public float month_culture;  
    public float month_traffic;  
    public float month_other;  
  
    Button btnYearMonthPicker;  
    Button btn_confirm;  
    //number picker 값받아오기  
    private TextView textView_Date;  
    private TextView textView_Date_e;  
    private DatePickerDialog.OnDateSetListener callbackMethod_e;
```

3. 앱 기능 코드

년월 선택 버튼 클릭 시 실행 코드

NEW ACTIVITY CODE

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.frag2);
    ActionBar actionBar = getSupportActionBar();
    actionBar.hide();

    btnYearMonthPicker = findViewById(R.id.btn_year_month_picker);
    btnYearMonthPicker.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View view) {
            MyYearMonthPicker pd = new MyYearMonthPicker();
            pd.setListener(d);
            pd.show(getSupportFragmentManager(), tag: "YearMonthPickerTest");
        }
    });
}
```

년월 선택 버튼 클릭 시
MyYearMonthPicker 실행

NEW ACTIVITY CODE

```
DatePickerDialog.OnDateSetListener d = new DatePickerDialog.OnDateSetListener() {
    @Override
    public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {
        select_year_month.setText(year + "년 " + month + "월");

        Log.d(tag: "YearMonthPickerTest", msg: "year= " + year + ", month= " + month);

        select_year = String.valueOf(year);
        select_month = String.valueOf(month);
        if (select_month.length() == 1) {
            select_month = "0" + select_month;
        }
        select_ym = select_year + select_month;
        Log.d(tag: "select_ym", msg: "select_ym = " + select_ym);
    }
};
```

받은 년월을 변수의 형태에 맞게 저장

3. 앱 기능 코드

조회 버튼 클릭 시 실행 코드

NEW ACTIVITY2에서 각각의 CLASS 실행 후
결과 값을 변수에 저장

값을 이용하여 내역별 비율을 계산
하여 변수에 저장

NEW ACTIVITY CODE

```
Button btn = (Button) findViewById(R.id.check); // 조회
select_year_month = (TextView) findViewById(R.id.select_year_month);
btn.setOnClickListener(new Button.OnClickListener() {
    @Override
    public void onClick(View v) {
        String resulttotalsum = "";
        String resultmonthfood = "";
        String resultmonthshop = "";
        String resultmonthlife = "";
        String resultmonthculture = "";
        String resultmonthtraffic = "";
        String resultmonthother = "";
        try {
            resulttotalsum = new NewActivity2.MonthTotal().execute().get();
            resultmonthfood = new NewActivity2.MonthFood().execute().get();
            resultmonthshop = new NewActivity2.MonthShop().execute().get();
            resultmonthlife = new NewActivity2.MonthLife().execute().get();
            resultmonthculture = new NewActivity2.MonthCulture().execute().get();
            resultmonthtraffic = new NewActivity2.MonthTraffic().execute().get();
            resultmonthother = new NewActivity2.MonthOther().execute().get();

            month_food = Float.parseFloat(resultmonthfood)/Float.parseFloat(resulttotalsum)*100;
            month_shop = Float.parseFloat(resultmonthshop)/Float.parseFloat(resulttotalsum)*100;
            month_life = Float.parseFloat(resultmonthlife)/Float.parseFloat(resulttotalsum)*100;
            month_culture = Float.parseFloat(resultmonthculture)/Float.parseFloat(resulttotalsum)*100;
            month_traffic = Float.parseFloat(resultmonthtraffic)/Float.parseFloat(resulttotalsum)*100;
            month_other = Float.parseFloat(resultmonthother)/Float.parseFloat(resulttotalsum)*100;
```


3. 앱 기능 코드

년월 선택 버튼 클릭 시 실행 코드

NEW ACTIVITY CODE

```
try {
    JSONArray jarray = new JSONObject(jsonString).getJSONArray( name: "res_list");
    for (int i = 0; i < jarray.length(); i++) {
        HashMap map = new HashMap<>();
        JSONObject jobject = jarray.getJSONObject(i);

        tran_date = jobject.optString( name: "tran_date");
        tran_time = jobject.optString( name: "tran_time");
        inout_type = jobject.optString( name: "inout_type");
        tran_type = jobject.optString( name: "tran_type");
        print_content = jobject.optString( name: "print_content");
        tran_amt = jobject.optString( name: "tran_amt");
        after_balance_amt = jobject.optString( name: "after_balance_amt");
        branch_name = jobject.optString( name: "branch_name");

        a[i] = tran_date;
        b[i] = inout_type;
        c[i] = tran_amt;

        if(a[i].contains(select_ym)&&b[i].contains("출금")){
            sum += Integer.parseInt(c[i]);
        }
    }
    result = Integer.toString(sum);
} catch (JSONException e) {
    e.printStackTrace();
}
return result;
```

년월 선택 버튼 클릭 시
MyYearMonthPicker 실행

NEW ACTIVITY CODE

```
try {
    JSONArray jarray = new JSONObject(jsonString).getJSONArray( name: "res_list");
    for (int i = 0; i < jarray.length(); i++) {
        HashMap map = new HashMap<>();
        JSONObject jobject = jarray.getJSONObject(i);

        tran_date = jobject.optString( name: "tran_date");
        tran_time = jobject.optString( name: "tran_time");
        inout_type = jobject.optString( name: "inout_type");
        tran_type = jobject.optString( name: "tran_type");
        print_content = jobject.optString( name: "print_content");
        tran_amt = jobject.optString( name: "tran_amt");
        after_balance_amt = jobject.optString( name: "after_balance_amt");
        branch_name = jobject.optString( name: "branch_name");

        a[i] = tran_date;
        b[i] = tran_type;
        c[i] = tran_amt;

        if(a[i].contains(select_ym)&&b[i].contains("식비")){
            sum += Integer.parseInt(c[i]);
        }
    }
    result = Integer.toString(sum);
} catch (JSONException e) {
    e.printStackTrace();
}
return result;
```

받은 년월을 변수의 형태에 맞게 저장

3. 앱 기능 코드

년월 선택 버튼 클릭 시 실행 코드

NEW ACTIVITY CODE

```
allprojects {
    repositories {
        google()
        jcenter()
        maven {url "https://jitpack.io"}
    }
}
```

```
dependencies {
    implementation 'androidx.appcompat:appcompat:1.2.0'
    implementation 'com.google.android.material:material:1.2.1'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    testImplementation 'junit:junit:4.+'
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'
    implementation 'com.github.PhilJay:MPAndroidChart:v2.2.4'
```

년월 선택 버튼 클릭 시
MyYearMonthPicker 실행

NEW ACTIVITY CODE

```
BarChart chart = findViewById(R.id.barchart);
ArrayList NoOfEmp = new ArrayList();

NoOfEmp.add(new BarEntry(month_food, xIndex: 0));
NoOfEmp.add(new BarEntry(month_shop, xIndex: 1));
NoOfEmp.add(new BarEntry(month_life, xIndex: 2));
NoOfEmp.add(new BarEntry(month_culture, xIndex: 3));
NoOfEmp.add(new BarEntry(month_traffic, xIndex: 4));
NoOfEmp.add(new BarEntry(month_other, xIndex: 5));

ArrayList day = new ArrayList();

day.add("식비");
day.add("쇼핑");
day.add("생활");
day.add("문화/여가");
day.add("교통");
day.add("기타");

BarDataSet bardataset = new BarDataSet(NoOfEmp, label: "월별 통계(%");
chart.animateY( durationMillis: 5000);
BarData data = new BarData(day, bardataset); // MPAndroidChart v3.X 오류 발생
bardataset.setColors(ColorTemplate.COLORFUL_COLORS);
chart.setData(data);

} catch (InterruptedException e) {
    e.printStackTrace();
} catch (ExecutionException e) {
    e.printStackTrace();
}
```

받아온 년월을 변수의 형태에 맞게 저장

4. 앱 실행