

android

오픈소스 라이브러리를 활용
한 어플리케이션 개발

오픈소스란 무엇인가?

Open-source software (OSS) is computer software with its source code made available with a license in which the copyright holder provides the rights to study, change, and distribute the software to **anyone** and for **any purpose**. Open-source software may be developed in a collaborative public manner. According to scientists who studied it, open-source software is a prominent example of open collaboration.

오픈소스 소프트웨어, OSS라고도 한다. 소프트웨어의 설계도에 해당하는 소스코드를 인터넷 등을 통하여 무상으로 공개하여 누구나 그 소프트웨어를 개량하고, 이것을 재배포할 수 있도록 하는 것 또는 그런 소프트웨어를 말한다

LICENSE

- ▶ GNU General Public License(GPL) 2.0
- ▶ GNU Lesser GPL(LGPL) 2.1
- ▶ Berkeley Software Distribution(BSD) License
- ▶ Apache License
- ▶ Mozilla Public License(MPL)
- ▶ MIT License

RETROFIT 2

Square, Inc에서 공개한 HTTP REST API 구현을 위한 오픈소스 라이브러리

- ▶ 다양한 HTTP Client를 플러그인 형태로 변경하여 사용 가능
- ▶ 다양한 Serialization, Deserialization 플러그인 제공
- ▶ 콜백을 활용한 비동기 통신 구현 가능
- ▶ RxJava 지원

SQUARE, INC에서 공개한 HTTP REST API 구현을 위한 오픈소스 라이브러리

Module

```
compile 'com.squareup.retrofit2:retrofit:2.3.0'
```

```
compile 'com.squareup.retrofit2:converter-gson:2.3.0'
```


Retrofit 2

```
public interface CommentCallService {

    @GET("{fullPath}")
    Call<List<CommentDTO>> comments(@HeaderMap Map<String, String> headers, @Path(value = "fullPath", encoded = true) String fullPath);

    @GET("/repos/{user}/{repository}/issues/{number}/comments")
    Call<List<CommentDTO>> comments(@Path("user") String user, @Path("repository") String repository, @Path("number") int number);

    @GET("/repos/{user}/{repository}/issues/{number}/comments/{id}")
    Call<CommentDTO> comment(@Path("user") String user, @Path("repository") String repository, @Path("number") int number, @Path("id") int id);

    @POST("/repos/{user}/{repository}/issues/{number}/comments")
    Call<CommentDTO> createComment(
        @Path("user") String user,
        @Path("repository") String repository,
        @Path("number") int number,
        @HeaderMap Map<String, String> headers,
        @Body Map<String, String> body);

    @PATCH("/repos/{user}/{repository}/issues/{number}/comments/{id}")
    Call<CommentDTO> updateComment(
        @Path("user") String user,
        @Path("repository") String repository,
        @Path("number") int number,
        @Path("id") int id,
        @HeaderMap Map<String, String> headers,
        @Body Map<String, String> body);

    @DELETE("/repos/{user}/{repository}/issues/{number}/comments/{id}")
    Call deleteComment(
        @Path("user") String user,
        @Path("repository") String repository,
        @Path("number") int number,
        @Path("id") int id,
        @HeaderMap Map<String, String> headers);
}
```

Retrofit 2

SQUARE, INC에서 공개한 HTTP REST API 구현을 위한 오픈소스 라이브러리

```
Retrofit retrofit = new Retrofit.Builder()
    .baseUrl("https://api.github.com/")
    .addConverterFactory(GsonConverterFactory.create())
    .build();
```

```
IssueCallService service = retrofit.create(IssueCallService.class);
```

```
Call<List<IssueDTO>> issues = service.issues("csi111", "LibrarySample");
```

GLIDE

Glide is a fast and efficient open source media management and image loading framework for Android that wraps media decoding, memory and disk caching, and resource pooling into a simple and easy to

BUMP 앱을 만든 BUMPTECH사가 구글에 인수되며 오픈된 이미지 로딩 라이브러리

- ▶ 사용하기 편리한 API
- ▶ 다양한 데이터 모델 지원(content://, file://, http://, android.resource://)
- ▶ GIF 이미지 로딩 지원
- ▶ Memory, Data Cache 처리
- ▶ Bitmap 재활용

GLIDE

BUMP 앱을 만든 BUMPTECH사가 구글에 인수되며 오픈된 이미지 로딩 라이브러리

Module

```
compile 'com.github.bumptech.glide:glide:3.6.1'
```

//Option

```
compile 'jp.wasabeef:glide-transformations:2.0.2'
```

```
compile 'jp.co.cyberagent.android.gpuimage:gpuimage-library:1.4.1'
```

```
static class CommentViewHolder extends RecyclerView.ViewHolder {
    LayoutCommentItemBinding layoutCommentItemBinding;

    public CommentViewHolder(LayoutCommentItemBinding commentItemBinding) {
        super(commentItemBinding.itemComment);
        this.layoutCommentItemBinding = commentItemBinding;
    }

    void bind(CommentItemViewModel viewModel) {
        layoutCommentItemBinding.setItemViewModel(viewModel);
        Glide.with(itemView.getContext())
            .load(viewModel.getProfileThumbnailUrl())
            .fitCenter()
            .placeholder(R.drawable.ic_person_black_24dp)
            .bitmapTransform(new CropCircleTransformation(itemView.getContext()))
            .into(layoutCommentItemBinding.profileThumbnailImageView);
    }
}
```

REALM

REALM이라는 이름의 새로운 모바일 데이터베이스로써 SQLITE를 대체할 수 있는 라이브러리

- ▶ Sqlite보다 뛰어난 성능
- ▶ 객체 자체가 실시간성으로 동기화 지원
(스레드나 프로세스별 변경시 알림이 가능)
- ▶ 데이터 암호화 지원(AES256)
- ▶ 리액티브 아키텍처를 지향

REALM

REALM이라는 이름의 새로운 모바일 데이터베이스로써 SQLITE를 대체할 수 있는 라이브러리
Project

```
buildscript {  
    repositories {  
        jcenter()  
    }  
    dependencies {  
        classpath "io.realm:realm-gradle-plugin:3.4.0"  
    }  
}
```

REALM

REALM이라는 이름의 새로운 모바일 데이터베이스로써 SQLITE를 대체할 수 있는 라이브러리
Module

```
apply plugin: 'com.android.application'
apply plugin: 'realm-android'

android {
    ...
}
```

REALM

```
public class Counter extends RealmObject {
    public static final String FIELD_COUNT = "count";
    private static AtomicInteger INTEGER_COUNTER = new AtomicInteger(0);
    @PrimaryKey
    private int count;

    public int getCount() {
        return count;
    }

    public String getCountString() {
        return Integer.toString(count);
    }

    // create() & delete() needs to be called inside a transaction.
    static void create(Realm realm) {
        create(realm, false);
    }

    static void create(Realm realm, boolean randomlyInsert) {
        Parent parent = realm.where(Parent.class).findFirst();
        RealmList<Counter> counters = parent.getCounterList();
        Counter counter = realm.createObject(Counter.class, increment());
        if (randomlyInsert && counters.size() > 0) {
            Random rand = new Random();
            counters.listIterator(rand.nextInt(counters.size())).add(counter);
        } else {
            counters.add(counter);
        }
    }

    static void delete(Realm realm, long id) {
        Counter counter = realm.where(Counter.class).equalTo(FIELD_COUNT, id).findFirst();
        // Otherwise it has been deleted already.
        if (counter != null) {
            counter.deleteFromRealm();
        }
    }

    private static int increment() {
        return INTEGER_COUNTER.getAndIncrement();
    }
}
```

REALM

```
public class DataHelper {

    public static void randomAddItemAsync(Realm realm) {
        realm.executeTransactionAsync(new Realm.Transaction() {
            @Override
            public void execute(Realm realm) {
                for (int i = 0; i < 3; i++) {
                    Counter.create(realm, true);
                }
            }
        });
    }

    public static void addItemAsync(Realm realm) {
        realm.executeTransactionAsync(new Realm.Transaction() {
            @Override
            public void execute(Realm realm) {
                Counter.create(realm);
            }
        });
    }

    public static void deleteItemAsync(Realm realm, final long id) {
        realm.executeTransactionAsync(new Realm.Transaction() {
            @Override
            public void execute(Realm realm) {
                Counter.delete(realm, id);
            }
        });
    }

    public static void deleteItemsAsync(Realm realm, Collection<Integer> ids) {
        // Create an new array to avoid concurrency problem.
        final Integer[] idsToDelete = new Integer[ids.size()];
        ids.toArray(idsToDelete);
        realm.executeTransactionAsync(new Realm.Transaction() {
            @Override
            public void execute(Realm realm) {
                for (Integer id : idsToDelete) {
                    Counter.delete(realm, id);
                }
            }
        });
    }
}
```

PARCELER

ENTITY를 손쉽게 PARCELABLE 인터페이스를 사용하는 객체로 만들어주는 라이브러리

- ▶ 다양한 Attribute Type을 지원
(Map, Set, ObservableField, SparseArray등)
- ▶ Parcelable 인터페이스를 사용하는 객체를 만들기 위한 보일러플레이트를 줄여줌
- ▶ 다양한 POJO베이스의 라이브러리와 호환 가능(GSON, Realm, SimpleXML등)

PARCELER

ENTITY를 손쉽게 PARCELABLE 인터페이스를 사용하는 객체로 만들어주는 라이브러리

Module

```
compile 'org.parceler:parceler-api:1.1.9'
```

```
annotationProcessor 'org.parceler:parceler:1.1.9'
```


PARCELER

```
@Parcel(Parcel.Serialization.BEAN)
public class Example {

    private String name;
    private int age;
    private boolean enabled;

    @ParcelConstructor
    public Example(String name, int age) {
        this.name = name;
        this.age = age;
    }

    public String getName() { return name; }
    public void setName(String name) { this.name = name; }

    public int getAge() { return age; }
    public void setAge(int age) { this.age = age; }

    public boolean isEnabled() { return enabled; }
    public void setEnabled(boolean enabled) { this.enabled = enabled; }
}
```

PARCELER

```
Parcelable wrapped = Parcels.wrap(new Example("Andy", 42));  
Example example = Parcels.unwrap(wrapped);  
example.getName(); // Andy  
example.getAge(); // 42
```

```
Bundle bundle = new Bundle();  
bundle.putParcelable("example", Parcels.wrap(example));
```

```
Parcelable listParcelable = Parcels.wrap(new ArrayList<Example>());  
Parcelable mapParcelable = Parcels.wrap(new HashMap<String, Example>());
```

EVENTBUS

컴포넌트간의 다양한 이벤트를 손쉽게 전달 할 수 있는 라이브러리

- ▶ Subscribe형태로 간단하게 이벤트를 전달 받을 수 있음
- ▶ Priority, Thread등을 지정하여 이벤트를 Subscribe 가능
- ▶ 컴포넌트간의 Sender/Receiver형태로 이루어진 커플링을 끊어줄 수 있음

컴포넌트간의 다양한 이벤트를 손쉽게 전달 할 수 있는 라이브러리

Module

```
compile 'org.greenrobot:eventbus:3.0.0'
```

컴포넌트간의 다양한 이벤트를 손쉽게 전달 할 수 있는 라이브러리

```
public static class MessageEvent {  
    /* Additional fields if needed */  
    List<IssueItemViewModel> issueItemViewModelList;  
}
```

```
@Subscribe(threadMode = ThreadMode.MAIN)  
public void onMessageEvent(MessageEvent event) {  
    /* Do something */  
}
```

컴포넌트간의 다양한 이벤트를 손쉽게 전달 할 수 있는 라이브러리

```
@Override
public void onStart() {
    super.onStart();
    EventBus.getDefault().register(this);
}

@Override
public void onStop() {
    super.onStop();
    EventBus.getDefault().unregister(this);
}
```

```
EventBus.getDefault().post(new MessageEvent());
```

DATABINDING

DATABINDING

애플리케이션 로직과 레이아웃간의 데이터를 바인딩을 쉽게 도와주는 라이브러리

- ▶ Google에서 오픈한 안드로이드 라이브러리
- ▶ 데이터 및 이벤트(리스너) 바인딩을 지원
- ▶ Observable을 통해 데이터 동기화가 가능
- ▶ Expression Language 지원(+, -, %, &&, ||, ==, >, <, >>, >>>, instanceof 등)

DATABINDING

애플리케이션 로직과 레이아웃간의 데이터를 바인딩을 쉽게 도와주는 라이브러리

Module

```
android {  
    ...  
  
    dataBinding {  
        enabled = true  
    }  
}
```

DATABINDING

@Nullable

@Override

```
public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup
    container, @Nullable Bundle savedInstanceState) {
    fragmentIssuesBinding = DataBindingUtil.inflate(inflater, R.layout.fragment_issues, container, false);
    return fragmentIssuesBinding.getRoot();
}
```

@Override

```
public void onViewCreated(View view, @Nullable Bundle savedInstanceState) {
    super.onViewCreated(view, savedInstanceState);
    fragmentIssuesBinding.issuesRecyclerView.setLayoutManager(new IssuesRecyclerViewLayoutManager(getContext()));
    fragmentIssuesBinding.issuesRecyclerView.addItemDecoration(new DividerItemDecoration(getActivity(),
    DividerItemDecoration.VERTICAL));
    fragmentIssuesBinding.issuesRecyclerView.setAdapter(mIssuesAdapter);
    fragmentIssuesBinding.scrollChildSwipeRefreshLayout.setOnRefreshListener(new
    SwipeRefreshLayout.OnRefreshListener() {
        @Override
        public void onRefresh() {
            issuesViewModel.refreshIssues();
        }
    });
    issuesViewModel.fetchIssues();
}
```

DATABINDING

```
<?xml version="1.0" encoding="utf-8"?>
<layout xmlns:android="http://schemas.android.com/apk/res/android">
    <data>
        <variable
            name="itemViewModel"
            type="com.robinhood.librarysample.ui.issues.viewmodel.IssueItemViewModel" />
    </data>

    <LinearLayout
        android:id="@+id/item_issue"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="@{itemViewModel::onItemClick}"
        android:orientation="horizontal"
        android:gravity="center_vertical"
        android:padding="5dp">

        .....

        <TextView
            android:id="@+id/id_textView"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@{itemViewModel.issueIdText}"
            android:textColor="@color/colorPrimary"
            android:textSize="14sp" />

        <TextView
            android:id="@+id/title_textView"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@{itemViewModel.titleText}"
            android:textColor="@color/colorPrimary"
            android:textSize="18sp" />

        .....
    </layout>
```

GUAVA

Google Core Libraries For Java

- ▶ 구글에서 만든 아주 유용한 유틸리티 라이브러리
- ▶ 컬렉션 관련 초기화 및 다양한 유틸리티를 제공(합,차,교집합등을 구하기 쉽게 API 제공)
- ▶ 멀티밸류맵, 바이맵등을 제공
- ▶ 제한된 함수형 스타일의 프로그래밍을 제공(filter, map등)
- ▶ Hashcode, equals, Comparator등을 손쉽게 구현할 수 있도록 API 제공

Google Core Libraries For Java

Module

```
compile 'com.google.guava:guava:22.0-android'
```

GUAVA

```
List names = new ArrayList<>>();

names.add("조승우");

names.add("배두나");

names.add("이준혁");

Joiner.on(", ").join(names); // test1, test2, test3
```

```
Multimap<String, Issue> issues= HashMultimap.create();

issues.put("LibrarySample", new Issue(0, "설치 안됨"));

issues.put("LibrarySample", new Issue(1, "앱 Crash"));

issues.put("LibrarySample", new Issue(2, "이미지 로딩 실패"));

Collection<Issue> issueCollection = issues.get("LibrarySample");
```


GUAVA

```
@Override  
public int hashCode() {  
    return Objects.hashCode(mIssues);  
}
```

```
if (count <= 0) {  
    throw new IllegalArgumentException("Count must be positive: " + count);  
}
```

```
checkArgument(count > 0, "Count must be positive: %s", count);
```

GUAVA

```
Set<String> nameSet = Collections.unmodifiableSet(new LinkedHashSet<>(Arrays.asList("조승우", "배두나", "이준혁")));  
Set<String> guavaNameSet = ImmutableSet.of("조승우", "배두나", "이준혁");
```

```
Map<String, Integer> map = new LinkedHashMap<>();  
map.put("조승우", 1);  
map.put("배두나", 2);  
map.put("이준혁", 3);  
Map<String, Integer> unModifiedMap = Collections.unmodifiableMap(map);
```

```
Map<String, Integer> guavaMap = ImmutableMap.<String, Integer>builder()  
    .put("조승우", 1).put("배두나", 2).put("이준혁", 3)  
    .build();
```

```
Map<String, Integer> guavaMap2 = ImmutableMap.of("조승우", 1, "배두나", 2, "이준혁", 3);
```

LEAKCANARY

Square, Inc에서 공개한 메모리 릭을 검출하는 오픈소스 라이브러리

- ▶ Activity Leak(Destroy 이후에도 GC 되지 않는 현상)을 잡을 수 있음
- ▶ 객체의 Reference를 추적하여 메모리 릭을 검출할 수 있음
- ▶ HProf파일을 저장, Memory leak에 대한 추적 및 분석 가능

Square, Inc에서 공개한 메모리 리크를 검출하는 오픈소스 라이브러리

Module

```
debugCompile 'com.squareup.leakcanary:leakcanary-android:1.5.1'
```

```
releaseCompile 'com.squareup.leakcanary:leakcanary-android-no-op:1.5.1'
```

```
testCompile 'com.squareup.leakcanary:leakcanary-android-no-op:1.5.1'
```

Square, Inc에서 공개한 메모리 릭을 검출하는 오픈소스 라이브러리

```
public class SampleApplication extends Application {  
  
    @Override  
    public void onCreate() {  
        super.onCreate();  
        if (LeakCanary.isInAnalyzerProcess(this)) {  
            // This process is dedicated to LeakCanary for heap analysis.  
            // You should not init your app in this process.  
            return;  
        }  
        LeakCanary.install(this);  
    }  
}
```

STETHO

Facebook에서 만든 안드로이드를 위한 디버그 브릿지 라이브러리

- ▶ 크롬 브라우저를 통해 다양한 디버깅을 할 수 있는 기능을 제공
- ▶ Network Inspection
- ▶ Database Inspection
- ▶ View Hierarchy
- ▶ SharedPreferences dump
- ▶ Javascript console

Facebook에서 만든 안드로이드를 위한 디버그 브릿지 라이브러리

Module

```
compile 'com.facebook.stetho:stetho:1.5.0'
```

```
compile 'com.facebook.stetho:stetho-okhttp3:1.5.0'
```

```
or
```

```
compile 'com.facebook.stetho:stetho-okhttp:1.5.0'
```

```
or
```

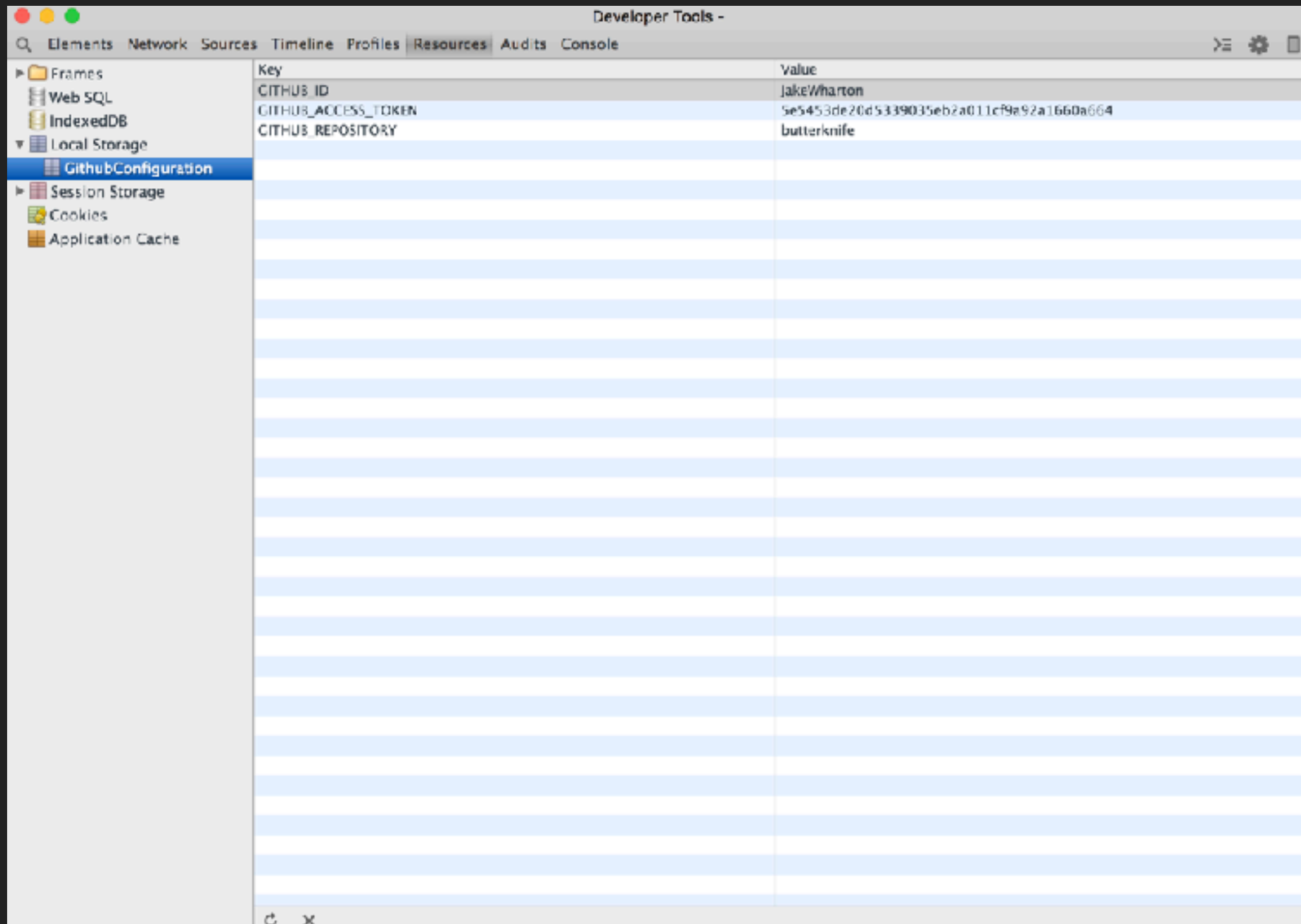
```
compile 'com.facebook.stetho:stetho-urlconnection:1.5.0'
```

Facebook에서 만든 안드로이드를 위한 디버그 브릿지 라이브러리

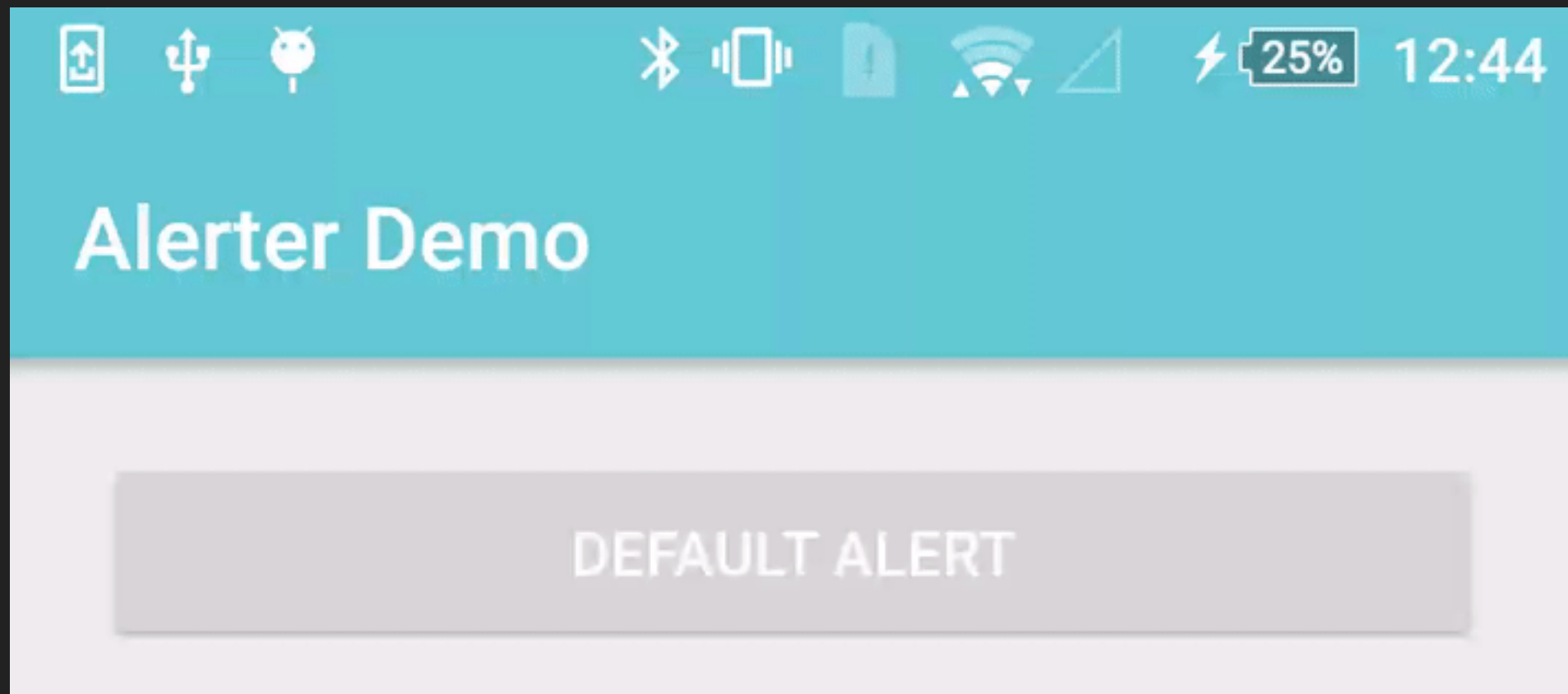
```
public class SampleApplication extends Application {  
  
    @Override  
    public void onCreate() {  
        super.onCreate();  
        Stetho.initializeWithDefaults(this);  
        new OkHttpClient.Builder()  
            .addNetworkInterceptor(new StethoInterceptor())  
            .build();  
    }  
}
```

`chrome://inspect/#devices`

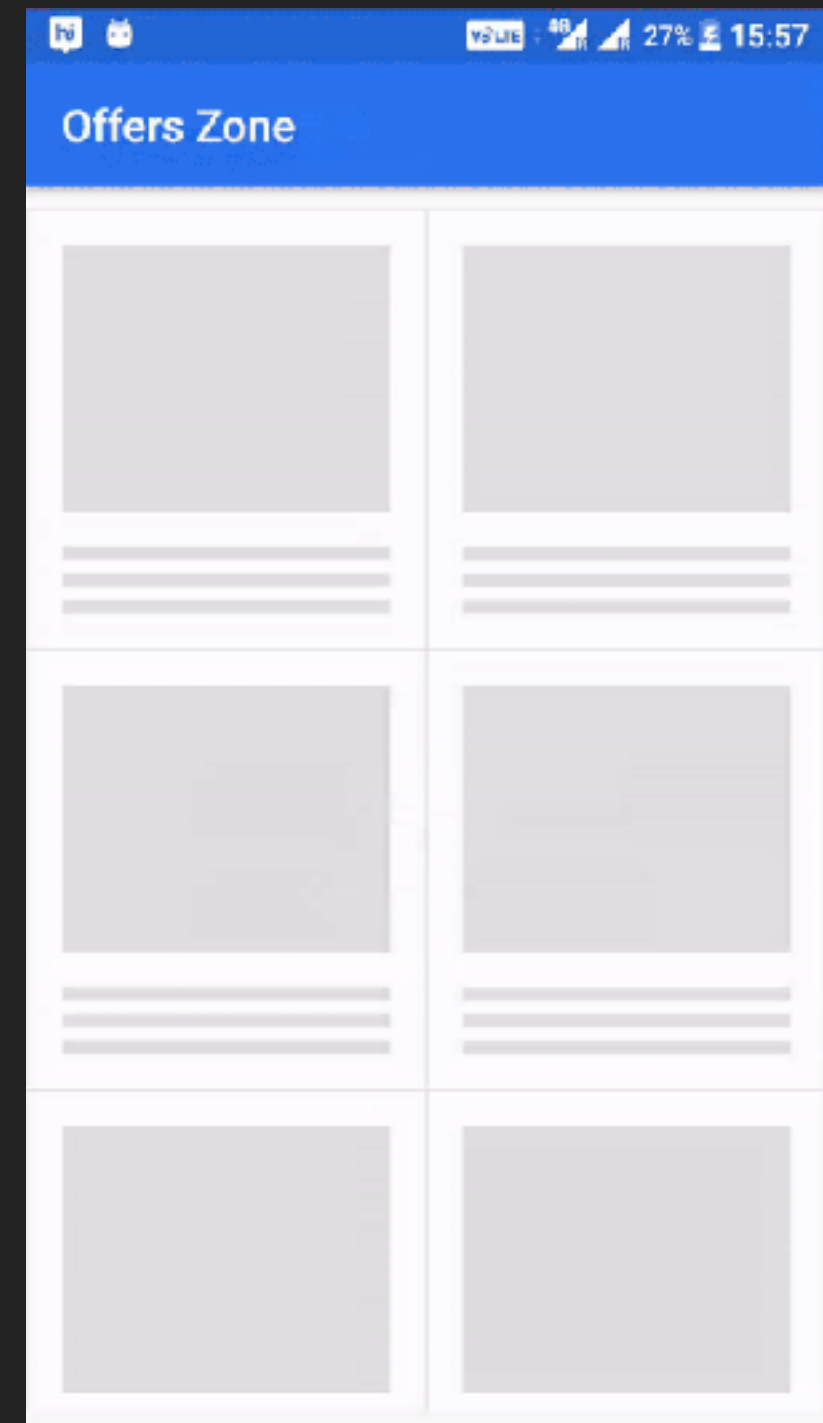
Facebook에서 만든 안드로이드를 위한 디버그 브릿지 라이브러리



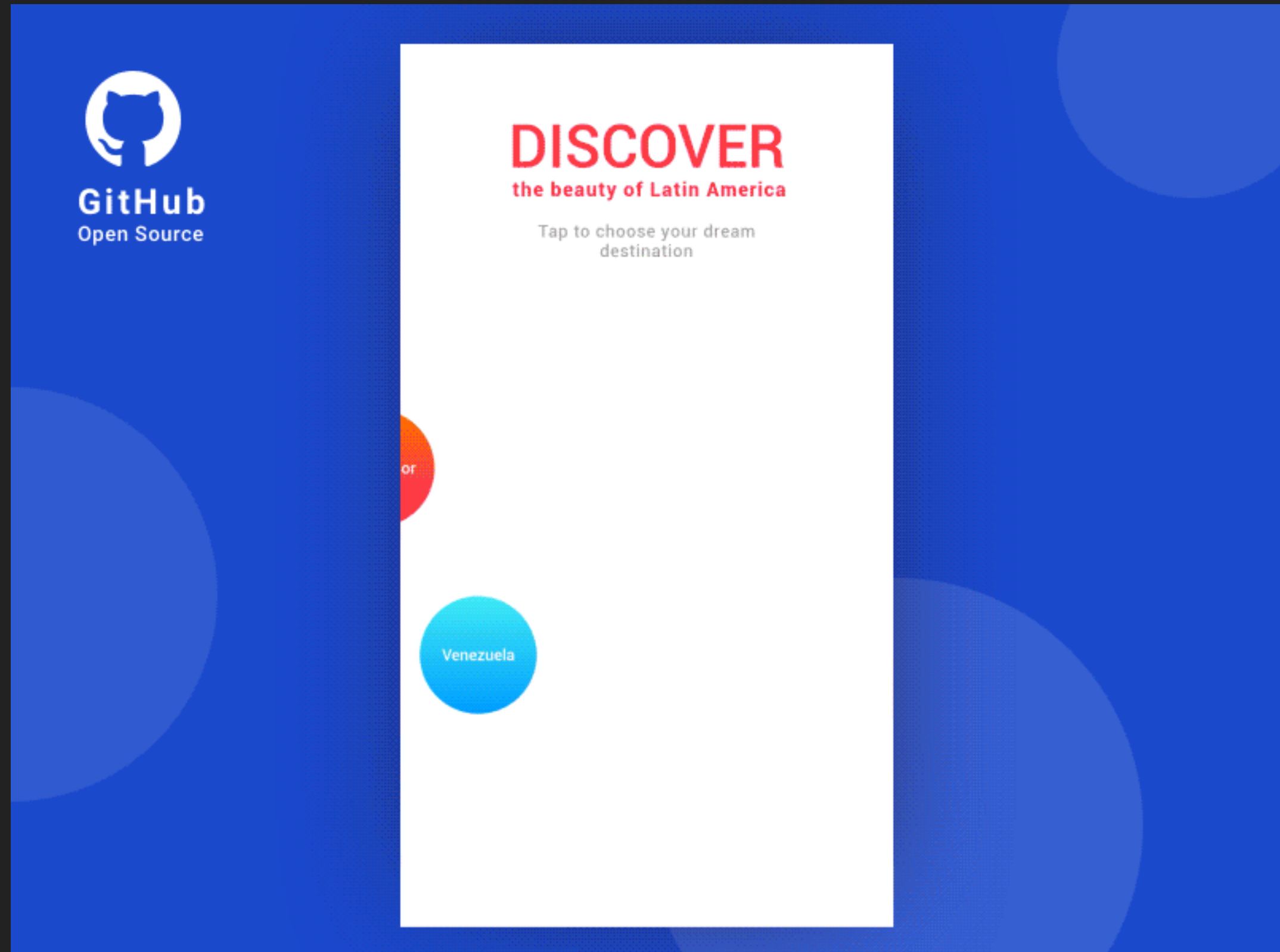
그 외 유용한 라이브러리



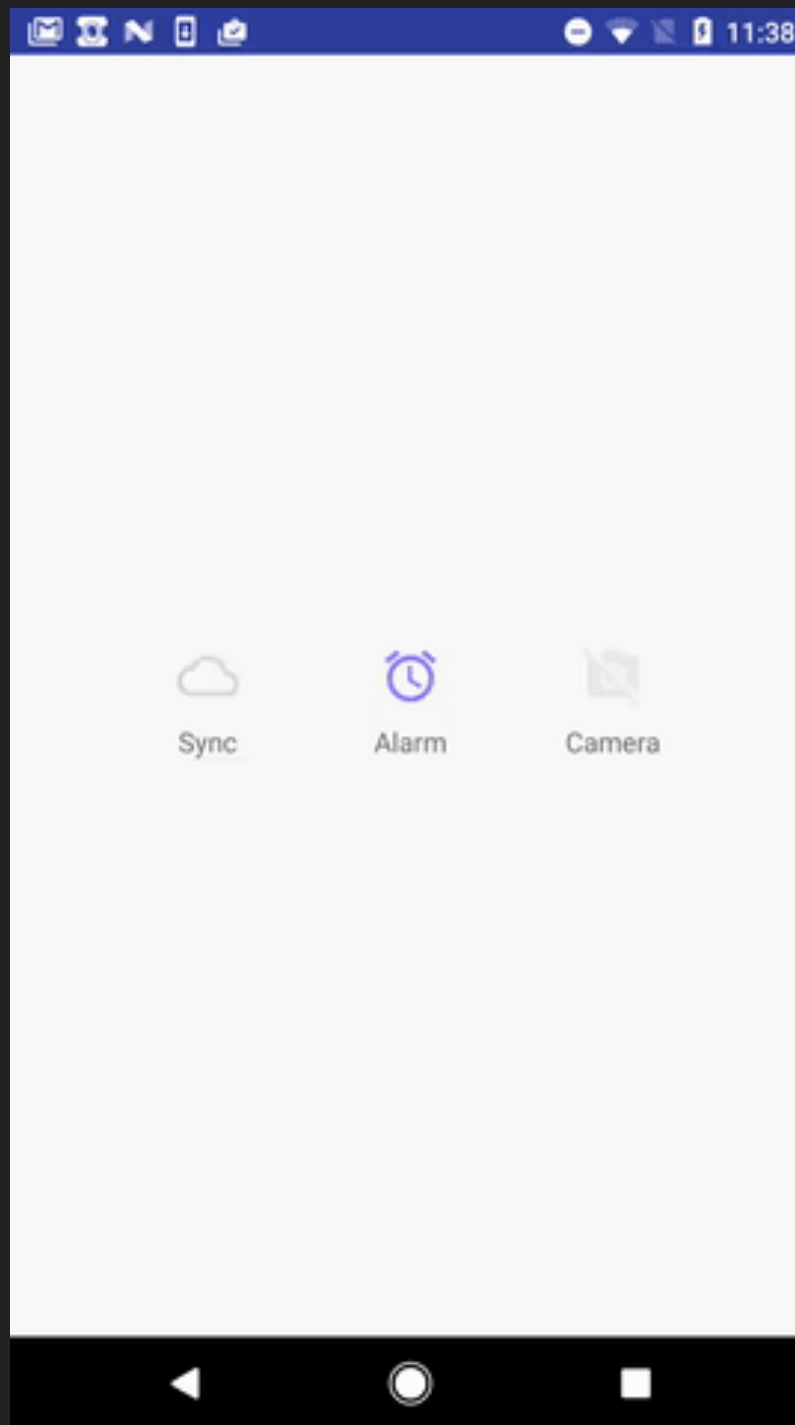
SHIMMERRECYCLERVIEW



BUBBLEPICKER



SWITCHICON



간단한 실습

<https://github.com/csi111/LibrarySample>