

업그레이드 모바일 개발환경

스칼로이드

이성호

pocorall@gmail.com

24 December 2013

온스퀘어

개발동기



- 실시간 상담 클라이언트



- 메트로놈과 튜너

안드로이드 앱 개발 중 겪은 일에서 출발

장황한 안드로이드 코드

```
public class MyService extends Service {
    private final IBinder mBinder = new LocalBinder();

    public class LocalBinder extends Binder {
        MyService getService() {
            return MyService.this;
        }
    }

    @Override
    public IBinder onBind(Intent intent) {
        return mBinder;
    }

    private final Random mGenerator = new Random();

    public int getRandomNumber() {
        return mGenerator.nextInt(100);
    }
}
```

```
public class BindingActivity extends Activity {
    MyService mService;
    boolean mBound = false;

    @Override
    protected void onStart() {
        super.onStart();
        Intent intent = new Intent(this, MyService.class);
        bindService(intent, mConnection, Context.BIND_AUTO_CREATE);
    }

    @Override
    protected void onStop() {
        super.onStop();
        if (mBound) {
            unbindService(mConnection);
            mBound = false;
        }
    }

    public void onClick(View v) {
        if (mBound) {
            int num = mService.getRandomNumber();
            Toast.makeText(this, "number: " + num, Toast.LENGTH_SHORT).show();
        }
    }

    private ServiceConnection mConnection = new ServiceConnection() {
        @Override
        public void onServiceConnected(ComponentName className,
            IBinder service) {
            LocalBinder binder = (LocalBinder) service;
            mService = binder.getService();
        }

        @Override
        public void onServiceDisconnected(ComponentName arg0) {
            mBound = false;
        }
    };
}
```

number: 92

장황한 안드로이드 코드

Java

```
Button button = new Button(context)
button.setText("Greet")
button.setOnClickListener(new OnClickListener() {
    public void onClick(view v) {
        Toast.makeText(context, "Hello!", Toast.LENGTH_SHORT).show()
    }
})
layout.addView(button)
```

Hello!

장황한 안드로이드 코드

XML layout description

```
• <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
•     android:orientation="vertical" android:layout_width="match_parent"
•     android:layout_height="wrap_content" android:padding="20dip">
•     <TextView android:layout_width="match_parent" android:text="ID"
•         android:id="@+id/userid" android:layout_height="wrap_content" />
•     <EditText android:layout_width="match_parent"
•         android:layout_height="wrap_content" android:id="@+id/userId"/>
•     <Button android:layout_width="match_parent"
•         android:layout_height="wrap_content" android:id="@+id/signin"
•         android:text="Sign in"/>
• </LinearLayout>
```

Java

```
• EditText userId = (EditText)findViewById(R.id.userid);
  Button signin = (Button)findViewById(R.id.signin);
  signin.setOnClickListener(new View.OnClickListener() {
      public void onClick(View v) {
          signin(userId.getText())
      }
  });
```

보다 적은 수의 논리로 설명이 가능한 경우,
많은 수의 논리를 세우지 말라



William of Ockham
1285 - 1349

개요

- 안드로이드 개발 효율을 향상시키는 스칼로이드
 - 기존 대비 두 배 안팎의 코드 단축
 - 가독성, 유지보수성 향상



<https://github.com/pocorall/scaloid>

JQuery와 스칼로이드



실행환경	안드로이드	웹브라우저
언어	스칼라	자바스크립트
개발형태	라이브러리	
장점	코드가 단축됨	

8 lines -> 1 line

Plain-old Android API

```
• val button=new Button(context)
  button.setText("Greet")
  button.setOnClickListener(new OnClickListener(){
    def onClick(v: View){
      Toast.makeText(context,"Hello!",Toast.LENGTH_SHORT).show()
    }
  })
  layout.addView(button)
```



Scaloid application

```
SButton("Greet", toast("hello!"))
```

XML layout description

```
• <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  •   android:orientation="vertical" android:layout_width="match_parent"
  •   android:layout_height="wrap_content" android:padding="20dip">
  •   <TextView android:layout_width="match_parent" android:text="ID"
    android:id="@+id/userid" android:layout_height="wrap_content" />
  •   <EditText android:layout_width="match_parent"
    android:layout_height="wrap_content" android:id="@+id/userId"/>
  •   <Button android:layout_width="match_parent"
    android:layout_height="wrap_content" android:id="@+id/signin"
    android:text="Sign in"/>
  • </LinearLayout>
```

Java

```
• EditText userId = (EditText)findViewById(R.id.userid);
  Button signin = (Button)findViewById(R.id.signin);
  signin.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
      signin(userId.getText())
    }
  });
```

Scaloid

```
• new SVerticalLayout {
  STextView("ID")
  val userId=SEditText()
  SButton("Sign in", signin(userId.text))
}.padding(20dip)
```

Two verbose things

VS. One simple thing

Responsive layout

- 화면 크기와 비율에 따른 다양한 콘텐츠 표시를 **한 개의 코드**로 실현
 - 스마트폰 앱 개발에 있어 **최초로** 시도됨

```
import org.scaloid.util.Config._

if(long) SButton("This button is shown only for a long screen "
  + "dimension (" + width + ", " + height + ")")
if(landscape) this += new SLinearLayout {
  SButton("Buttons for")
  SButton("landscape layout")
  if(dpi >= HDPI) SButton("You have a high resolution display!")
}
```

서비스

- <http://developer.android.com/guide> says:

```
public class MyService extends Service {
    private final IBinder mBinder = new LocalBinder();

    public class LocalBinder extends Binder {
        MyService getService() {
            return MyService.this;
        }
    }

    @Override
    public IBinder onBind(Intent intent) {
        return mBinder;
    }

    private final Random mGenerator = new Random();

    public int getRandomNumber() {
        return mGenerator.nextInt(100);
    }
}
```

```
public class BindingActivity extends Activity {
    MyService mService;
    boolean mBound = false;

    @Override
    protected void onStart() {
        super.onStart();
        Intent intent = new Intent(this, MyService.class);
        bindService(intent, mConnection, Context.BIND_AUTO_CREATE);
    }

    @Override
    protected void onStop() {
        super.onStop();
        if (mBound) {
            unbindService(mConnection);
            mBound = false;
        }
    }

    public void onClick(View v) {
        if (mBound) {
            int num = mService.getRandomNumber();
            Toast.makeText(this, "number: " + num, Toast.LENGTH_SHORT).show();
        }
    }

    private ServiceConnection mConnection = new ServiceConnection() {
        @Override
        public void onServiceConnected(ComponentName className,
            IBinder service) {
            LocalBinder binder = (LocalBinder) service;
            mService = binder.getService();
            mBound = true;
        }

        @Override
        public void onServiceDisconnected(ComponentName arg0) {
            mBound = false;
        }
    };
}
```

LocalService를 이용한 단축 표현

```
class MyService extends LocalService {  
    private val generator = new Random()  
  
    def getRandomNumber() = generator.nextInt(100)  
}
```

```
class MyActivity extends SActivity {  
    val random = new LocalServiceConnection[MyService]  
  
    def onClick(v:View) {  
        random( s => toast("number: " + s.getRandomNumber()))  
    }  
}
```

스칼로이드가 도와주는 일

- 라이프싸이클 관리
- 리소스 해제
- 비동기 작업 처리
- 스타일링
- 로컬 서비스
- 프리퍼런스

자세한 내용은
Github 페이지의
도움말 참조

<https://github.com/pocorall/scaloid>

스칼로이드를 써보세요

- 배우기 쉽습니다
- 인기 앱에서 사용중입니다
- 용량이 적고 빠릅니다
- 부분적으로 도입할 수 있습니다

배우기 쉽습니다

Plain-old Android API

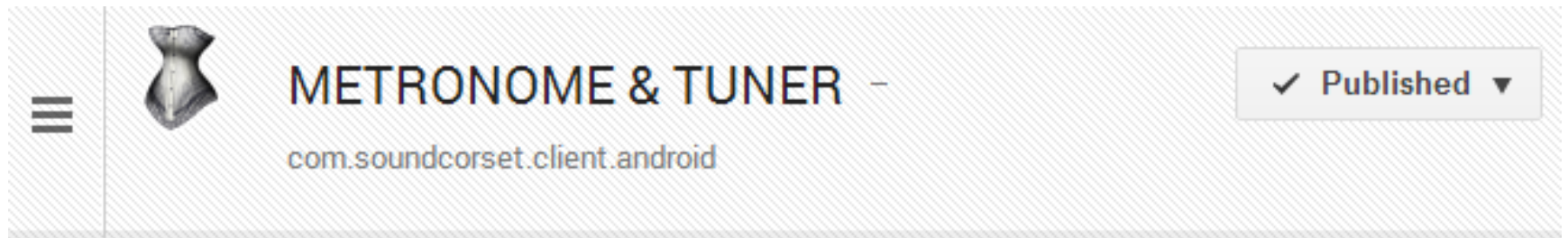
```
• val button=new Button(context)
  button.setText("Greet")
  button.setOnClickListener(new OnClickListener(){
    def onClick(v: View){
      Toast.makeText(context,"Hello!",Toast.LENGTH_SHORT).show()
    }
  })
  layout.addView(button)
```



Scaloid application

```
SButton("Greet", toast("hello!"))
```


스칼로이드 적용사례



STATISTICS

Total installs by user

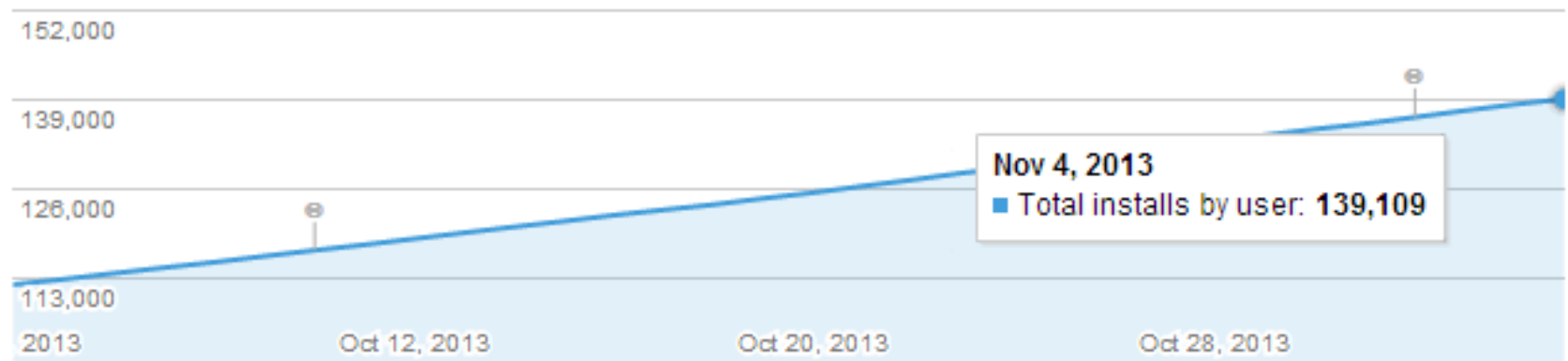


for Oct 4, 2013 - Nov 4, 2013

[Export as CSV](#)

Show: [last month](#) [3m](#) [6m](#) [1y](#) [all](#)

The total number of unique users who have ever installed this app on one or more of their devices. [Learn more](#)




용량과 속도

- Soundcorset metronome & tuner
 - **1MB 미만**
- 스칼로이드는 JVM용 바이트코드로 컴파일
 - 일반 자바 코드와 비슷하게 **매우 빠름**

점진적으로 도입 가능

- 기존 앱의 일부분만 스칼로이드로 작성 가능
- 스칼로이드가 제공하는 기능의 일부분만 활용 가능
- 어떻게 도입해야 할지 잘 모르겠는 부분은
 - 종전대로 안드로이드 API를 사용하면 됨



새로운 걸 도입했다가 괜히 사서 골치씨을지 모른다는
염려가 없습니다

스칼로이드 포럼

<https://groups.google.com/forum/#!forum/scaloid-user>

Scaloid-user 공개적으로 공유함

총 13개 주제 중 13개 ★ 

[관리](#) · [회원](#) · [정보](#) 

Scaloid is an open-source library aimed to simplify your Android code. It makes your code easy to write and understand by leveraging **Scala language**.

Feel free to discuss about new features, share knowledge, and ask questions.

[환영 메시지](#) [수정](#) [환영 메시지](#) [삭제](#)

<input type="checkbox"/>		Call UI methods from Object 작성자: Marcio Giaxa Marinho - 소식 4개 - 11회 조회	오전 12:39
<input type="checkbox"/>		How to create menus? 작성자: Mike Limansky - 소식 2개 - 3회 조회	10월 27일
<input type="checkbox"/>		How to make a BaseAdapter the Scaloid way? 작성자: Mats Sjöberg - 소식 5개 - 12회 조회	10월 27일
<input type="checkbox"/>		Create Bare-bones IntelliJ IDEA/Scala Project 작성자: Nephilim Faustus - 소식 7개 - 15회 조회	10월 21일
<input type="checkbox"/>		Error on load lib .so 작성자: Marcio Giaxa Marinho - 소식 2개 - 3회 조회	10월 20일
<input type="checkbox"/>		Fatal exception on String.toDouble conversion 작성자: Mats Sjöberg - 소식 3개 - 6회 조회	10월 5일
<input type="checkbox"/>		Using SFragmentManager (and other things from the support package) 작성자: Mark Eibes - 소식 2개 - 21회 조회	10월 3일
<input type="checkbox"/>		n00b struggling with scala/maven 작성자: Gregory Huff - 소식 6개 - 82회 조회	7월 1일
<input type="checkbox"/>		SWebView not recognised? 작성자: John May - 소식 3개 - 24회 조회	6월 23일



Results for scaloid



Top / All / People you follow



Pascal Hartig @passy

29 Sep

Scaloid, a Scala wrapper for the Android API, looks fantastic. So much less boilerplate code to write. [github.com/pocorall/scalo...](https://github.com/pocorall/scaloid)

Expand



Bartek Ciszkowski @bartekci

18 Jun

So **Scaloid**, which just had a huge 2.0 release yesterday looks awesome. [github.com/pocorall/scalo...](https://github.com/pocorall/scaloid)

[View summary](#)



Rex Villafior @iQHQ

11 Jun

SW Development :: Music Player with **Scaloid** - oDesk: I found a cool project: [github.com/pocorall/scaloi...](https://github.com/pocorall/scaloid) bit.ly/18u8Z36

Expand



Santiago Basulto @santiagobasulto

3 Jan

Scala for android development. Seems like a cool promise. I'd love to get rid of all those XMLs. [github.com/pocorall/scalo...](https://github.com/pocorall/scaloid) @sserrano44
#scala

[View summary](#)

585개의 github 북마크

GitHub repository page for pocorall / scaloid. The page shows the repository name, a search bar, and navigation links (Explore, Gist, Blog, Help). The repository is owned by pocorall. The star count is 585, which is highlighted by a red box.

Stargazers



Dave

Base79

+ Follow



Nelson Chen

SmartReceipt Inc.

+ Follow



Lewis O'Driscoll

Oxfordshire, United Kingdom

+ Follow



Carl Mercier

Codified Inc

+ Follow



Michael Wall

Joined on Mar 05, 2008

+ Follow



Antonio Pardo

aLabs

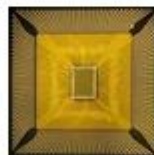
+ Follow



Tim Cowlshaw

Itinerant software and data w...

+ Follow



maurycy

52.2318381, 21.0059946

+ Follow



Chase Colman

Joined on Apr 06, 2008

+ Follow



Andrey Dolganov



+ Follow



Ståle Pettersen

Oslo, Norway

+ Follow



Lalit Shandilya

New Delhi, India

+ Follow

MOBILE

[Tweet](#) 43 [Like](#) 46 [Share](#) [+1](#) [Dribbble](#) [Email](#) [Print](#) [Permalink](#)

Developing Android Apps with Scala and Scaloid: Part 1

By [Gaston Hillar](#), September 24, 2013[2 Comments](#)**Build a UI layout by writing type-safe Scala code and wire your logic into the layout**

[Scaloid](#) is an open-source library that enables Scala developers to create Android apps without having to migrate to Java. Scaloid takes full advantage of many of Scala's features, such as the efficient way of creating Domain Specific Languages (DSLs), implicit conversions, pattern matching, and type safety. Scaloid proposes a novel method of developing Android apps, which is worth a second look. In this lead article of a two-part series on Scaloid, I explain some of its most interesting features.

Simplifying Android Code with Scala Features

Scaloid focuses on simplifying and reducing the required Android code as much as possible while taking advantage of type safety. You can make incremental use of many Scaloid features in your Android projects because you can mix Scala and Scaloid with Java and the Android API.

Think of Scaloid as a library that provides shortcuts to tasks that usually require a large amount of code. Scaloid replaces the XML layout description required by the Android SDK as well as the Java code that specifies the logic with a single piece of Scaloid code that uses a Scala DSL. In this way, you can build a UI layout by writing type-safe Scala code and wire your logic into the layout. However, you will use Scaloid to access widgets defined in XML layouts, so you can continue working with XML layouts if you don't want to make a big paradigm shift in your Android development process.

There is an automatic layout converter that translates an Android XML layout description into

MOBILE

[Tweet](#) 23 [Like](#) 12 [Share](#) [+1](#) [Dribbble](#) [Email](#) [Print](#) [Permalink](#)

Developing Android Apps with Scala and Scaloid: Part 2

By [Gaston Hillar](#), October 08, 2013[Post a Comment](#)**Starting with templates, Android features can be added quickly with a single line of DSL code.**

In the [first installment](#) of this two-part series on developing Android Apps with Scala and Scaloid, I explained how Scaloid simplifies and reduces the required Android code as much as possible while leveraging type safety. In this article, I explain how to utilize asynchronous task processing, the execution of methods from system services, and specific Scaloid classes and traits.

Creating a Scaloid Project Based on a Template

The easiest way to start a new Scaloid project is to use one of the available template projects as a starting point. Following the instructions on how to do this can be difficult, so I provide some details on starting a successful Scaloid project with Maven. Scaloid also works with Scala's simple build tool ([sbt](#)) although there were issues in previous versions.

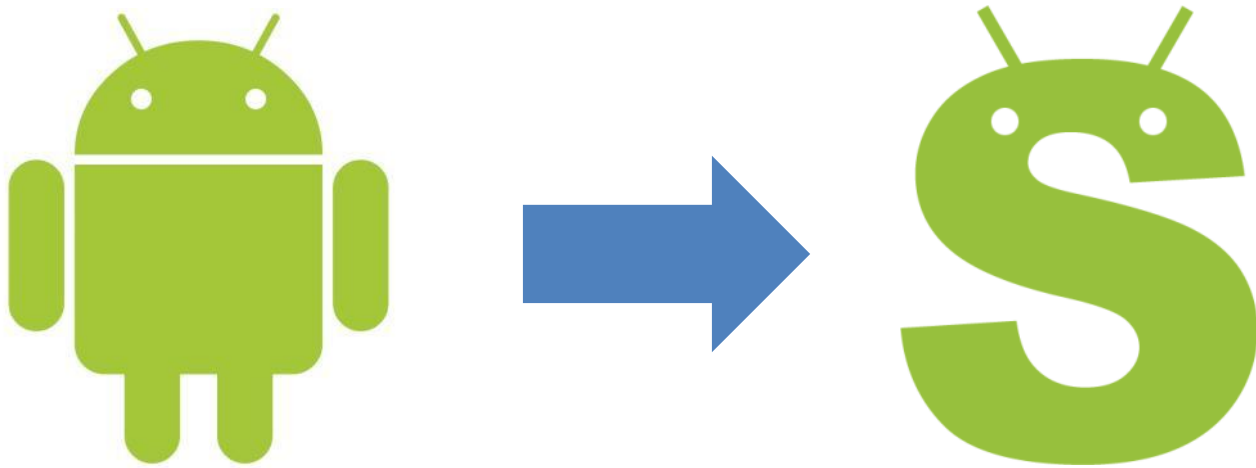
First, you need to fork the [Hello world of Scaloid for maven](#) project from its GitHub repository. Then, open the `pom.xml` file, located in the root folder, and change the value of the target Android SDK platform from 8 to the API Level value of the Android Virtual Device you want to target. You can find the value in project -> build -> plugins -> plugin -> configuration -> sdk -> platform. The following lines show the default value and the previous lines in the XML file:

```
1 ...
2 <build>
```

미국의 유명 개발 웹진인 Dr. Dobb's에서
2회에 걸쳐 스칼로이드 소개 기사를 게재

- <http://www.drdobbs.com/mobile/developing-android-apps-with-scala-and-s/240161584>
- <http://www.drdobbs.com/mobile/developing-android-apps-with-scala-and-s/240162204>

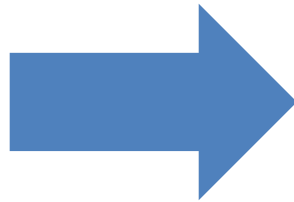
발전방향



안드로이드 개발의 새로운 표준으로 발전

사실상의 표준 *de facto* standard

J2EE



Sun에서 개발

위원회 방식으로 운영

방만한 스펙과 사용법

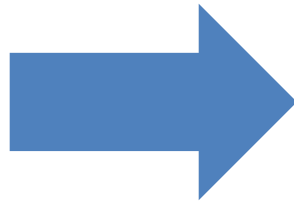
Rod Johnson이 창시

간결함을 목표로 디자인

J2EE의 번거로운 점을 해소

사실상의 표준 *de facto* standard

Javascript
DOM access

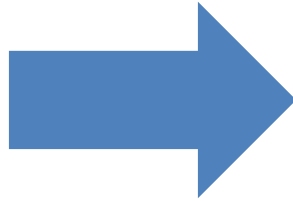


작성하기 번거로움

에러 발생 위험이 높음

John Resig이 창시

간결함을 목표로 디자인



요약

- 불필요한 코드를 줄여주는 스칼로이드
- 실무에 도입 가능한 완성도
 - 배우기 쉬움
 - 인기 앱에서 사용중
 - 용량이 작고 실행이 빠름
 - 부분적인 도입 가능
- 안드로이드 개발의 차세대 표준이 되는 것이 목표
 - 국내에서 개발된 글로벌 오픈소스

보조자료

스칼라 언어

- 함수형 + 객체지향 하이브리드 언어
- 강력한 타입 시스템
 - IDE에서 자동완성 지원 용이
- Java bytecode로 컴파일됨
 - 자바와 호환성 우수
- 안드로이드 개발 가능



안드로이드 + 스칼라

- 2개의 프로덕션 앱을 스칼라로 개발중
 - 톡플러스
 - 웹사이트 방문객과의 실시간 상담 서비스
 - 11,000여개의 웹사이트에 부착
 - 사운드코르셋
 - 악기 연습 가이드 모바일 앱
 - 13만 다운로드 돌파



예제 1:

프로그램에서 버튼을 추가하기

Java

```
Button button = new Button(context)
button.setText("Greet")
button.setOnClickListener(new OnClickListener() {
    public void onClick(view v) {
        Toast.makeText(context, "Hello!", Toast.LENGTH_SHORT).show()
    }
})
layout.addView(button)
```


스칼라로 직역한 코드

Java

```
• Button button = new Button(context)
  button.setText("Greet")
  button.setOnClickListener(new OnClickListener() {
      public void onClick(view v) {
          Toast.makeText(context, "Hello!", Toast.LENGTH_SHORT).show()
      }
  })
  layout.addView(button)
```



Scala

```
• val button=new Button(context)
  button.setText("Greet")
  button.setOnClickListener(new OnClickListener(){
      def onClick(v: View){
          Toast.makeText(context,"Hello!",Toast.LENGTH_SHORT).show()
      }
  })
  layout.addView(button)
```

스칼라로 직역한 코드

Java

```
• Button button = new Button(context)
  button.setText("Greet")
  button.setOnClickListener(new OnClickListener() {
    public void onClick(view v) {
      Toast.makeText(context, "Hello!", Toast.LENGTH_SHORT).show()
    }
  })
  layout.addView(button)
```

???

장점이 거의 없어 보이는데?

Scala

```
• val button=new Button(context)
  button.setText("Greet")
  button.setOnClickListener(new OnClickListener(){
    def onClick(v: View){
      Toast.makeText(context,"Hello!",Toast.LENGTH_SHORT).show()
    }
  })
  layout.addView(button)
```

스칼라로 직역한 코드

Java

```
• Button button = new Button(context)
  button.setText("Greet")
  button.setOnClickListener(new OnClickListener() {
    public void onClick(view v) {
      Toast.makeText(context, "Hello!", Toast.LENGTH_SHORT).show()
    }
  })
  layout.addView(button)
```

언어의 문제가 아니라
라이브러리의 문제임

Scala

```
• val button=new Button(context)
  button.setText("Greet")
  button.setOnClickListener(new OnClickListener(){
    def onClick(v: View){
      Toast.makeText(context,"Hello!",Toast.LENGTH_SHORT).show()
    }
  })
  layout.addView(button)
```

Context

Application

- `val button=new Button(context)`
 `button.setText("Greet")`
 `button.setOnClickListener(new OnClickListener(){`
 `def onClick(v: View){`
 `Toast.makeText(context,"Hello!",Toast.LENGTH_SHORT).show()`
 `}`
 `})`
 `layout.addView(button)`

Context as an implicit value

Scaloid

- Class `SButton(implicit ctx:Context)` extends `Button(ctx)`

Application

- `implicit val ctx = ...`

- `val button=new SButton`

```
    button.setText("Greet")
```

```
    button.setOnClickListener(new OnClickListener(){
```

```
        def onClick(v: View){
```

```
            Toast.makeText(context,"Hello!",Toast.LENGTH_SHORT).show()
```

```
        }
```

```
    })
```

```
    layout.addView(button)
```

Listeners

Scaloid

- Class SButton(implicit ctx:Context) extends Button(ctx)

Application

- implicit val ctx = ...

- val button=new SButton
button.setText("Greet")

```
button.setOnClickListener(new OnClickListener(){  
    def onClick(v: View){  
        Toast.makeText(context,"Hello!",Toast.LENGTH_SHORT).show()  
    }  
})
```

```
layout.addView(button)
```

Shorter listener

Scaloid

- ```
class SButton(implicit ctx:Context) extends Button(ctx) {
 def onClick(arg: => Unit) = setOnClickListener(
 new OnClickListener(){
 def onClick(v: View){
 arg
 }
 })
}
```

## Application

- ```
val button=new SButton  
button.setText("Greet")  
button.onClick {  
  Toast.makeText(context,"Hello!",Toast.LENGTH_SHORT).show()  
}  
layout.addView(button)
```

Toast

Scaloid

- `def toast(text:CharSequence)(implicit ctx:Context) =
 Toast.makeText(ctx, text, Toast.LENGTH_SHORT).show()`

Application

- `val button=new SButton
 button.setText("Greet")
 button.onClick(toast("hello!"))
 layout.addView(button)`

Function apply()

Scaloid

- object SButton {
 def apply(txt:CharSequence, arg: =>Unit)
 (implicit ctx:Context): SButton = {
 val button = new Sbutton
 button.setText(txt)
 button.onClick(arg)
 button
 }
}

Application

- val button= SButton("Greet", toast("hello!"))
 layout.addView(button)

Implicit layout context

Scaloid

- object SButton {
 def apply(txt:CharSequence, arg: =>Unit)
 (implicit ctx:Context, layout:ViewGroup): SButton = {
 val button = new Sbutton
 button.setText(txt)
 button.onClick(arg)
 layout.addView(button)
 button
 }
}

Application

SButton("Greet", toast("hello!"))

8 lines -> 1 line

Plain-old Android API

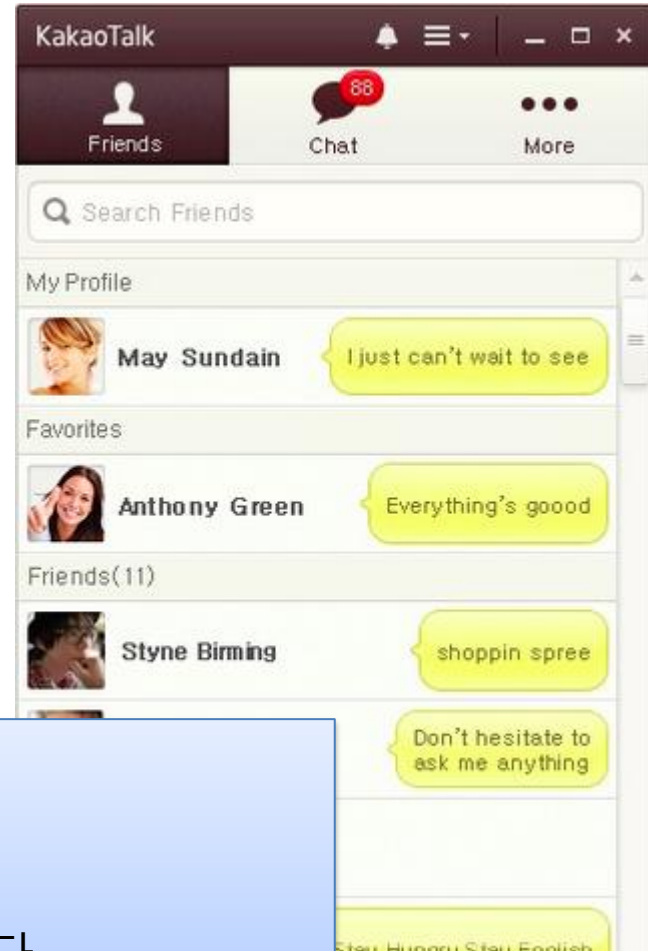
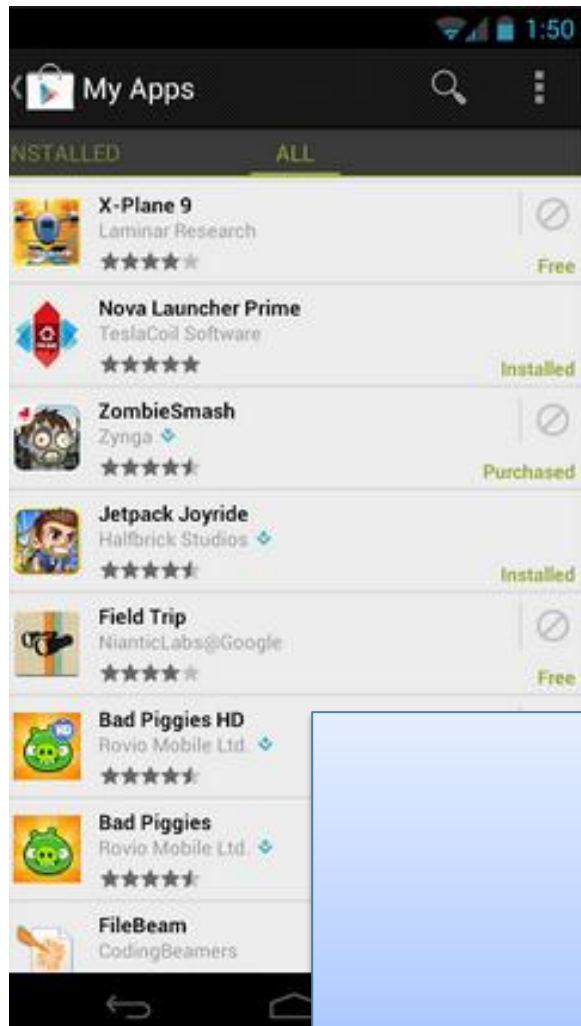
```
• val button=new Button(context)
  button.setText("Greet")
  button.setOnClickListener(new OnClickListener(){
    def onClick(v: View){
      Toast.makeText(context,"Hello!",Toast.LENGTH_SHORT).show()
    }
  })
  layout.addView(button)
```



Scaloid application

```
SButton("Greet", toast("hello!"))
```

여러 개의 버튼 목록 표시하기



1. 버튼 목록이
2. 동적으로 생성되어야 한다

여러 개의 버튼 목록 표시하기

Scaloid

```
names.foreach(name => SButton(name, toast(name)))
```

XML

```
<Button android:layout_width="wrap_content" android:layout_height="wrap_content"
  android:text="2"/>
<Button android:layout_width="wrap_content" android:layout_height="wrap_content"
  android:text="3"/>
<Button android:layout_width="wrap_content" android:layout_height="wrap_content"
  android:text="4"/>
<Button android:layout_width="wrap_content" android:layout_height="wrap_content"
  android:text="5"/>
<Button android:layout_width="wrap_content" android:layout_height="wrap_content"
  android:text="6"/>
<Button android:layout_width="wrap_content" android:layout_height="wrap_content"
  android:text="7"/>
<Button android:layout_width="wrap_content" android:layout_height="wrap_content"
  android:text="8"/>
<Button android:layout_width="wrap_content" android:layout_height="wrap_content"
  android:text="9"/>
<Button android:layout_width="wrap_content" android:layout_height="wrap_content"
  android:text="10"/>
```

...

Java

```
• for(String name : names) {
    Button button = new Button(context)
    button.setText(name)
    button.setOnClickListener(new OnClickListener() {
        public void onClick(view v) {
            Toast.makeText(context, name,
                Toast.LENGTH_SHORT).show()
        }
    })
    layout.addView(button)
}
```

스칼로이드 소개



<http://scaloid.org>

- 장점
 - 우아한 안드로이드 코딩을 도움
 - 간결함
 - 엄격한 타입 검사 수행
 - 사용하기 쉬움
 - 기존 코드와 완벽하게 호환
 - 활발하게 개발중

JQuery의 비유



실행환경	안드로이드	웹브라우저
언어	스칼라	자바스크립트
개발형태	라이브러리	
장점	코드가 단축됨	

스칼로이드가 도와주는 일

- 라이프싸이클 관리
- 리소스 해제
- 비동기 작업 처리
- 스타일링
- 로컬 서비스
- 프리퍼런스

라이프사이클 관리

- BroadcastReceiver를 등록하고 해제하기

Plain-old Android API

- var connectivityListener: BroadcastReceiver = null

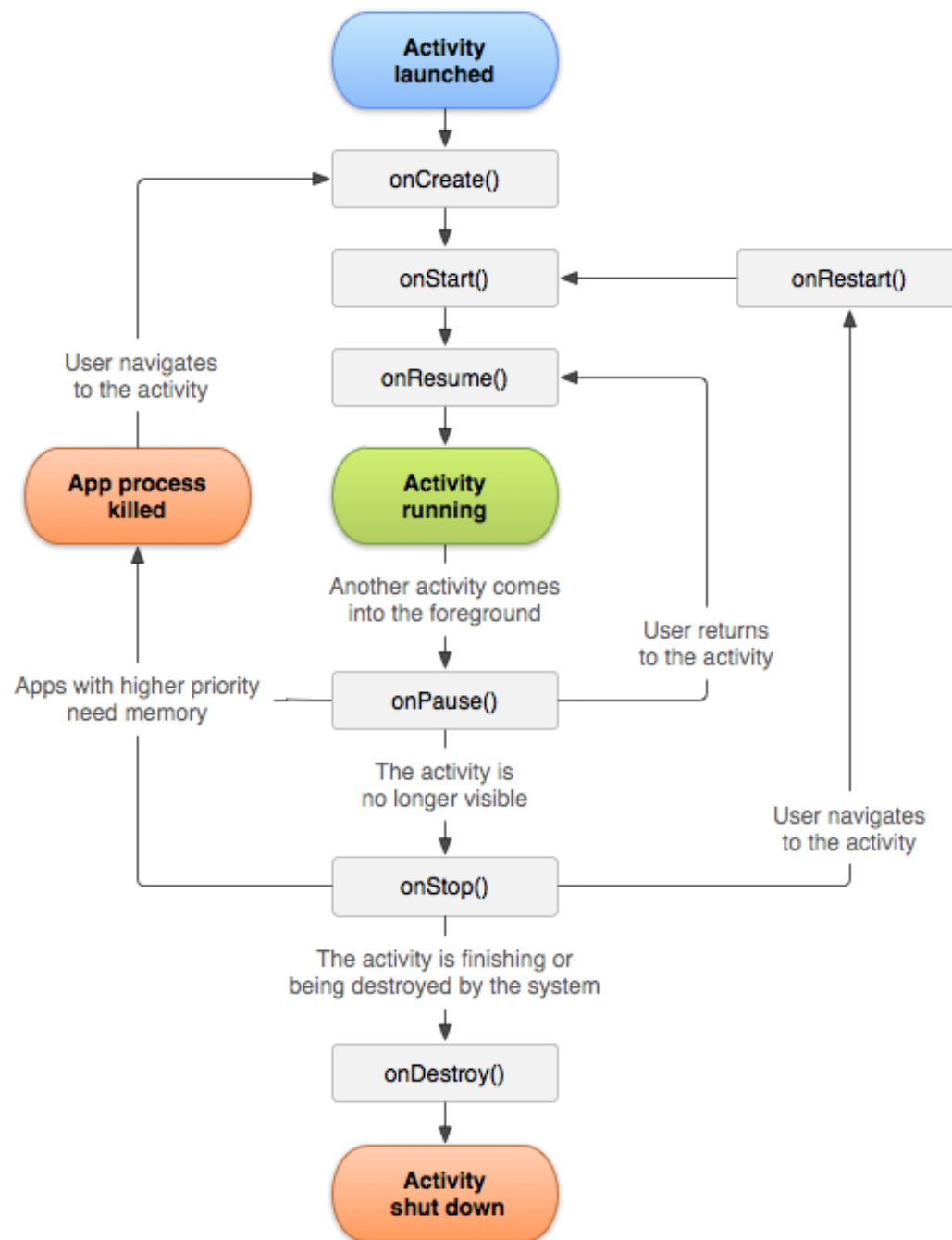
```
def onResume() {  
    super.onResume()  
    // ...  
    connectivityListener = new BroadcastReceiver {  
        def onReceive(context: Context, intent: Intent) {  
            doSomething()  
        }  
    }  
    registerReceiver(connectivityListener, new IntentFilter(ConnectivityManager.CONNECTIVITY_ACTION))  
}  
  
def onPause() {  
    unregisterReceiver(connectivityListener)  
    // ...  
    super.onPause()  
}
```

라이프사이클 관리

- BroadcastReceiver를 등록하고 해제하기

Scaloid

- `broadcastReceiver(ConnectivityManager.CONNECTIVITY_ACTION)`
 { (context, intent) =>
 doSomething()
 }



라이프싸이클 관리

- BroadcastReceiver를 등록하고 해제하기

Scaloid

- `broadcastReceiver(ConnectivityManager.CONNECTIVITY_ACTION)`
 { (context, intent) =>
 doSomething()
 }(this, onStartStop)

더 나은 리소스 해제

- 문제:
여러 종류의 리소스를 각각 관리해야 함

```
class MyService extends Service {  
    protected List<WorkerThread> threads = new ArrayList<WorkerThread>  
    protected List<File> files = new ArrayList<File>()  
  
    public void openFile(...) {  
        //...  
        receivers.add(openedFile);  
    }  
    public void startNewJob(...) {  
        //...  
        threads.add(workerThread);  
    }  
  
    public void onDestroy() {  
        for(file : files) {  
            file.close();  
        }  
        for(thread: threads) {  
            thread.sendMessage();  
        }  
        super.onDestroy();  
    }  
}
```

해법

- onDestroy()를 도입

```
class MyService extends SService {  
  def openFile(...) {  
    //...  
    onDestroy(openedFile.close())  
  }  
  def startNewJob(...) {  
    //... brace can also be used  
    onDestroy {  
      workerThread.sendMessage()  
    }  
  }  
}
```

비동기 작업 처리

Plain-old Android API

- `activity.runOnUiThread {`
 `new Runnable() {`
 `def run() {`
 `debug("Running only in Activity class")`
 `}`
 `}`
}

비동기 작업 처리

Scaloid

```
runOnUiThread(debug("Running in any context"))
```


비동기 작업 처리

Using AsyncTask

- ```
new AsyncTask[String, Void, String] {
 def doInBackground(params: Array[String]) = {
 doAJobTakeSomeTime(params)
 }

 override def onPostExecute(result: String) {
 alert("Done!", result)
 }
}.execute("param")
```

# 비동기 작업 처리

## Using future and runOnUiThread

- ```
future {  
    val result = doAJobTakeSomeTime(params)  
    runOnUiThread(alert("Done!", result))  
}
```

비동기 작업 처리

Scaloid

- ```
future {
 alert("Done!", doAJobTakeSomeTime(params))
}
```

# 좋은 언어 vs. 코드 생성기

- WYSIWIG코드 생성기가 필요한 경우
  - 코드를 입력하는 일이 골치아프기 때문에 존재함
    - 언어가 잘못 디자인되었다는 뜻
    - 읽기도 힘들고 관리가 불가능함
- 좋은 DSL을 설계
  - 읽고 관리하기 쉬움
  - 왜 코드를 자동 생성하려고 하지요?

# Responsive layout

- 레이아웃 요소들이 런타임에 결정되는 경우
  - 어차피 WYSIWYG로 못 만듭니다

```
import org.scaloid.util.Config._

if(long) SButton("This button is shown only for a long screen "
 + "dimension (" + width + ", " + height + ")")
if(landscape) this += new SLinearLayout {
 SButton("Buttons for")
 SButton("landscape layout")
 if(dpi >= HDPI) SButton("You have a high resolution display!")
}
```

## XML layout description

```
• <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
 • android:orientation="vertical" android:layout_width="match_parent"
 • android:layout_height="wrap_content" android:padding="20dip">
 • <TextView android:layout_width="match_parent" android:text="ID"
 android:id="@+id/userid" android:layout_height="wrap_content" />
 • <EditText android:layout_width="match_parent"
 android:layout_height="wrap_content" android:id="@+id/userId"/>
 • <Button android:layout_width="match_parent"
 android:layout_height="wrap_content" android:id="@+id/signin"
 android:text="Sign in"/>
 • </LinearLayout>
```

## Java

```
• EditText userId = (EditText)findViewById(R.id.userid);
 Button signin = (Button)findViewById(R.id.signin);
 signin.setOnClickListener(new View.OnClickListener() {
 public void onClick(View v) {
 signin(userId.getText())
 }
 });
```

## Scaloid

```
• new SVerticalLayout {
 STextView("ID")
 val userId=SEditText()
 SButton("Sign in", signin(userId.text))
}.padding(20dip)
```

Two verbose things

VS. One simple thing

# 프로그래머를 위한 스타일링

## XML layout

- <TextView  
    android:id="@+id/helloText"  
    android:layout\_width="fill\_parent"  
    android:layout\_height="wrap\_content"  
    android:layout\_centerHorizontal="true"  
    android:layout\_alignParentBottom="true"  
    android:text="hello"/>



## Scaloid

```
STextView("hello").< <.centerHorizontal.alignParentBottom.> >
```

# 안드로이드에서의 캐스캐이드 스타일 시트

## Repeated code

- `SButton("first").textSize(20 dip).<<.margin(5 dip).>>`  
`SButton("prev").textSize(20 dip).<<.margin(5 dip).>>`  
`SButton("next").textSize(20 dip).<<.margin(5 dip).>>`  
`SButton("last").textSize(20 dip).<<.margin(5 dip).>>`

## Reduced with a function

- `def myStyle = (_: SButton).textSize(20 dip).<<.margin(5 dip).>>`  
`myStyle(SButton("first"))`  
`myStyle(SButton("prev"))`  
`myStyle(SButton("next"))`  
`myStyle(SButton("last"))`

## Further reduced using foreach

- `def myStyle = (_: SButton).textSize(20 dip).<<.margin(5 dip).>>`  
`List("first", "prev", "next", "last").foreach(title => myStyle(SButton(title)))`



# 안드로이드에서의 캐스캐이드 스타일 시트

## Using style

- style {  
    case b: SButton => b.textSize(20 dip).<<.margin(5 dip).>>  
}

SButton("first")

SButton("prev")

SButton("next")

SButton("last")

# 안드로이드에서의 캐스캐이드 스타일 시트

I am 10 dip tall

Me too

I am taller than you

Yellow input field

Red alert!

## Code

- `TextView("I am 10 dip tall")`  
`TextView("Me too")`  
`TextView("I am taller than you")`  
`EditText("Yellow input field")`  
`Button("Red alert!")`

# 안드로이드에서의 캐스케이드 스타일 시트

I am 10 dip tall  
Me too  
I am taller than you

Yellow input field

Red alert!



I am 10 dip tall  
Me too  
I am taller than you

Yellow input field

Red alert!

## Using style

```
• style {
 case b: SButton =>
 b.textColor(Color.RED).onClick(toast("Bang!"))
 case t: STextView => t.textSize(10 dip)
 case v => v.backgroundColor(Color.YELLOW)
}

STextView("I am 10 dip tall")
STextView("Me too")
// overriding textSize
STextView("I am taller than you").textSize(15 dip)
SEditText("Yellow input field")
SButton("Red alert!")
```

# Scaloid Layout Converter

This program converts an Android XML layout into a [Scaloid](http://layout.scaloid.org) layout.

**Paste Android XML layout here:**

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
 android:orientation="vertical" android:layout_width="match_parent"
 android:layout_height="wrap_content" android:padding="20dip">
 <TextView android:layout_width="match_parent"
 android:layout_height="wrap_content" android:text="Sign in"
 android:layout_marginBottom="25dip"
 android:textSize="24.5sp"/>
 <TextView android:layout_width="match_parent"
 android:layout_height="wrap_content" android:text="ID"/>
```

제출

<http://layout.scaloid.org>

Converted [Scaloid](http://layout.scaloid.org) layout is:

```
override def onCreate(savedInstanceState: Bundle) {
 super.onCreate(savedInstanceState)

 contentView = new SVerticalLayout {
 STextView("Sign in").<<.marginBottom(25 dip).>>.textSize(24.5 sp)
 STextView("ID")
 SEditText()
 STextView("Password")
 SEditText().inputType(TEXT_PASSWORD)
 SButton("Sign in")
 }
```

Currently, this converter is in Alpha stages. The conversion result may omit some proper. Please check the equality of the layout manually.

Fork [this project on Github](#) and please help improve this!

# Preferences

## Plain-old Android API

- ```
val ec = pref.getInt("executionCount", 0)  
val editor = pref.edit()  
editor.putInt("executionCount", ec + 1)  
editor.commit()
```

Scaloid

- ```
val ec = pref.executionCount(0)
pref.executionCount = ec + 1
```

Type dynamic을 응용

예제:

## 안드로이드 앱 평가를 권유하기

```
onCreate {
 val prefs = Preferences()
 val ec = prefs.executionCount(0)
 prefs.executionCount = ec + 1
 if (ec % 20 == 19 && !prefs.wontRate(false)) {
 new AlertDialogBuilder(null, "Please take a moment to rate it.") {
 positiveButton("Rate it", {
 openUri("market://details?id=com.soundcorset.client.android")
 prefs.wontRate = true
 })
 neutralButton("Remind me later")
 negativeButton("No thanks", {
 prefs.wontRate = true
 })
 }.show()
 }
}
```

# 요약

- 좋은 DSL을 설계하는 것은 중요
  - 스칼로이드는 안드로이드 라이브러리를 향상시킴
    - 번잡한 것들을 줄임
    - 관리하기 용이해짐
    - 엄격한 타입 검사
    - 점진적으로 포팅 가능
- <http://scaloid.org>





- 실시간 상담 웹사이트
  - Spring + Scala
  - 5백만 페이지뷰/일
- 상담용 클라이언트
  - Android + Scala



- 메트로놈과 튜너
  - Android + Scala
  - 13만 다운로드