Scala on Android

Painless Android Development with Scala

⇒ 47 Degrees, a global consulting agency & Typesafe Consulting Partner.

@raulraja

@javielinux

@47deg

http://47deg.com/blog

Build Tools

SBT ⇒ Android SDK Plugin

https://github.com/pfn/android-sdk-plugin

SBT ⇒ Android SDK Plugin

- Supports all Android SDK tasks
  - dex
  - ***typedResourcesGenerator***
  - ***proguard***
  - buildConfigGenerator
  - (+ 20... more)

SBT ⇒ Android SDK Plugin ⇒ typedResourcesGenerator

```
object TR {
  val title = TypedResource[TextView](R.id.title)

  object layout {
    val abc_screen_toolbar = TypedLayout[ActionBarOverlayLayout](R.layout.abc_screen_toolbar)
  }
}

class MyActivity extends AppCompatActivity with TypedActivity {

  val titleTextView = findView(title) //titleTextView inferred as TextView, no casting needed

}
```

SBT ⇒ Android SDK Plugin ⇒ *proguard*

Size Matters

Scala byte code size reduced ~ (2.8M)

SBT ⇒ Android SDK Plugin

- ***https://github.com/pfn/android-sdk-plugin***
- Active
- Fast (incremental compilation and proguard caching)
- Proguard + MultiDexApplication integration *(Circumvents 65K method limit)*
- Supports *AAR*, *JAR* and *APK* artifact types

Java Vs Scala ⇒ *NullPointerExceptions*

```
Person person = getPerson();

String name = null;

if (person != null && person.getJob() != null) {
  name = person.getJob().getName();
}

if (name != null) {
  return name;
} else {
  return DEFAULT_NAME;
}
```

Java Vs Scala ⇒ *NullPointerExceptions*

```
val jobName : Option[String] = person.job map (_.name)
```

OR

```
val jobName : Option[String] = for {
   job <- person.job
} yield job.name
```

OR

Option, Try, Either, \/, Validation

Java Vs Scala ⇒ *Contexts*

```
public class MainActivity
    extends Activity {

  public void bar() {
    FooUtils.get(getContext(), R.string.name);
  }

}


public class FooUtils {
  static String get(Context c, int res) {
     return c.getString(res);
  }
}
```

Java Vs Scala ⇒ *Contexts*

```
class MainActivity extends Activity {

   implicit val ctx = getApplicationContext

   def bar = FooUtils.get(R.string.name)

}

object FooUtils {
  def get(res : Int)(implicit ctx : Context) = {
     ctx.getString(res)
  }
}
```

Java Vs Scala ⇒ *Contexts*

```
trait Contexts { self : Activity =>

    implicit val appContext = getApplicationContext

    implicit val activityContext = this

}

class MainActivity extends Activity with Contexts {

   def bar = FooUtils.get(R.string.name)

}

object FooUtils {
  def get(res : Int)(implicit ctx : Context) = {
     ctx.getString(res)
  }
}
```

Java Vs Scala ⇒ *Implicit Classes*

```
public class TextViewUtils {

    public void loadFont(TextView textView, String font) {
        textView.setFontType(font,...)
    }
}

TextViewUtils.loadFont(textView, "robotto");
```

Java Vs Scala ⇒ *Implicit Classes*

```
object Helpers {
  implicit class TextViewHelpers(textView: TextView) {
    def loadFont(font: String) = ???
  }
}

import Helpers._

textView.loadFont(“Roboto.ttf”)
```

Java Vs Scala ⇒ *Async*

```
public class MyTask1 extends AsyncTask<Void, Void, Integer> {
    protected Integer doInBackground(Void... v) {
      return r1;
    }

    protected void onPostExecute(Integer result) {
      new MyTask2().execute(result);
    }
 }

public class MyTask2 extends AsyncTask<Integer, Void, Integer> {
    protected Integer doInBackground(Integer... r1) {
      return r2;
    }

     protected void onPostExecute(Integer result) {
         r1 + r2;
     }
 }

new MyTask1().execute();
```

Java Vs Scala ⇒ *Async*

```
def myTask1: Future[Int] = Future(1)

def myTask2: Future[Int] = Future(2)

def sumResponses: Future[Int] =
  for {
    r1 <- myTaks1
    r2 <- myTaks2
  } yield (r1 + r2)

sumResponses map println
```

Java Vs Scala ⇒ *Async*

```
Future.reduce(List(Future(1), Future(2))) { (a, b) =>
    a + b
}
```

Java Vs Scala ⇒ *Async*

```
Future.reduce(List(Future(1), Future(2))) (_ + _)
```

Java Vs Scala ⇒ *Models*

```java
public class Person {

    private String name;

    private String lastName;

    public String getName() {
        return name;
    }

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

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
}
```

Java Vs Scala ⇒ *Models*

```
case class Person(
  name : String,
  lastName : String)
```

Java Vs Scala ⇒ *Pattern Matching*

```
public boolean onTouchEvent(MotionEvent ev) {
  ...
  switch (action) {
      case MotionEvent.ACTION_DOWN:
          ...
          break;
      case MotionEvent.ACTION_MOVE:
          ...
          break;
      case MotionEvent.ACTION_UP:
      case MotionEvent.ACTION_CANCEL:
          ...
          break;
  }
}
```

Java Vs Scala ⇒ *Pattern Matching*

```
def onTouchEvent(ev: MotionEvent) {
  import MotionEvent._
  action match {
     case ACTION_DOWN => ???
     case ACTION_MOVE => ???
     case ACTION_UP | ACTION_CANCEL => ???
  }
}
```

Java Vs Scala ⇒ *Pattern Matching*

```
person match {
  case Person(_, lastName) if lastName == “Pacheco” =>
    println(“Guapetón”)
  case Person(name, _) if name == “Raúl” =>
    println(“Resultón”)
  case _ =>
    println(“Programadores Java”)
}
```

Java Vs Scala ⇒ *Singletons*

```
public class MySingleton {

  private MySingleton(){}

  public synchronized static MySingleton getInstance() {
        if(instance == null) instance = new MySingleton ();
        return instance;
  }

  public void bar(){
    ...
  }
}
```

Java Vs Scala ⇒ *Singletons*

```
object MySingleton {
  def bar = ...
}
```

Java Vs Scala ⇒ *Mixins*

```
class Mammal

class PlatyPus extends Mammal

new PlatyPus().eggs() <- Compilation Error!!!
```

Java Vs Scala ⇒ *Mixins*

```
trait Mammal

trait EggsSupport {
  def eggs : Eggs
}

class PlatyPus extends Mammal with EggsSupport

PlatyPus().eggs <- Success!!!
```

OS Apps ⇒ *Translate Bubble*

https://play.google.com/store/apps/details?id=com.fortysevendeg.translatebubble

https://github.com/47deg/translate-bubble-android

OS Apps ⇒ *Scala Days Official App*

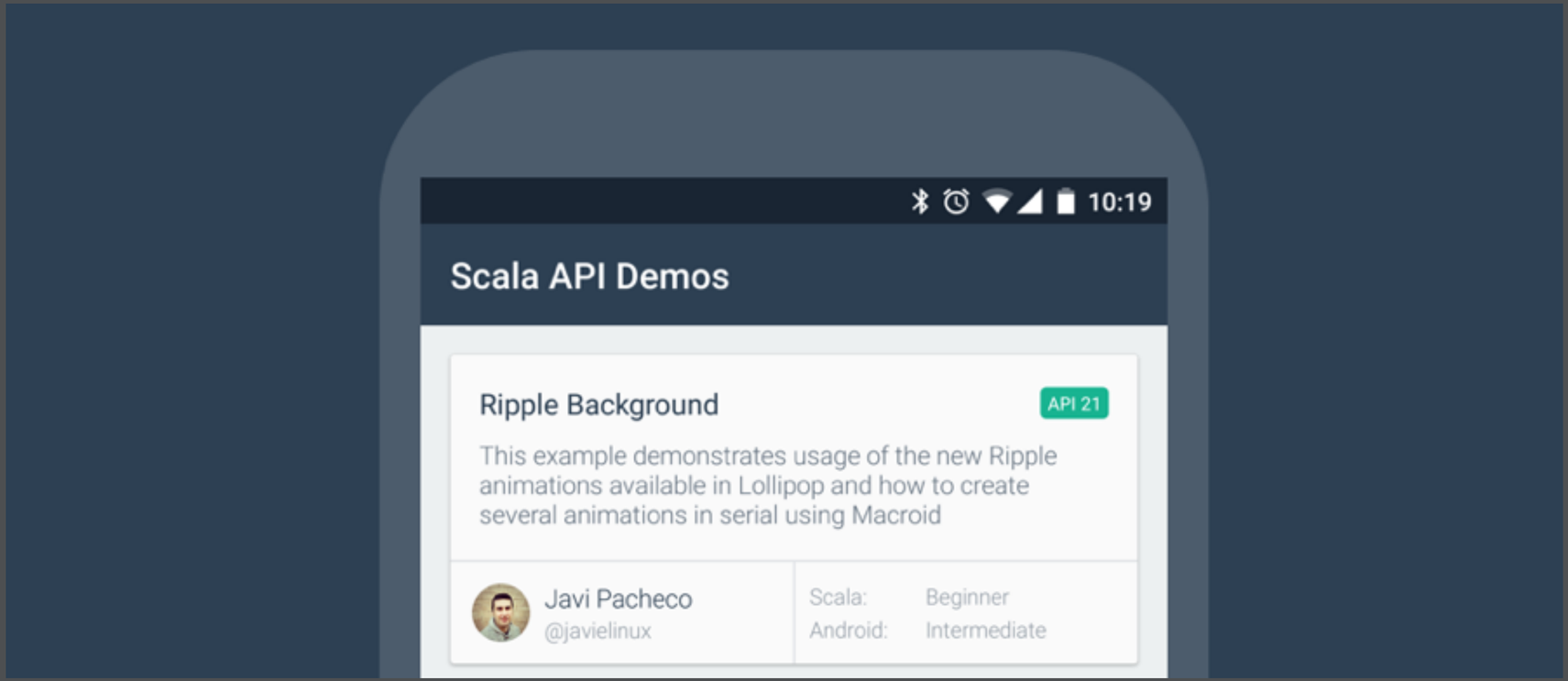
https://play.google.com/store/apps/details?id=com.fortysevendeg.android.scaladays

https://github.com/47deg/scala-days-android

OS Apps ⇒ *Scala API Demos*

https://play.google.com/store/apps/details?id=com.fortysevendeg.scala.android

https://github.com/47deg/scala-android



Thank you

@47deg

@javielinux

@raulraja

http://47deg.com/blog

https://speakerdeck.com/raulraja/painless-android-development-with-scala-deck

https://github.com/47deg/painless-android-development-with-scala-deck