

OPEN PLATFORM

Build applications with **the guardian**

or our subscriptions with **the guardian**



goto;
conference
aarhus

HOW WE MOVED FROM JAVA TO SCALA

Graham Tackley
guardian.co.uk
@tackers

INTERNATIONAL
SOFTWARE DEVELOPMENT
CONFERENCE



gotocon.com

OPEN PLATFORM

Build applications with **the guardian**

open subscriptions with **the guardian**



goto;
conference
aarhus

mostly


HOW WE MOVED FROM JAVA TO SCALA

Graham Tackley
guardian.co.uk
@tackers

INTERNATIONAL
SOFTWARE DEVELOPMENT
CONFERENCE



gotocon.com



Opportunities in Social Work | Devon

Devon
County Council



Newspaper of the year | 7 October 2011 | Last updated less than one minute ago

Your search terms...

The Guardian

Search

the guardian

Weather

London



14°C

8°C

[News](#) | [Sport](#) | [Comment](#) | [Culture](#) | [Business](#) | [Money](#) | [Life & style](#) | [Travel](#) | [Environment](#) | [TV](#) | [Blogs](#) | [Data](#) | [Mobile](#) | [Offers](#) | [Jobs](#)

News | [UK](#) | [World](#) | [Development](#) | [US](#) | [Politics](#) | [Media](#) | [Education](#) | [Society](#) | [Science](#) | [Tech](#) | [Law](#) | [Football](#) | [Guardian](#) | [Observer](#)

Breaking news: Helicopter firm to cut 375 jobs - PA

Osborne defends UK banks after rating downgrade



Chancellor insists UK banks are sound despite Moody's cutting credit rating of 12 institutions

175 comments

- **LIVE blog: latest reaction from the City**
- This is worst financial crisis ever, says Mervyn King
- How the views of the Bank's key players shifted
- **Quantitative easing – the key questions**
- QE2 will damage pensions
- Producer price inflation soars

Twigg may take over from Burnham

Last updated two minutes ago



Move comes in Ed Miliband's first reshuffle which will also see Labour leader lose John Denham and John Healy from frontbench

- **Denham to leave in Labour reshuffle**

How British do you feel?
Send us your videos on nationality and identity



Guardian app
Now on Windows

Nobel peace prize 2011

guardianjobs

Find a job

the networks
the insights
the jobs



Online dating



Find your soulmate with the Guardian's dating site

Join Guardian Soulmates

Moscow & St
Petersburg

History

- Java shop since 2006
- guardian.co.uk: java + spring + velocity + hibernate + oracle

History

- Java shop since 2006
- guardian.co.uk: java + spring + velocity + hibernate + oracle
- ~100k lines production java code

History

- Java shop since 2006
- guardian.co.uk: java + spring + velocity + hibernate + oracle
- ~100k lines production java code
- ~45k lines in velocity templates

History

- Java shop since 2006
- guardian.co.uk: java + spring + velocity + hibernate + oracle
- ~100k lines production java code
- ~45k lines in velocity templates
- ... and ~35k xml

The Reality

- This code base still lives!
- We can still enhance and improve
- We still release (at least) every two weeks

The Reality

- This code base still lives!
- We can still enhance and improve
- We still release (at least) every two weeks

BUT

The Java “BUT”

- Want to innovate faster
- Lots of code in current solution...
- ... takes a while to do what should be simple things ...
- ... and often solving a problem once or twice removed from the actual problem

We tried an alternative...



django

Python + Django

- Easy to learn
- Great web-focused framework, so most things we wanted were out of the box
- Really good documentation & web support

Python + Django

- Easy to learn
- Great web-focused framework, so most things we wanted were out of the box
- Really good documentation & web support

BUT

The Python “BUT”

The Python “BUT”

- Had to throw away years of java experience

The Python “BUT”

- Had to throw away years of java experience
- Dev environment totally different (virtualenv, testing, ci, packaging...)

The Python “BUT”

- Had to throw away years of java experience
- Dev environment totally different (virtualenv, testing, ci, packaging...)
- Runtime behaviour totally different (mod_wsgi, pgbouncer...)

The Python “BUT”

- Had to throw away years of java experience
- Dev environment totally different (virtualenv, testing, ci, packaging...)
- Runtime behaviour totally different (mod_wsgi, pgbouncer...)
- Diagnostic tools totally different (stats capture, heap dumps, stack dumps, logging...)

The Python “BUT”

- Had to throw away years of java experience
- Dev environment totally different (virtualenv, testing, ci, packaging...)
- Runtime behaviour totally different (mod_wsgi, pgbouncer...)
- Diagnostic tools totally different (stats capture, heap dumps, stack dumps, logging...)
- More moving parts (e.g. redis where we'd use singletons on the jvm)

The Python “BUT”

- We lost as much (if not more) time operationally as we saved in development time
- **For us**, switching platform simply did not make sense

Is there a Third Way?

Is there a Third Way?



Is there a Third Way?



Is there a Third Way?



Scala is a statically typed, compiled language
running on the jvm
that “feels” like a dynamic language.

What's so good about Scala?

Same tools, libs, deployment & runtime characteristics as Java

- We still use IntelliJ (with the scala plugin)
- It's just bytecode
- All the experience of running jvm in production still applies
- Can still use java libraries

Mixed Scala-Java Projects *Work*

- Scala compiler can parse java, so can have one project containing both java & scala...
- ... with bi-directional dependencies
- We use this extensively to get the benefits of scala without big bang porting

Much less verbose

```
package net.tackley;

public class WelcomeClass {
    private String name;

    public WelcomeClass(String name) {
        this.name = name;
    }

    public String sayIt() {
        return "Hello " + name;
    }
}
```

=>

```
package net.tackley

class ScalaWelcomeClass(name: String) {
    def sayIt = "Hello " + name
}
```


Type inference

```
ArrayList<String> myList = new ArrayList<String>();
```

=>

```
val myList = new ArrayList[String]
```

Console / REPL

- Encourages ipython / irb attitude
- Awesome for experimentation and exploration

Powerful collections library

```
def convertTags(document: SolrDocument, apiRequest: ShowParameters): List[Tag] = {  
  apiRequest.showTags match {  
    case Nil => Nil  
    case requiredTagTypes =>  
      document.stringFields("tag-ids")  
        .flatMap(tagRepository.tag(_))  
        .filter(tag => requiredTagTypes.contains(tag.itemType))  
        .toList  
  }  
}
```

Our First Scala Project: Content API

Content API

- Provide API to access all of our website content
- Limited beta released early 2009
- Started implementation of final version late 2009
- Live May 2010:
<http://content.guardianapis.com>

```
content.guardianapis.com/search.json?q=aarhus&show-fields=all

{
  - response: {
    status: "ok",
    userTier: "free",
    total: 174,
    startIndex: 1,
    pageSize: 10,
    currentPage: 1,
    pages: 18,
    orderBy: "newest",
  - results: [
    - {
      id: "football/2011/jun/25/spain-switzerland-european-under-21-championship",
      sectionId: "football",
      sectionName: "Football",
      webPublicationDate: "2011-06-25T22:46:20+01:00",
      webTitle: "Spain beat Switzerland to secure European Under-21 Championship title",
      webUrl: "http://www.guardian.co.uk/football/2011/jun/25/spain-switzerland-european-under-21-championship",
      apiUrl: "http://content.guardianapis.com/football/2011/jun/25/spain-switzerland-european-under-21-championship",
      - fields: {
        trailText: "Goals from Ander Herrera and Thiago Alcântara secured a third European Under-21 Championship title for Spain",
        headline: "Spain beat Switzerland to secure European Under-21 Championship title",
        strap: "European Under-21 Championship final",
        lastModified: "2011-06-25T23:14:35+01:00",
        hasStoryPackage: "false",
        standfirst: "* Switzerland 0-2 Spain",
        shortUrl: "http://gu.com/p/3v639",
        thumbnail: "http://static.guim.co.uk/sys-images/Football/Clubs/Club_Home/2011/6/25/1309034707629/Thiago-Alcantara-is-congr-003.jpg",
        commentable: "true",
        byline: "Agencies",
        publication: "guardian.co.uk",
        shouldHideAdverts: "false",
        liveBloggingNow: "false"
      }
    },
    - {
      id: "science/2011/jun/05/scientists-create-antimatter-study",
      sectionId: "science",
      sectionName: "Science",
      webPublicationDate: "2011-06-05T17:59:00+01:00",
      webTitle: "Scientists trap antimatter long enough to study how it works",
      webUrl: "http://www.guardian.co.uk/science/2011/jun/05/scientists-create-antimatter-study",
      apiUrl: "http://content.guardianapis.com/science/2011/jun/05/scientists-create-antimatter-study",
      - fields: {
        newspaperPageNumber: "4"
      }
    }
  ]
}
```

2009

2010

2011

Nov

- Started implementation
- java + guice + guice servlets + apache solr
- 3-4 java devs

2009

2010

2011

Nov

- A few people had played with scala
- No production scala code

2009

2010

2011

Jan

- Integration tests with ScalaTest
- maven-scala-plugin

2009

2010

2011

```
@RunWith(classOf[JUnitRunner])
abstract class PaginationTest extends FeatureSpec with ShouldMatchers {
  protected def formatString: String

  feature("search pagination") {
    scenario("number of results should be displayed") {

      val results = Api.searchQuery
        .withFormat(formatString)
        .withUserTier("free")
        .search

      results.total should be > (0)
    }
  }
}

class XmlPaginationTest extends PaginationTest {
  protected override def formatString = "xml"
}

class JsonPaginationTest extends PaginationTest {
  protected override def formatString = "json"
}
```


2009

2010

2011

Jan

A fair bit of our test code was java-without-semicolons as we learnt about Scala...

2009

2010

2011

Ian

```
private def bootstrap: Unit = {
  info("***** BOOTSTRAPPING A LOCAL COPY OF THE API BECAUSE NOTHING IS LISTENING ON PORT %d ***"
    startJetty
    populateIndex
    info("***** YOUR LOCAL COPY OF THE API IS NOW RUNNING ON PORT %d *****" format PORT)
}

private def populateIndex: Unit = {
  val server = new CommonsHttpSolrServer("http://localhost:" + PORT + "/solr")
  val query = new SolrQuery("*:~")
  var response = server.query(query)
  var found = response.getResults.getNumFound
  if (found > 100) {
    info("Not rebuilding index because it contains %d items already" format found)
    return
  }
  info("Populating index")
  try {
    IndexBuild.main("-t 10 -b 3000 -f".split("\\s"))
  }
  catch {
    case e: Exception => {
      info("INDEX POPULATION FAILED: " + e)
      e.printStackTrace
      throw e
    }
  }
}
```

2009

2010

2011

Feb

... despite that we loved it so much that after a month we decided to convert the whole app to Scala

java + guice + guice servlets + apache solr

2009

2010

2011

Feb

... despite that we loved it so much that after a month we decided to convert the whole app to Scala

scala + guice + guice servlets + apache solr

2009

2010

2011

May

Live!

2009

2010

2011

Jul

Switched from maven to simple-build-tool
Mainly for incremental compilation

2009

2010

2011

Jul

scala + guice + guice servlets + apache solr

2009

2010

2011

Jul

scala + lift + apache solr

5k loc => 3.5k (mostly due to writing better scala)

2009

2010

2011

Today

- Scala used by default on all new jvm-based projects
- Still do some (externally-hosted) new things in python
- Team of 20 ex-java devs all audibly groan when working on java rather than scala

The Scala “BUT”?

The Scala “BUT”?

But, isn't Scala really complex?

The Scala “BUT”?

```
public static <T,U> T[] copyOf(U[] original, int newLength, Class<? extends T[]> newType) {  
    T[] copy = ((Object)newType == (Object)Object[].class)  
        ? (T[]) new Object[newLength]  
        : (T[]) Array.newInstance(newType.getComponentType(), newLength);  
    System.arraycopy(original, 0, copy, 0,  
        Math.min(original.length, newLength));  
    return copy;  
}
```

This example taken from Cay Horstmann

<http://weblogs.java.net/blog/cayhorstmann/archive/2011/10/05/javaone-2011-day-3>

The Four Scala Features That Scare Java Developers...

The Four Scala Features That Scare Java Developers...

... can all be explained in one slide each

Symbolic method names

```
class Complex(real: Double, imag: Double) {  
  def +(other: Complex) = ....  
}
```


Infix & Postfix Operations

- $\text{obj.op}(x)$ can be written $\text{obj op } x$
- obj.op can be written obj op

Infix & Postfix Operations

- `obj.op(x)` can be written `obj op x`
- `obj.op` can be written `obj op`

So given:

```
val a = new Complex(1, 3)
```

```
val b = new Complex(3, 4)
```

Can write:

```
a.+(b)
```

as

```
a + b
```

Higher order functions

```
scala> def apply(f: Int => Int, v: Int) = f(v)  
apply: (f: Int => Int, v: Int)Int
```

```
scala> apply(i => i * 2, 7)  
res1: Int = 14
```


Higher order functions

```
scala> def apply(f: Int => Int, v: Int) = f(v)  
apply: (f: Int => Int, v: Int)Int
```

```
scala> apply(i => i * 2, 7)  
res1: Int = 14
```

```
scala> apply(_ * 2, 7)  
res2: Int = 14
```

Higher order functions (ii)

// Java

```
Integer run(Callable<Integer> fn) throws Exception {  
    return fn.call();  
}
```

```
Integer processInt(final Integer i) throws Exception {  
    return run(new Callable<Integer>() {  
        public Integer call() throws Exception {  
            return i + 1;  
        }  
    });  
}
```

Higher order functions (ii)

// Java

```
Integer run(Callable<Integer> fn) throws Exception {  
    return fn.call();  
}
```

```
Integer processInt(final Integer i) throws Exception {  
    return run(new Callable<Integer>() {  
        public Integer call() throws Exception {  
            return i + 1;  
        }  
    });  
}
```

// Scala

```
def run(f: => Int) = f  
def processInt(i: Int) = run(i + 1)
```


Implicit Conversion

- given an in-scope declaration
`implicit def conv(a: A): B`
- “conv” will be called whenever you have an A and need a B
- or you call a method on an instance of A that doesn't exist on A but does on B

Writing Good Scala

Writing Good Scala

- Express your Intent. Simply.

Writing Good Scala

- Express your Intent. Simply.
- The pram is full of toys.
Use only to achieve the above.

Demo

Summary

- Smooth migration path from Java
- Take it easy, and don't fear java-without-semicolons in the early days
- You'll lose if you stay there though!
- Incrementally embrace Scala features to increase readability
- Expect to keep learning

OPEN PLATFORM

Build applications with **theguardian**

- <http://content.guardianapis.com>
- <http://www.guardian.co.uk/open-platform>
- <http://github.com/guardian/open-platform-content-api-scala-client>

graham.tackley@guardian.co.uk ■ @tackers