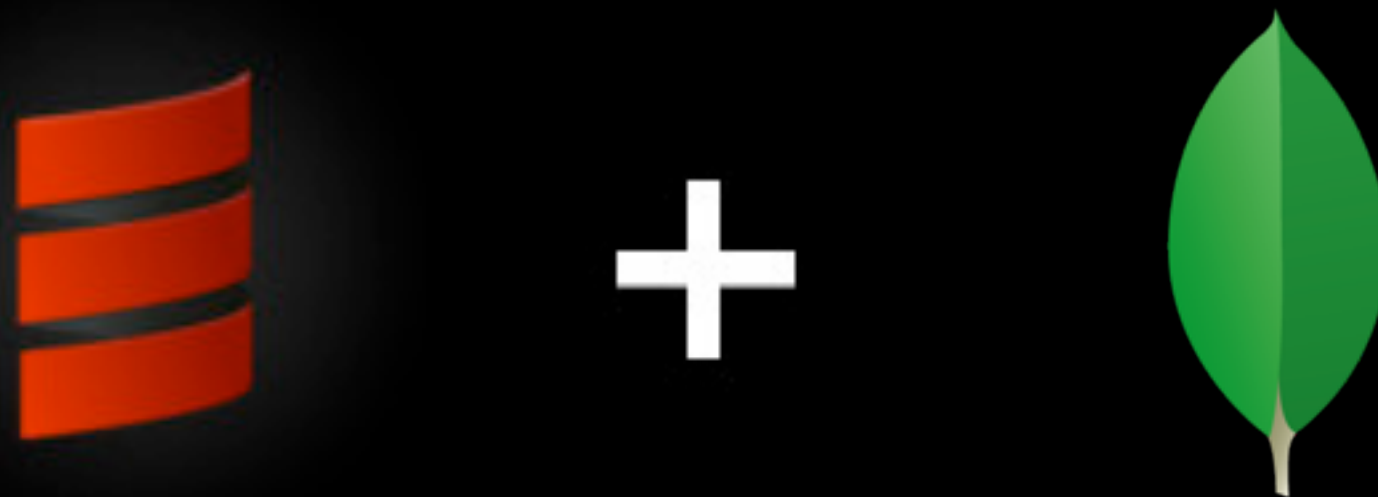


# Scala DSLs

## Dissecting Rogue

**Konrad Malawski** / @ktosopl  
GDG / PJUG / KSUG / SCKRK  
ScalaCamp 23.01.2012



# Scala DSLs

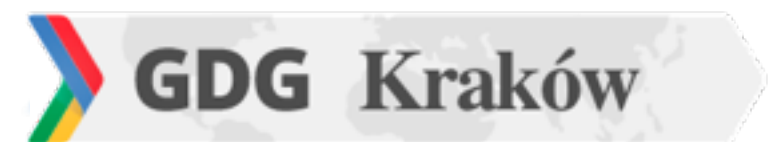
## Dissecting Rogue

**Konrad Malawski** / [@ktosopl](#)  
[GDG](#) / [PJUG](#) / [KSUG](#) / [SCKRK](#)  
[ScalaCamp](#) 23.01.2012

**Łukasz Dubiel**  
[@bambuchaAdm](#)  
[Hackerspace Kraków](#)



Konrad Malawski  
github: ktoso  
twitter: ktosopl



**KRAKOW SCALA**  
USER GROUP

**D  
S  
L**

**D  
S  
L**

**D  
S  
L**

# Domain

# S

# L

**Domain  
Specific  
L**



# **Domain Specific Language**

# Domain Specific Language

(internal)

```
list    should contain (1)
pet     should equal  ("Kapibara")
```

( ScalaTest )

# Domain Specific Language

(internal? external?)

```
create random Client
```

```
create Client from Poland having PremiumSubscription
```

# Domain Specific Language

(internal? external?)

```
find all clients withMore than one negative comments
```

# Domain Specific Language

=

# Domain Specific Language

=

```
def internal =  
    "defined within our programming language"
```

# Domain Specific Language

=

```
def internal =  
    "defined within our programming language"
```

```
define External =  
    /* AKA */    "let's write a parser"
```

# Querying Mongo



# Querying Mongo

# Querying Mongo

# Querying Mongo

QL = Plain JSON

# Querying Mongo

QL = Plain **JSON**  
**JavaScript** Console

# Querying Mongo

QL = Plain JSON

JavaScript Console

Java Driver

# Querying Mongo

QL = Plain **JSON**

**JavaScript** Console

**Java** Driver

**Casbah** = **Scala** Driver

# Querying Mongo

QL = Plain **JSON**

**JavaScript** Console

**Java** Driver

**Casbah** = **Scala** Driver

... used by **Rogue**

# Querying Mongo: JS



Console API



# Querying Mongo: JS

```
db.inventory.find( {} )
```

# Querying Mongo: JS

```
db.inventory.find( {} )
```



**“find all”**

# Querying Mongo: JS

```
db.inventory.find(  
  {  
    qty: 20  
  }  
)
```

# Querying Mongo: JS

```
db.inventory.find(  
  {  
    qty: 20  
  }  
)
```

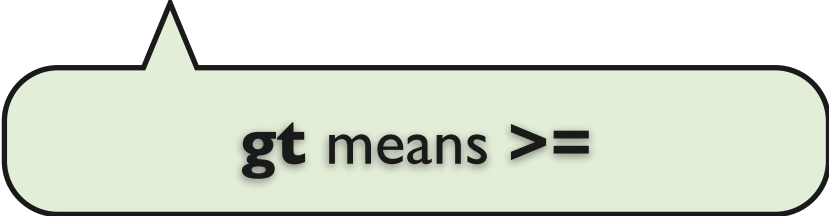
"where **qty = 20**"

# Querying Mongo: JS

```
db.inventory.find(  
  {  
    qty: { $gt: 20 }  
  }  
)
```

# Querying Mongo: JS

```
db.inventory.find(  
  {  
    qty: { $gt: 20 }  
  }  
)
```



**gt** means **>=**

# Querying Mongo: JS

```
db.pople.find(  
  {  
    age: { $gte: 20 },  
    $or: [  
      { name: "Ken" },  
      { name: "Kenshiro" },  
      { city: "Tokyo" },  
      ...  
    ]  
  }  
)
```

# Querying Mongo: JS

```
db.pople.find(  
  {  
    age: { $gte: 20 },  
    $or: [  
      { name: "Ken" },  
      { name: "Kenshiro" },  
      { city: "Tokyo" },  
      ...  
    ]  
  }  
)
```

**\$or** may contain **any clause**



# Querying Mongo: Java



Java API  
(Morhpia)

# Querying Mongo: Java

```
ds.find(Employee.class).field("manager").equal(null).get();
```

(morphism)

# Querying Mongo: Java

```
ds.find(Employee.class).field("manager").equal(null).get();
```



You call this typesafe!?

(morphism)

# Mongo with **Rogue**



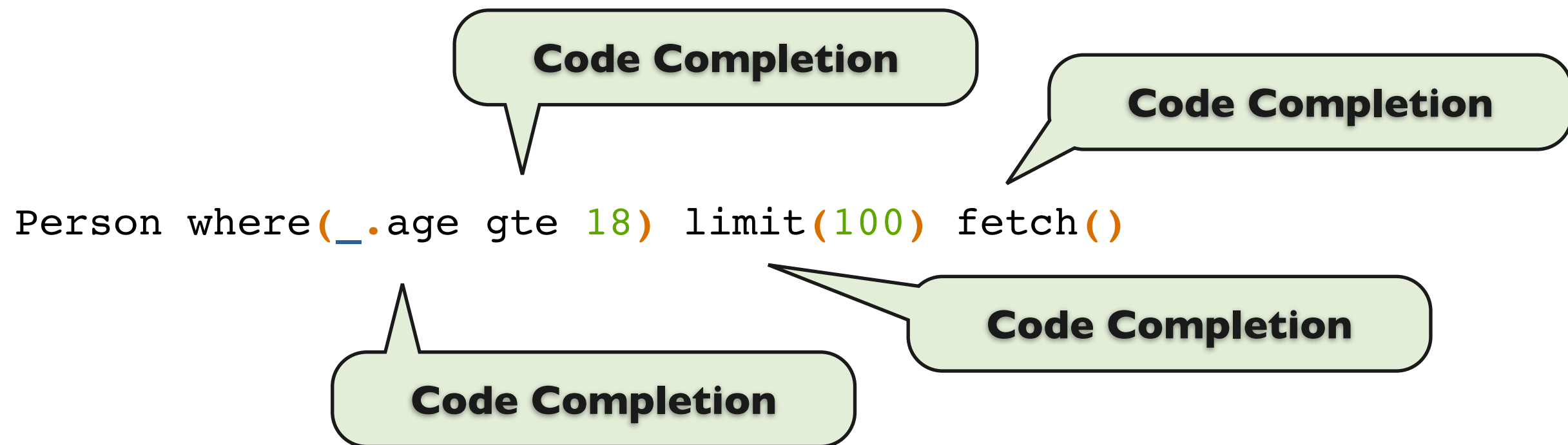
# Rogue in Action

```
import com.foursquare.rogue.Rogue._
```

# Rogue in Action

```
Person where(_.age gte 18) limit(100) fetch()
```

# Rogue in Action



# Rogue in Action

```
Person where(_.age eqs "Bazinga") fetch()
```



# Rogue in Action

**eqs** means **==**

Person where `(_.age eqs "Bazinga")` fetch()

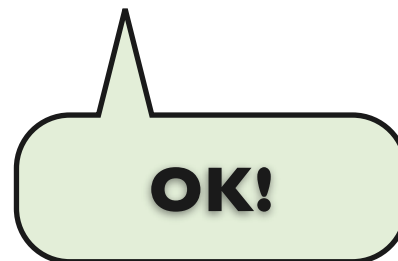
**Won't compile - age is IntField**

# Rogue in Action

```
Person where(_.age gte 18) fetch()
```

# Rogue in Action

Person where(\_.age gte 18) fetch()



# Rogue in Action

```
Person where(_.age gte 18) limit() fetch()
```

# Rogue in Action

Person where(\_.age gte 18) limit() fetch()



Limit

# Rogue in Action

```
val _: List[String] =  
    Person where(_.age gte 18) select(_.name) fetch()
```

# Rogue in Action

**Type Safe!**

```
val _: List[String] =  
    Person where(_.age gte 18) select(_.name) fetch()
```

# Rogue in Action

```
Person where( _.age gte 18) limit(100) foreach { p =>  
    println(s"Name: ${p.firstName}")  
}
```



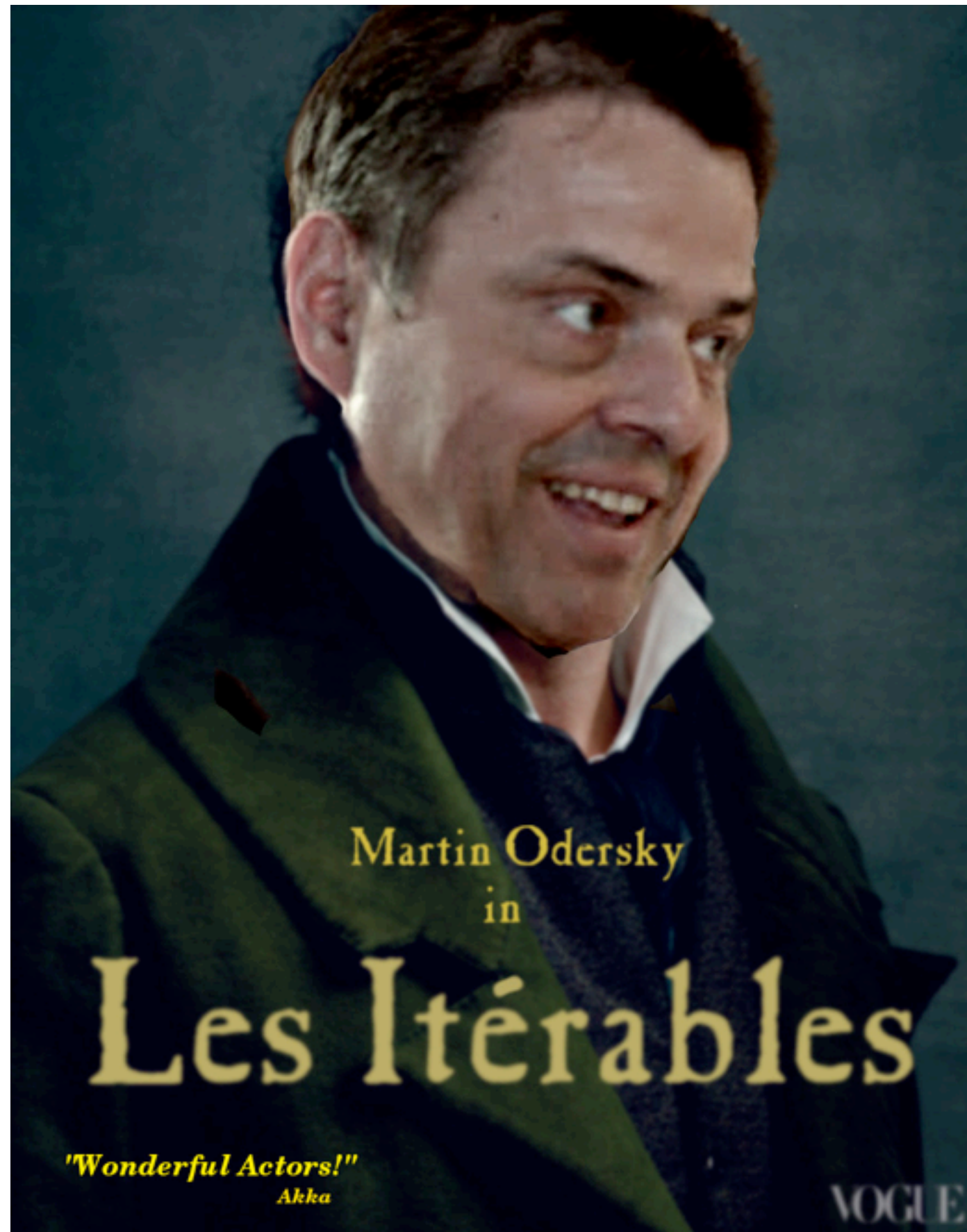
# Rogue in Action



**foreach** found person

```
Person where( _.age gte 18) limit(100) foreach { p =>  
  println(s"Name: ${p.firstName}")  
}
```

people foreach { }



# Rogue in Action

```
Person or(  
  _.where(_.name eqs "Ken"),  
  _.where(_.name eqs "ken")  
  _.whereOpt(Option("Kenshiro"))(_.name eqs _))  
) get()
```

# Rogue in Action

```
Person or(  
  _.where(_.name eqs "Ken"),  
  _.where(_.name eqs "ken")  
  _.whereOpt(Option("Kenshiro"))(_.name eqs _))  
) get()
```



**Get me one Person.**

# Rogue in Action

```
Person or(  
  _.where(_.name eqs "Ken"),  
  _.where(_.name eqs "ken")  
  _.whereOpt(Option("Kenshiro"))(_.name eqs _))  
) get()
```

**Get me one Person.**

Issue #69: fix in my pull req

# Live Hacking

# Live Hacking

# Live Hacking



# Live Hacking

**with** ŁukaszDubiel

# Live Hacking

*with* ŁukaszDubiel

## Advanced Stuff.

# Live Hacking

**with** ŁukaszDubiel

Advanced Stuff.  
Fast.

# Live Hacking

**with** ŁukaszDubiel

Advanced Stuff.

Fast.

Ask Questions!

# Talk is cheap. Show me the code.

— Linus Torvalds



# def links =

- **Scala Lang** <http://www.scala-lang.org/>
- **Scala Koans** <http://www.scalakoans.org>
- **Blog.Project I 3.pl** - <http://www.blog.projectI3.pl>
- **MongoDB** ref - [docs.mongodb.org/manual/reference/operators/](http://docs.mongodb.org/manual/reference/operators/)
- Foursquare **Rogue** - <https://github.com/foursquare/rogue>
- Java **Morphia** - <http://code.google.com/p/morphia/>
- **Kraków Scala UG** - <http://krakowscala.pl> / <http://scalacamp.pl>

# def links =

- **Scala Lang** <http://www.scala-lang.org/>
- **Scala Koans** <http://www.scalakoans.org>
- **Blog.Project I 3.pl** - <http://www.blog.projectI3.pl>
- **MongoDB** ref - [docs.mongodb.org/manual/reference/operators/](http://docs.mongodb.org/manual/reference/operators/)
- Foursquare **Rogue** - <https://github.com/foursquare/rogue>
- Java **Morphia** - <http://code.google.com/p/morphia/>
- **Kraków Scala UG** - <http://krakowscala.pl> / <http://scalacamp.pl>

Mailing lists rock!





Thanks!  
Dziękuję!  
ありがとう!

カピバラさん



**Konrad Malawski** / @ktosopl  
GDG / PJUG / KSUG / SCKRK  
ScalaCamp 23.01.2012

**Łukasz Dubiel**  
@bambuchaAdm  
Hackerspace Kraków



Thanks!  
Dziękuję!  
ありがとう!

カピバラさん



I love feedback! <3

**Konrad Malawski** / @ktosopl  
GDG / PJUG / KSUG / SCKRK  
ScalaCamp 23.01.2012

**Łukasz Dubiel**  
@bambuchaAdm  
Hackerspace Kraków