LUNATECH

RESEARCH

Play framework and Scala

2011-08-12 – Erik Bakker - @eamelink - lunatech.com

What is Scala?

Scala is a general purpose programming language designed to express common programming patterns in a concise, elegant, and type-safe way.

LUNATECH
RESEARCH

What is Scala?

Scala is a general purpose programming language designed to express common programming patterns in a concise, elegant, and type-safe way.

Concise & elegant: Unlike Java

LUNATECH
RESEARCH

What is Scala?

Scala is a general purpose programming language designed to express common programming patterns in a concise, elegant, and type-safe way.

Concise & elegant: Unlike Java

Type-safe: Unlike Python, Ruby, PHP etc.

LUNATECH
RESEARCH

Scala

Language that runs on the JVM

Compiles to regular .class files

Interoperable with Java

- **Call Java code from Scala**
- **Call certain Scala code from Java**

LUNATECH
RESEARCH

Ways to use Scala in Play

Use a library written in Scala

- **Just put the jar in your lib dir**

LUNATECH
RESEARCH

Ways to use Scala in Play

Use a library written in Scala

- **Just put the jar in your lib dir**

Write some parts of your Play app in Scala

- **Install the scala module, and create scala files**

LUNATECH
RESEARCH

Ways to use Scala in Play

Use a library written in Scala

- **Just put the jar in your lib dir**

Write some parts of your Play app in Scala

- **Install the scala module, and create scala files**

Use Play's Scala API, templates and DAL

- **Write controllers and models in scala and views in scala templates.**

LUNATECH
RESEARCH

Play Scala Module

Play Scala module provides:

Scala compiler

Scala API, taylored to power of Scala

Anorm SQL data access layer

Type-safe templating system

LUNATECH
RESEARCH

Play Scala Module

Play Scala module provides:

Scala compiler

Scala API, taylored to power of Scala

Anorm SQL data access layer

Type-safe templating system

Fix the bug, reload & wait cycle

LUNATECH
RESEARCH

Sort of outline

Controllers

Templates

Anorm

LUNATECH
RESEARCH

Controllers

Scala objects, extending Controller

Actions return values like Ok, Html, Text, Json, Action or Redirect

Or plain strings, xml or binary streams

Composition with traits

LUNATECH
RESEARCH

Controllers

```
object App extends Controller {

  def index = "Hello!"

  def greeter(name: String = "world") = {
    <h1>Hello {name}</h1>;
  }

  def greeter2(name: Option[String]) = {
    "Hello " + name.getOrElse("you");
  }

}
```

LUNATECH
RESEARCH

Templates

Scala based, very concise

Type safe

Compile to plain Scala

Very fast

Easily composable

LUNATECH
RESEARCH

Templates

```
@(customer:models.Customer, orders:Seq[models.Order])

<h1>Welcome @customer.name!</h1>

@if(orders) {

    <h2>Here is a list of your current orders:</h2>

    <ul>
    @orders.map { order =>
        <li>@order.title</li>
    }
    </ul>

} else {
    <h2>You don't have any order yet...</h2>
}
```

LUNATECH
RESEARCH

Template composition

main.scala.html:

```
@(title:String)(content: => Html)

<h1>@title</h1>
<div id="main">
   @content
</div>
```

Part of Application/index.scala.html:

```
@main(title = "Home") {
  <h1>Home page</h1>
}
```

LUNATECH
RESEARCH

Tags

In some template:

```
@notice("error") { color =>
    Oops, something is <span style="color:@color">wrong</span>
}
```

In tags/notice.scala.tag:

```
@(level:String = "error")(body: (String) => Html)

@level match {
    case "success" => {
        <p class="success">@body("green")</p>
    }

    case "error" => {
        <p class="error">@body("red")</p>
    }
}
```

LUNATECH
RESEARCH

Anorm philosophy

ORM is required in java

- **Plain JDBC is cumbersome, with checked exceptions and messy data transformations**
- **But JPA is not expressive**

There's a great and expressive DSL for databases...

LUNATECH
RESEARCH

Anorm philosophy

ORM is required in java

- **Plain JDBC is cumbersome, with checked exceptions and messy data transformations**
- **But JPA is not expressive**

There's a great and expressive DSL for databases... SQL

LUNATECH
RESEARCH

Anorm philosophy

ORM is required in java

- **Plain JDBC is cumbersome, with checked exceptions and messy data transformations**
- **But JPA is not expressive**

There's a great and expressive DSL for databases... SQL

And Scala is great for transforming data

LUNATECH
RESEARCH

Anorm

SQL Strings, easy, expressive and powerful but not type safe

Pattern matching or parser combinators for retrieval

Magic helper to automatically create parsers for case classes

LUNATECH
RESEARCH

Anorm

```
case class Country(
   code: Id[String],
   name: String,
   population: Int,
   headOfState: Option[String]
)

object Country extends Magic[Country]

val countries:List[Country] = SQL("select *
     from Country").as(Country*)
```

LUNATECH
RESEARCH

Finally

Using JPA?

Using modules?

Writing a module?

Tool support?

Should you use it in production?

Roadmap

LUNATECH
RESEARCH

```
scala> slides.filter(!_.done).length
res0: Int = 0
```

erik.bakker@lunatech.com

@eamelink

www.lunatech.com

LUNATECH
RESEARCH