Play Framework

Presentation version 0.1

Mozart Brocchini

Mozart Brocchini

- Self taught programming as a teenager
- Software Engineer at Human Genome Sequencing Center with Baylor College of Medicine
- Writing java apps for the last 13 years
- Used Professionally: Java, Groovy, JavaScript, Pascal, LISP, C, BASIC
- Had fun with : Lua, Scala
- http://www.linkedin.com/in/brocchini

What you will learn

- How different Play is from most other java frameworks
- Tour features with live demos
- Comparison between Play 2 and Play 1
- Q&A

Why Play ?

- Simplicity
- Java

But how simple should it be ?

Albert Einstein

- "Everything should be made as simple as possible, but no simpler."

Key differences in favor of simplicity

- Shortens “code-compile-deploy-test” cycle - No restart !
- No Servlets. No XML
- HTTP to code mapping
- Static methods
- No getters or setters in POJOs
- Built in REST support

More differences in favor of simplicity

- Stateless MVC
- Full Stack: Server, JPA, Templates Engine, Internationalization, Testing Environment, Database Evolutions, Embedded Database, WS API, Functional API, image manipulation API …
- There is more...
  Modules ! CRUD, Security, Social Authentication, GWT, Mongo DB, RabbitMQ, Simple Search, Elastic Search...

More Simplicity

- Easy to scale, AKKA, Built in distributed Cache, Elastic Search, No SQL, Suspendable Requests.
- Easy bug fixes with on the browser error tips and stripped down stack traces.
- Multi environment configuration in a single file
- Seamless integration with IDEs

Demos

- Go to terminal
- Setup new app `$ play new`
- Run app `$ play run`
- Help `localhost:9000/@documentation`
- Ide integration `$ play eclipsify | play idealize | play netbensify`
- Save and hit reload

Sample App

- Generic web service to translate low level system errors
- Ex:

Convert from : “`blah bla ORA-0001 java.sql.SQLException something is not right`”

To friendlier version : “`Sorry database is too busy`” :)

Demo Sample App

- Embedded database `localhost:9000/@db user=sa, password=blank`
- Testing on web browser `localhost:9000/@tests`
- Headless test `$ play auto-test`
- http to code mapping
- JPA models
- Enhanced POJOS
- Modules

Play 1.x Versus Play 2.x

Play 1

- From: http://www.playframework.org/documentation/1.2.4/overview
- “The Play framework is a clean alternative to bloated Enterprise Java stacks. It focuses on developer productivity and targets RESTful architectures. Play is a perfect companion to agile software development.”
- The Play framework’s goal is to ease web applications development while sticking with Java. Let’s see how this is possible.

Play 2

- From: http://www.playframework.org/
- “The Play framework makes it easier to build web applications with Java & Scala.”
- “Play is based on a lightweight, stateless, web-friendly architecture and features predictable and minimal resource consumption (CPU, memory, threads) for highly-scalable applications - thanks to its reactive model, based on Iteratee IO.”

1.2.x versus 2.0 Matrix

	Play 1.2.4	**Play 2.0**
Framework written in	Java	Scala
Applications language	Java / Scala (module)	Java / Scala
Templates language	Groovy / Others (module)	Scala / Others (module)
Routes	Plain text file	Compiles to Scala (adds type safety, gives up speed in dev mode)
Build	Minimalistic Python script	SBT (gives up simplicity in favor of gaining a lot of power)
Deployment	Framework / WAR (gives up asynchronous IO)	JAR **REVIEW **
Data Storage	JPA(Enhanced Hibernate) / No SQL (module)	JPA, Ebean, Anorm / No SQL (module)

Thank You !

- https://github.com/brocchini
- http://www.linkedin.com/in/brocchini
- http://mozartmb.wordpress.com/