

Play Framework

Columbus Scala Enthusiasts

March 14, 2016

Now owned by Lightbend, formerly Typesafe

Been around since mid-2010

Based on Activator

Demonstrate Activator

- activator new

- >400 templates – too many, use for example only

- activator ui

- useful for demo or learning only: Eclipse and IntelliJ both better

- activator run

- probably what will spend the most time running; restart when app configuration changes

- activator test

- useful when IDE gets confused

Demonstrate application

- try blank name

- try script injection

- play a few games

Examine gross structure

- First, app subdirectory

- controllers

- views

other classes

Then, conf directory

application.conf

routes

public directory

test directory

build.sbt

./activator and ./activator.bat

Follow app design in more detail

GameController.scala

trait/object arrangement: see test

public interface creates and returns Actions

Action constructor can take closure with or without request

Closure returns Result: shortcuts Ok, BadRequest, and Redirect

Legacy practice: use Global to hold singletons

Current practice: DI with Guice

WebSocket treated differently

Form validation, reverse routing

Views

front_page.scala.html

game_page.scala.html

Global

Lifecycle: see superclass

Special structure for testing

StatisticsService

Anorm: better than ORM

Special structure for testing: parameterize function, anyone?

Tests for StatisticsService

IncomingActor

Relationship with GameActor

GameController.socket()

Utils.js

JsValue library

PoisonPill closes WebSocket

GameActor

PlayerStates, nextId, gameId

Async communication with DatabaseActor

DatabaseActor

createTables: hack.

Special structure for testing

Anorm: writing data

evolutions