

# Ratpack 101

Álvaro Sánchez-Mariscal  
Web Architect - **odobo**



# About me

- **Passionate Software Developer.**
  - Former BEA/Sun instructor, Groovy fanboy since 2006.
- Speaker at Codemotion, JavaLand, **GeeCON**, Spring IO, GR8Conf, GGX and Greach.
- Author of several Grails plugins.
- Working now at **odobo** as a Web Architect.

# About odobo

- **HTML5 platform** for gambling games.
- We provide game developers a **Javascript SDK**.
- Server side **logic and maths** are handled by our industry **certified game engines**.
- **Seamless integration** with several casinos.
- Check out **[play.odobo.com](http://play.odobo.com)** and play for free!

# Agenda

- Introduction.
- The handlers chain.
- The context.
- Ratpack modules.
- Asynchronous & Non Blocking model.
- Q&A.

# Agenda

- Introduction.
- **The handlers chain.**
- The context.
- Ratpack modules.
- Asynchronous & Non Blocking model.
- Q&A.

# Agenda

- Introduction.
- The handlers chain.
- **The context.**
- Ratpack modules.
- Asynchronous & Non Blocking model.
- Q&A.

# Agenda

- Introduction.
- The handlers chain.
- The context.
- **Ratpack modules.**
- Asynchronous & Non Blocking model.
- Q&A.

# Agenda

- Introduction.
- The handlers chain.
- The context.
- Ratpack modules.
- **Asynchronous & Non Blocking model.**
- Q&A.

# Agenda

- Introduction.
- The handlers chain.
- The context.
- Ratpack modules.
- Asynchronous & Non Blocking model.
- Q&A.

# Slides, code and workshop available

Fork me on GitHub

- Slides:
  - <http://www.slideshare.net/alvarosanchezmariscal/>
- Companion code:
  - <https://github.com/alvarosanchez/ratpack-101-talk>
- Workshop:
  - <https://github.com/alvarosanchez/ratpack-101>

# Before we start

- Raise your hand if you know:
  - Java.
  - Groovy.
  - Spock.
  - Ratpack?

# Introduction

- Ratpack is:
  - A set of Java 8 libraries...
  - ... lightweight, un-opinionated, Netty-based...
  - ... for writing non-blocking HTTP applications...
  - ... focused on performance and efficiency...
  - ... with an emphasis on testability..
  - ... embracing Java 8 or Groovy as programming languages.

# When to use Ratpack?

- For microservices.
- For high performance applications.
- For lightweight services (no container, batteries included).

# Ratpack is developer friendly

- Startup in milliseconds.
- Development-time reloading with Spring Loaded.
- First-class support for functional testing.

# Demo

1. Groovy script.
2. Gradle build with Lazybones.

# Agenda

- Introduction.
- **The handlers chain.**
- The context.
- Ratpack modules.
- Asynchronous & Non Blocking model.
- Q&A.

# Handlers

- An incoming request is passed through the handler chain.
- A handler can:
  - Respond to the request.
  - Delegate to the next handler in the chain.
  - Insert more handlers in the chain and delegate to them.

# Handlers

- Each handler is given a Context instance.
  - To interact with the request / response objects.
  - To access the registry.
  - Etc.
- Note that Ratpack is not based on the Servlet API.

# Demo

## 3. Handlers.

# Agenda

- Introduction.
- The handlers chain.
- **The context.**
- Ratpack modules.
- Asynchronous & Non Blocking model.
- Q&A.

# The Context

- The Context provides:
  - Access to the HTTP Request and Response objects.
  - Interaction with other handlers through `next()` and `insert()` methods.
  - Access to objects in the registry.

# The registry

- The context is also a Registry of objects.
- Handlers can store objects into the context for use by downstream handlers.

# Demo

## 4. Working with the context.

# Agenda

- Introduction.
- The handlers chain.
- The context.
- **Ratpack modules.**
- Asynchronous & Non Blocking model.
- Q&A.

# Ratpack modules

- Ratpack can integrate with Google Guice for:
  - Dependency injection for services, etc.
  - Officially supported modules.
  - Reusing your own components as your modules.

# Official modules (1/2)

- Benchmarks (JMH).
- Metrics.
- H2 Database.
- Handlebars.
- HikariCP JDBC pool.
- Hystrix.
- Jackson.
- New Relic.
- Pac4J.
- Reactor.
- Groovy Remote Control.

# Official modules (2/2)

- RxJava.
- HTTP Sessions.
- Spring Boot.
- Thymeleaf.

# Demo

## 5. Using Guice and the Jackson module.

# Agenda

- Introduction.
- The handlers chain.
- The context.
- Ratpack modules.
- **Asynchronous & Non Blocking model.**
- Q&A.

# The classic approach

- A thread-per-request.
  - There is a relatively large thread pool.
  - Each request is assigned a thread from the pool.
  - If the request handling code performs blocking I/O, the thread sleeps until the result is received.
- Overhead: thread management, memory consumption.

# Ratpack's value proposition

- HTTP IO is event-driven (non blocking).
  - Powered by Netty.
- Handler chain is organised as a pipeline of asynchronous functions.
  - Small compute thread pool.
  - Blocking operations are executed in a separate pool.
  - Integration with existing libraries like RxJava.

# Async model

- **Ratpack guarantees a deterministic flow.**
  - Promises are executed in the order they are defined.
  - Once the promises are resolved, they are executed in compute threads.
- **Async functions can be composed.**
  - To avoid the callback hell.

# Demo

## 6. Asynchronous model.

# Dziękuję!

Álvaro Sánchez-Mariscal

Web Architect - odobo



Images courtesy of  
shutterstock

