



Real Time Ratpack

Dan Woods – @danveloper

Rob Fletcher – @rfletcherEW



Quick Overview - Ratpack

- Originally inspired by Ruby Sinatra framework
- Has evolved into much, much more than just a microframework

```
@Grab("io.ratpack:ratpack-groovy:0.9.5")
import static ratpack.groovy.Groovy.*

ratpack {
    handlers {
        get {
            render "Hello world!"
        }
    }
}
```



Quick Overview - Ratpack

- Core APIs Written in Java 7
- Provides first-class support for Groovy and Java 8
- Employs newer Groovy features, like static compilation and type safety





Quick Overview - Ratpack

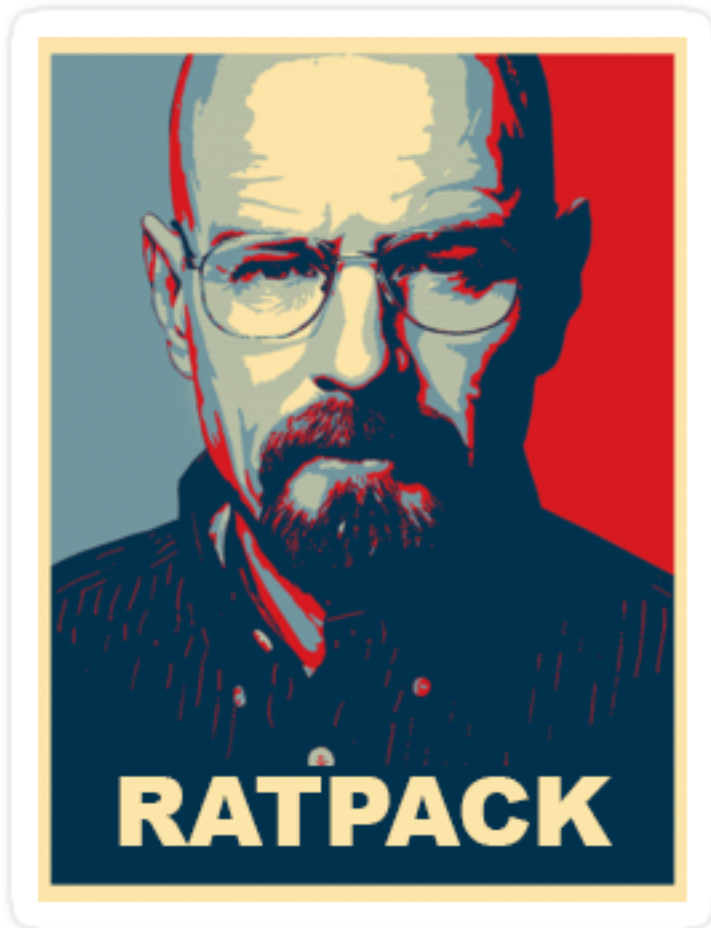
- High-speed, non-blocking web framework
- Application startup times in *milliseconds*
- Development time reloading with Springloaded



It's all about SPEED!



Quick Overview - Ratpack



Ratpack is your gateway
drug to non-blocking.



The Handler Chain

- Request handlers are able to be programmatically chained
- This defines the “edge” of your application
- If you think about traditional servlet programming, handlers embrace the concept of both a Servlet and a Filter
- The Groovy DSL allows handlers to be defined as Closures



Dependency Injection

- Ratpack abstracts dependency injection through its registry, allowing the DI framework to be pluggable
- Guice is provided and supported by default; Spring proof of concept is also available from Pivotal (Dave Syer)
- Groovy DSL allows dependencies to be injected as variable arguments to a handler closure
- Configuring dependencies is a one-liner & default support for Guice modules – enforces modularity without the pain



The Registry

- Registry provides runtime component lookups
- This is the abstraction on the dependency injection framework
- Allows for Remote Control during live functional or smoke tests
- The one place to go for component bindings in the application



Real Time Ratpack

- Websocket support out of the box
- Ratpack path bindings make WebSockets flexible

```
@Grab("io.ratpack:ratpack-groovy:0.9.5")
import static ratpack.groovy.Groovy.*
import static java.util.concurrent.Executors.*

ratpack {
  handlers {
    get("ws") {
      websocket(context) { ws ->
        newSingleThreadScheduledExecutor()
          .scheduleAtFixedRate(
            { ws.send(new Date().toString()) },
            0, 2, TimeUnit.SECONDS)
      } connect {}
    }
  }
}
```



Building an Event Broadcaster (SSE)

- Acts as an internal actuator for publishing application events to the client
- “Bound” within the registry, so any component can make use of it
- WebSocket handlers are registered with a broadcaster
- Broadcaster can conditionally route events to the right clients



Binding Clients to Context-Driven WebSocket Endpoints

- Use `ratpack-session` to manage the client-to-websocket endpoint relationship
- Provide an endpoint for retrieving session-stored websocket endpoint
- Build a handler that takes a path token as the client's unique websocket endpoint



Real Time Ratpack

Enterprise Integration

- The “Real Time Web” is about more than just a single application
- Need to be able to receive and broadcast “system-wide” events, and to the right consumers
- Ratpack’s un-opinionated approach to architecture makes it a one-liner to wire messaging to a real time web application



Ratpack Message Driven Architecture

Building a Messaging Module

- Apache Camel provides a fluent DSL for integrating with external systems (like JMS, AMQP, Twitter, etc.)
- Module exposes an internal message router that delegates to “message handlers”
- Handlers can be registered through the `binding` block of the Ratpack Groovy app DSL



Ratpack Message Driven Architecture

Broadcasting Enterprise Events

- App subscribes to JMS topic, receives message from enterprise system
- Message router determines the appropriate message handler
- Message handler uses the WebSocket event broadcaster to publish to the client



Summary

To Summarize...

- Ratpack is a high-speed web application framework
- Provides WebSocket support out of the box, making it a powerful utility in real time web applications
- As an un-opinionated framework, Ratpack makes it easy to build a message driven architecture
- Ratpack can be leveraged as a platform for real time enterprise eventing