From Monolith to Microservices with lagom

An Introduction

Steve Swing

@sswing

Who Am I?

- Steve Swing
- Independent consultant, working with Manifest Solutions since 2004.
- Java developer since 1999
- Not affiliated with Lightbend
- Lagom-curious since late-spring

Lagom Framework

- The Reactive Microservices Framework
- “The Opinionated Microservices Framework for moving away from the Monolith”
- Lagom — Swedish word for 'just right’
- Microservices Just Right

About What Is Lagom Opinionated?

- Requires Java 8
- Message Driven
- Asynchronous Request/Response
- Streaming
- Distributed Persistence, Event Sourcing/CQRS, Cassandra
- Domain-Driven Design
- Immutability commands/messages/entities/collections

Origins

- Open Source -- Licensed under the Apache License, Version 2.0
- https://github.com/lagom/lagom.git
- Earliest public commits March 2016
- Java API implemented in Scala (Scala API in progress)
- Release version 1.0.0 also released 1.1.0-RC1
- Maven support just added August 18

Technologies (1.1.0-RC1)

- Scala — 2.11.8
- Java — 8
- Play! Framework — 2.5.4
- Akka Streams, Clustering, & Persistence — 2.4.8
- Cassandra — 3.0.2
- Guice — 4.0
- Guava — 19.0
- Jackson — 2.7.2
- Immutables — 2.1.3
- Netty — 4.0.36
- ScalaTest — 2.2.4
- sbt — 0.13.12
- Maven — 3.3.9

Features

- Optimized for rapid development
  - Embedded container
  - Hot code reloading
  - Automatic Service Discovery
  - Activator Templates
  - Maven Archetypes
  - sbt & Maven runAll commands
- No application server container means no fighting with classpath and classloaders

“Any sufficiently advanced technology
is indistinguishable from magic.”

– Arthur C. Clarke

“All magic comes with a price, dearie!”

— Rumplestiltskin

Lagom Project Structure

```
├── front-end
├── <domain>-api
├── <domain>-impl
├── item-api
├── item-impl
├── order-api
├── order-impl
├── integration-tests
└── project
```

Lagom API: Writing Services

ServiceCall Interface

```
interface ServiceCall<Request, Response> {
  CompletionStage<Response> invoke(Request request);
}
```

Lagom API: Writing Services

Service Descriptors

```
import com.lightbend.lagom.javadsl.api.*;

import static com.lightbend.lagom.javadsl.api.Service.*;

public interface HelloService extends Service {
    ServiceCall<String, String> sayHello();

    default Descriptor descriptor() {
        return named(“hello")
            .withCalls(call(this::sayHello));
    }
}
```

Lagom API: Writing Services

Implementing Services

```java
import com.lightbend.lagom.javadsl.api.*;
import akka.NotUsed;
import static java.util.concurrent.CompletableFuture.completedFuture;

public class HelloServiceImpl implements HelloService {

    public ServiceCall<String, String> sayHello() {
        return name -> completedFuture("Hello " + name);
    }
}
```

Lagom API: Writing Services

Consuming Services: Binding a Service Client

```
import com.google.inject.AbstractModule;
import com.lightbend.lagom.javadsl.server.ServiceGuiceSupport;
import docs.services.HelloService;

public class Module extends AbstractModule implements ServiceGuiceSupport {

    protected void configure() {
        bindClient(HelloService.class);
    }
}
```

Lagom API: Writing Services

Consuming Services: Using a Service Client

```java
public class MyServiceImpl implements MyService {
  private final HelloService helloService;

  @Inject
  public MyServiceImpl(HelloService helloService) {
    this.helloService = helloService;
  }

  @Override
  public ServiceCall<NotUsed, String> sayHelloLagom() {
    return msg -> {
      CompletionStage<String> response = helloService.sayHello().invoke("Lagom");
      return response.thenApply(answer -> "Hello service said: " + answer);
    };
  }
}
```

Lagom API: Writing Services

Testing Services

```
import static com.lightbend.lagom.javadsl.testkit.ServiceTest.*;
import static java.util.concurrent.TimeUnit.SECONDS;
import static org.junit.Assert.assertEquals;
import org.junit.Test;

public class HelloServiceTest {

  @Test
  public void shouldSayHello() throws Exception {
    withServer(defaultSetup(), server -> {
      HelloService service = server.client(HelloService.class);

      String msg = service.sayHello().invoke("Alice").toCompletableFuture().get(5, SECONDS);
      assertEquals("Hello Alice", msg);
    });
  }
}
```

Lagom API: Writing persistent & clustered services

Persistent Entity

```java
import com.lightbend.lagom.javadsl.persistence.PersistentEntity;

public class Post1 extends PersistentEntity<BlogCommand, BlogEvent, BlogState> {
  @Override
  public Behavior initialBehavior(Optional<BlogState> snapshotState) {
    BehaviorBuilder b = newBehaviorBuilder(snapshotState.orElse(BlogState.EMPTY));

    // TODO define command and event handlers

    return b.build();
  }
}
```

Lagom API: Writing persistent & clustered services

Persistent Read-Side

```
@Override
public ServiceCall<NotUsed, Source<PostSummary, ?>> getPostSummaries() {
  return request -> {
    Source<PostSummary, ?> summaries = cassandraSession.select(
        "SELECT id, title FROM postsummary;").map(row ->
          PostSummary.of(row.getString("id"), row.getString("title")));
    return CompletableFuture.completedFuture(summaries);
  };
}
```

Getting Started

Maven

```
$ mvn archetype:generate -DarchetypeGroupId=com.lightbend.lagom \
  -DarchetypeArtifactId=maven-archetype-lagom-java -DarchetypeVersion=1.1.0-RC1
```

Getting Started

Activator (sbt)

```
$ activator new my-first-system lagom-java

$ activator new twitter-clone lagom-java-chirper
```

Example Projects

Resources

- **Martin Fowler**
  - Microservices • GOTO 2014 • Jan 15, 2015
  - https://www.youtube.com/watch?v=wgdBVIX9ifA
  - http://martinfowler.com/bliki/CQRS.html

Resources

- **Greg Young**
    - A Decade of DDD, CQRS, Event Sourcing
    - https://www.youtube.com/watch?v=LDW0QWie21s
    - “CQRS/ES is not a top-level architecture”
    - “Event Sourcing fits really well with the functional programming model. It does not fit well with object-oriented [imperative] model.”

Resources

- **Eric Evans**
  - DDD & Microservices: At Last, Some Boundaries! • GOTO 2015
  - https://www.youtube.com/watch?v=yPvef9R3k-M

Resources

- **Yannick De Turck**
  - Lagom in Practice
  - https://youtu.be/JOGlZzY6ycI
  - https://github.com/yannickdeturck/lagom-shop

Resources

- **Jonas Bonér**
  - Co-founder and CTO of Lightbend, inventor of the Akka project, co-author of the Reactive Manifesto and a Java Champion.
  - Free eBook: “Reactive Microservices Architecture — Design Principles for Distributed Systems.” 2016 O’Reilly Media, Inc.
  - https://www.lightbend.com/blog/reactive-microservices-architecture-free-oreilly-report-by-lightbend-cto-jonas-boner

Resources

- **Markus Eisele**
  - Lightbend Java Developer Advocate, Java Champion, & Former Java EE Spec lead.
  - Free eBook: “Developing Reactive Microservices: Enterprise Implementation in Java”, 2016 O’Reilly Media, Inc.
  - https://www.lightbend.com/blog/developing-reactive-microservices-free-oreilly-mini-book-by-java-champion-markus-eisele
  - https://github.com/lagom/activator-lagom-cargotracker

Resources

- **James Roper**
  - Lightbend Play! Framework Lead Developer
  - https://www.lagomframework.com/
  - http://www.lightbend.com/lagom