

Documentação – Configuração e utilização do Lagom

Para o desenvolvimento dos microserviços utilizados no projeto, se utilizará o *Lagom Framework*, que contém ferramentas e APIs que auxiliarão na criação do sistema.

Instalação e configuração do Maven

É necessário instalar e configurar o *Maven*, um software de automação de compilação, para fazer uso do *Lagom*. As instruções deste arquivo são focadas em sistemas *Linux*, mas os conceitos são facilmente transferíveis para outros sistemas operacionais.

- Baixe a versão mais recente do *Maven* (atualmente é a **3.5.3**)
- Descompacte o arquivo e mova o diretório para o local de sua preferência
- Configure as variáveis de ambiente para que o *Maven* seja reconhecido pelo sistema e integrar com a *JDK 1.8*.

Exemplo: no arquivo criado `/etc/profile.d/maven.sh` -

```
export JAVA_HOME=/opt/proprietario/jdk1.8.0_161/  
export M2_HOME=/opt/maven  
export PATH=${M2_HOME}/bin:${PATH}
```

No terminal

```
source /etc/profile.d/maven.sh
```

- Execute o seguinte comando em seu terminal para checar a instalação do *Maven*: `mvn -version`

O resultado deve ser parecido com o seguinte:

```
Maven home: /opt/maven  
Java version: 1.8.0_161, vendor: Oracle Corporation  
Java home: /opt/proprietario/jdk1.8.0_161/jre  
Default locale: en_US, platform encoding: UTF-8  
OS name: "linux", version: "4.9.0-6-amd64", arch:  
"amd64", family: "unix"
```

Criando um projeto *Lagom*

- No local em que deseja criar o seu projeto, utilize o comando do *Maven* `archetype:generate` da seguinte maneira:

```
mvn archetype:generate -Dfilter=com.lightbend.lagom:maven-archetype-lagom-java
```

- Digite **1**
- Digite o número correspondente à versão estável mais recente do *Lagom* (No momento, a versão **1.4.4**) [por problemas do meu computador, pude apenas utilizar a versão 1.3.0]
- Insira um **groupId** – exemplo: com.example.hello
- No campo **artifactId**, insira o nome do projeto - exemplo: hello-world
- No campo **version**, insira a versão do projeto.
- A opção default de *package* (padrão dos pacotes no projeto) é o mesmo valor do *groupId*
- Digite **Y**
- Caso dê certo, aparecerá uma mensagem de *BUILD SUCCESS*

Dentro do diretório criado, digite `mvn lagom:runAll` para inicializar todos os serviços e componentes relacionados ao projeto.

[Caso o tempo de boot do servidor *Cassandra* não for o suficiente para deixá-lo funcionando (pois o tempo máximo de espera padrão é 20 segundos), altere o arquivo `pom.xml` para incluir o seguinte trecho de configuração:

```
<plugin>
  <groupId>com.lightbend.lagom</groupId>
  <artifactId>lagom-maven-plugin</artifactId>
  <version>${lagom.version}</version>
  <configuration>
    <cassandraMaxBootWaitingSeconds>60</cassandraMaxBootWaitingSeconds>
  </configuration>
</plugin>
```

No caso acima, o tempo foi configurado para 60 segundos.]

Para testar se os serviços estão funcionando, você deve acessar

<http://localhost:9000/api/hello/World>. O retorno para essa entrada deve ser “Hello, World!”.

Os subprojetos no *Lagom* são divididos entre **api** e **impl**.

api – contém as interfaces dos serviços

impl – contém as implementações dos serviços.

Instalação do m2eclipse

Algumas versões do *Eclipse* tem o plugin do Maven por padrão. Caso não seja o caso, execute os seguintes passos:

- Clique em **Help** → **Install New Software**

- Em **Work with**, digite o repositório de sua versão do eclipse

Exemplo: <http://download.eclipse.org/releases/neon>

- Pesquise por *m2e*
- Instale o pacote.

Caso seja necessário, clique em **Window** → **Preferences** → **Java** → **Installed JREs** → **Edit**. Em **Default VM arguments**, insira `-Dmaven.multiModuleProjectDirectory=$M2_HOME`

Criando um projeto *Lagom* no *Eclipse*

Clique em **File**, vá em **New** → **Project**.

Selecione a opção **Maven project** e clique em **Next**

Na seleção do *Archetype*, caso não exista a entrada de *Artifact Id* `maven-archetype-lagom-java` com a versão mais recente do *Lagom*, clique em **Add Archetype**

- Insira em **Archetype Group Id**: `com.lightbend.lagom`
- Em **Archetype Artifact Id**: `maven-archetype-lagom-java`
- **Version**: `1.4.4`
- Deixe **Repository URL** em branco.

Importando um projeto *Maven* no *Eclipse*

Em **File**, clique em **Import**.

Escolha **Existing Maven Projects**.

Clique em **Next**.

Em **Root Directory**, coloque o caminho para o projeto.

Selecione todos os subprojetos e clique em **Finish**.

Executando um projeto Lagom no Eclipse

Clique com o botão direito em cima do projeto Lagom principal.

Vá em *Run as* → *Maven build...*

Em *Goals*, digite `lagom:runAll`.

Garanta que sua *JRE* está corretamente configurada e clique em *Run*.

Instalação do Java Client for Google Maps Services

No arquivo `pom.xml` no diretório principal do seu projeto, insira a seguinte dependência à lista:

```
<dependency>
    <groupId>com.google.maps</groupId>
    <artifactId>google-maps-services</artifactId>
    <version>(versão mais recente - no momento, 0.2.7)</version>
</dependency>
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-nop</artifactId>
    <version>1.7.25</version>
</dependency>
```

APIs do Lagom

O Lagom fornece APIs que facilitam o desenvolvimento de microserviços, dentre elas:

Services API – auxilia a declarar e implementar as interfaces dos serviços.

Persistence API – fornece entidades persistentes para serviços que armazenam dados.

Servidores do Lagom

O Lagom inicia por padrão, no comando `runAll`, dois servidores:

Kafka – responsável por intermediar informações compartilhadas entre dois serviços. Pode ser iniciado como serviço independente, sem necessitar da execução de todos os serviços do projeto, pelo comando `lagom:startKafka`, e encerrado pelo comando `lagom:stopKafka`.

Cassandra – banco de dados utilizado para persistir dados. Pode ser iniciado e finalizado independentemente dos outros serviços seguindo o mesmo padrão do servidor *Kafka*. Apesar de ser a opção padrão do *Lagom*, há compatibilidade com outros bancos de dados, como o *PostgreSQL*. *Cassandra*, ao contrário do *PostgreSQL*, não é um banco de dados relacional.

Utilizando *PostgreSQL* para armazenamento de Entidades Persistentes

Em aplicações futuras, utilizaremos o *PostgreSQL* para armazenar os dados.

Para utilizá-lo para a persistência dos dados, deve-se inserir no arquivo *pom.xml* do diretório raiz o seguinte trecho:

```
<dependency>

    <groupId>com.lightbend.lagom</groupId>
    <artifactId>lagom-javads1-persistence-jdbc_${scala.binary.version}</artifactId>
    <version>${lagom.version}</version>
</dependency>
```

Para configurar o servidor, adicione ao arquivo *application.conf* a seguinte configuração:

```
db.default {

    driver = "org.postgresql.Driver"

    url = "jdbc:postgresql://database.exemplo.com/lagom-db"
}

jdbc-defaults.slick.profile = "slick.jdbc.PostgresProfile$"
```