apache-camel-spring-boot-1.x (0.0.1)

Maksim Kostromin

Version 0.0.1, 2018-06-14 21:49:15 UTC

Table of Contents

Travis CI status: [Build Status]

Chapter 1. Introduction

[Reference documentation](#)

gradle

```
./gradlew
java -jar build/libs/*.jar
bash build/libs/*.jar
```

maven

```
./mvnw
java -jar target/*.jar
bash target/*.jar
```

generated by [generator-jvm](#) yeoman generator (java-spring-boot_1.x)

Chapter 2. Implementation

2.1. camel spring configuration

src/main/resources/application.properties

```
camel.springboot.main-run-controller=true
```

2.2. move all files from directory

source directory: /tmp/camel-in/ destination directory: /tmp/camel-out/

src/main/java/daggerok/app/MoveFilesCamelConfig.java

```java
@Configuration
@RequiredArgsConstructor
public class MoveFilesCamelConfig {

  @Bean
  public RouteBuilder moveFilesRoute() {
    return new RouteBuilder() {
      @Override
      public void configure() throws Exception {
        from("file:///tmp/camel-in")
            .routeId("dir-to-dir")
            .to("file:///tmp/camel-out");
      }
    };
  }
}
```

2.3. move files via jms

source directory: /tmp/camel-in-jms/ process queue: files destination directory: /tmp/camel-jms-out/

move files via jms queue `jms:queue:files`

```java
@Slf4j
@Configuration
@RequiredArgsConstructor
public class MoveFilesViaJmsCamelConfig {

  @Bean
  public RouteBuilder moveFilesViaJmsRoute() {
    return new RouteBuilder() {
      @Override
      public void configure() throws Exception {

        from("file:///tmp/camel-jms-in")
            .routeId("in-dir-to-jms")
            .transform()
            .body(GenericFile.class, genericFile -> {
              final File file = File.class.cast(genericFile.getFile());
              try (final FileInputStream inputStream = new FileInputStream(file)) {
                final InputStreamReader streamReader = new InputStreamReader
(inputStream);
                final BufferedReader in = new BufferedReader(streamReader);
                return in.lines().collect(joining());
              } catch (IOException e) {
                log.error(e.getLocalizedMessage());
                throw new RuntimeException(e);
              }
            })
            .to("jms:queue:files")
        ;

        from("jms:queue:files")
            .routeId("jms-to-out-dir")
            .to("file:///tmp/camel-jms-out")
        ;
      }
    };
  }
}
```

jms component customizer

```java
@Component
@RequiredArgsConstructor
class ApplicationJmsComponentCustomizer implements ComponentCustomizer<JmsComponent> {

  final ConnectionFactory connectionFactory;

  @Override
  public void customize(JmsComponent component) {
    component.setConnectionFactory(connectionFactory);
  }
}
```

```java
@Component
@RequiredArgsConstructor
class ApplicationJmsComponentCustomizer implements ComponentCustomizer<JmsComponent> {
```

Chapter 3. BOM

maven

```xml
<dependencyManagement>
  <dependencies>
    <!-- now we can easily use compatible apache camel components -->
    <dependency>
      <groupId>org.apache.camel</groupId>
      <artifactId>camel-spring-boot-dependencies</artifactId>
      <version>${apache.camel.version}</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
```

gradle

```gradle
dependencyManagement {
  imports {
    mavenBom "org.apache.camel:camel-spring-boot-dependencies:$apacheCamelVersion"
```

Chapter 4. Links

GitHub repo GitHub pages

GitHub repo GitHub pages