**Expose Rest web services with Apache Camel**

1. create a maven web project with following command:

mvn archetype:generate -DgroupId=com.rest -DartifactId=CamelRest -Dversion=1.0-SNAPSHOT -DarchetypeArtifactId=maven-archetype-webapp -DinteractiveMode=false

2. pom.xml

|  |
| --- |
| <project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd"*>  <modelVersion>4.0.0</modelVersion>  <groupId>camel.rest</groupId>  <artifactId>CamelRest</artifactId>  <packaging>war</packaging>  <version>1.0-SNAPSHOT</version>  <name>CamelRest Maven Webapp</name>  <url>http://maven.apache.org</url>    <properties>  <springframework.version>4.2.5.RELEASE</springframework.version>  <jackson.version>2.5.3</jackson.version>  </properties>    <dependencies>  <dependency>  <groupId>junit</groupId>  <artifactId>junit</artifactId>  <version>3.8.1</version>  <scope>test</scope>  </dependency>  <dependency>  <groupId>org.apache.camel</groupId>  <artifactId>camel-servlet</artifactId>  <version>2.16.1</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-web</artifactId>  <version>4.2.5.RELEASE</version>  </dependency>  <dependency>  <groupId>org.apache.camel</groupId>  <artifactId>camel-spring</artifactId>  <version>2.16.1</version>  </dependency>  <dependency>  <groupId>org.apache.camel</groupId>  <artifactId>camel-spring-javaconfig</artifactId>  <version>2.16.1</version>  </dependency>  <dependency>  <groupId>org.apache.camel</groupId>  <artifactId>camel-http4</artifactId>  <version>2.16.1</version>  </dependency>  <dependency>  <groupId>org.apache.camel</groupId>  <artifactId>camel-http</artifactId>  <version>2.16.1</version>  </dependency>  <dependency>  <groupId>javax</groupId>  <artifactId>javaee-api</artifactId>  <version>7.0</version>  </dependency>  <dependency>  <groupId>org.apache.camel</groupId>  <artifactId>camel-jackson</artifactId>  <version>2.15.2</version>  </dependency>  <dependency>  <groupId>org.json</groupId>  <artifactId>json</artifactId>  <version>20090211</version>  </dependency>  <dependency>  <groupId>com.fasterxml.jackson.core</groupId>  <artifactId>jackson-core</artifactId>  <version>2.6.5</version>  </dependency>  <dependency>  <groupId>com.fasterxml.jackson.core</groupId>  <artifactId>jackson-annotations</artifactId>  <version>2.6.5</version>  </dependency>  <dependency>  <groupId>org.codehaus.jackson</groupId>  <artifactId>jackson-core-asl</artifactId>  <version>1.9.13</version>  </dependency>  <dependency>  <groupId>com.google.code.gson</groupId>  <artifactId>gson</artifactId>  <version>2.4</version>  </dependency>  </dependencies>  <build>  <finalName>CamelRest</finalName>  <pluginManagement>  <plugins>  <plugin>  <groupId>org.apache.maven.plugins</groupId>  <artifactId>maven-compiler-plugin</artifactId>  <version>3.5.1</version>  <configuration>  <source>1.8</source>  <target>1.8</target>  <compilerArgument>-Xlint:all</compilerArgument>  <compilerArgument>-parameters</compilerArgument>  <showDeprecation>true</showDeprecation>  <showWarnings>true</showWarnings>  </configuration>  </plugin>  <plugin>  <groupId>org.apache.maven.plugins</groupId>  <artifactId>maven-war-plugin</artifactId>  <configuration>  <failOnMissingWebXml>false</failOnMissingWebXml>  </configuration>  </plugin>  </plugins>  </pluginManagement>  </build>  </project> |

3. AppConfiguration.java

|  |
| --- |
| package com.camel.rest;  import java.util.ArrayList;  import java.util.List;  import org.apache.camel.CamelContext;  import org.apache.camel.builder.RouteBuilder;  import org.apache.camel.spring.javaconfig.CamelConfiguration;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.beans.factory.annotation.Qualifier;  import org.springframework.context.annotation.ComponentScan;  import org.springframework.context.annotation.Configuration;  @Configuration  @ComponentScan("com.camel.rest")  public class AppConfiguration extends CamelConfiguration {  @Autowired  @Qualifier("routerImpl")  private CamelRestRouteImpl camelRestRouterImpl;  @Override  protected void setupCamelContext(CamelContext camelContext) throws Exception {  super.setupCamelContext(camelContext);  }  @Override  public List<RouteBuilder> routes() {  List<RouteBuilder> list = new ArrayList<>();  list.add(camelRestRouterImpl);  return list;  }  } |

4. CamelRestRouteImp.java

|  |
| --- |
| package com.camel.rest;  import org.apache.camel.ExchangePattern;  import org.apache.camel.builder.RouteBuilder;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.beans.factory.annotation.Qualifier;  import org.springframework.stereotype.Component;  @Component("routerImpl")  public class CamelRestRouteImpl extends RouteBuilder {  @Autowired  @Qualifier("myProcessor")  private MyProcessor myProcessor;  @Autowired  @Qualifier("exceptionProcessor")  private CustomExceptionProcessor customExceptionProcessor;  @Autowired  @Qualifier("requestProcessor")  private RequestProcessor requestProcessor;  @Autowired  @Qualifier("callRestProcessor")  private CallRestProcessor callRestProcessor;  @Override  public void configure() throws Exception {  restConfiguration().component("servlet");  rest("/hello").description("Test Camel Rest").get().produces("application/json").to("direct:hello");  rest("/camelPostRest").description("Camel REST WS with POST request").post().produces("application/json")  .consumes("application/json").route().setExchangePattern(ExchangePattern.InOut)  .convertBodyTo(String.class).process(requestProcessor);  rest("/cameRestHttp").get().produces("application/json").route().setExchangePattern(ExchangePattern.InOut)  .process(callRestProcessor);  onException(Exception.class).handled(true).process(customExceptionProcessor);  from("direct:hello").transform().constant("Hello Camel REST API");  }  } |

5. WebAppInitializer.java

|  |
| --- |
| package com.camel.rest;  import javax.servlet.ServletContext;  import javax.servlet.ServletException;  import javax.servlet.ServletRegistration;  import org.apache.camel.component.servlet.CamelHttpTransportServlet;  import org.springframework.web.WebApplicationInitializer;  import org.springframework.web.context.ContextLoaderListener;  import org.springframework.web.context.support.AnnotationConfigWebApplicationContext;  public class WebAppInitializer implements WebApplicationInitializer {  @Override  public void onStartup(ServletContext servletContext) throws ServletException {  AnnotationConfigWebApplicationContext ctx = new AnnotationConfigWebApplicationContext();  ctx.register(AppConfiguration.class);  servletContext.addListener(new ContextLoaderListener(ctx));  ctx.setServletContext(servletContext);  ServletRegistration.Dynamic dispatcher = servletContext.addServlet("CamelServlet",  CamelHttpTransportServlet.class);  dispatcher.addMapping("/api/v1/\*");  dispatcher.setLoadOnStartup(1);  }  } |

6. CallRestProcessor.java

|  |
| --- |
| package com.camel.rest;  import org.apache.camel.CamelContext;  import org.apache.camel.Exchange;  import org.apache.camel.Processor;  import org.apache.camel.ProducerTemplate;  import org.apache.camel.impl.DefaultExchange;  import org.springframework.stereotype.Component;  @Component("callRestProcessor")  public class CallRestProcessor implements Processor {  @Override  public void process(Exchange exchange) throws Exception {  CamelContext context = exchange.getContext();  ProducerTemplate template = context.createProducerTemplate();  Exchange exchange2 = new DefaultExchange(context);  Exchange exchange3 = template.send("http://localhost:9091/SpringRest/employees", exchange2);  String response = exchange3.getOut().getBody(String.class);  System.out.println(response);  exchange.getOut().setBody(response);  }  } |

7. CustomExceptionProcessor.java

|  |
| --- |
| package com.camel.rest;  import org.apache.camel.Exchange;  import org.apache.camel.Processor;  import org.springframework.stereotype.Component;  @Component("exceptionProcessor")  public class CustomExceptionProcessor implements Processor {  @Override  public void process(Exchange exchange) throws Exception {  String body = exchange.getIn().getBody(String.class);  System.out.println("body: " + body);  System.out.println("CustomExceptionProcessor.process(Exchange):12");  }  } |

8. InputRequest.java

|  |
| --- |
| package com.camel.rest;  import java.io.Serializable;  public class InputRequest implements Serializable {  private static final long serialVersionUID = 7016767119769364735L;  private String firstName;  private String middleName;  private String lastName;  public String getFirstName() {  return firstName;  }  public void setFirstName(String firstName) {  this.firstName = firstName;  }  public String getMiddleName() {  return middleName;  }  public void setMiddleName(String middleName) {  this.middleName = middleName;  }  public String getLastName() {  return lastName;  }  public void setLastName(String lastName) {  this.lastName = lastName;  }  } |

9. MyBean.java

|  |
| --- |
| package com.camel.rest;  import org.apache.camel.Exchange;  public class MyBean {  public Exchange getBody(Exchange exchange) {  System.out.println("MyBean.getBody():8");  String body = (String) exchange.getProperty("requestInput");  System.out.println("MyBean.getBody(Exchange):10: " + body);  exchange.getOut().setBody(body);  return exchange;  }  } |

10. MyProcessor.java

|  |
| --- |
| package com.camel.rest;  import org.apache.camel.Exchange;  import org.apache.camel.Processor;  import org.springframework.stereotype.Component;  @Component("myProcessor")  public class MyProcessor implements Processor {  @Override  public void process(Exchange exchange) throws Exception {  System.out.println("MyProcessor.process(Exchange):10");  }  } |

11. RequestProcessor.java

|  |
| --- |
| package com.camel.rest;  import org.apache.camel.Exchange;  import org.apache.camel.Processor;  import org.springframework.stereotype.Component;  import com.fasterxml.jackson.databind.ObjectMapper;  @Component("requestProcessor")  public class RequestProcessor implements Processor {  @Override  public void process(Exchange exchange) throws Exception {  ObjectMapper mapper = new ObjectMapper();  InputRequest request = mapper.readValue((String) exchange.getIn().getBody(), InputRequest.class);  request.setFirstName("fn");  request.setLastName("ln");  request.setMiddleName("mn");  String json = mapper.writeValueAsString(request);  exchange.getOut().setBody(json);  }  } |