
PA165 Integration Lab - Apache Camel

Filip Nguyen

Lab Software Architectures and Information Systems

Seminar 14

December 17, 2013

Today

- Review the Lecture
 - Play with Apache Camel
-

Source code

- <https://github.com/nguyenfilip/pa165camel.git>
 - Contains solved/done seminar lab
 - camel-standalone
 - camel-webapp
-

Standardization

- We want to add `<items>` element around all the items

```
<order>
```

```
<items>
```

```
<item><product>Skis</product><quantity>20</quantity></item>
```

```
<item><product>Skis2</product><quantity>20</quantity></item>
```

```
</items>
```

```
</order>
```

Enrichment

- We want to add `<created>` element

```
<order>
```

```
<created>17-12-2013</created>
```

```
<items>
```

```
<item><product>Skis</product><quantity>20</quantity></item>
```

```
<item><product>Skis2</product><quantity>20</quantity></item>
```

```
</items>
```

```
</order>
```

Problem

- We have a folder with orders in old format as well as folder with orders in new format (but without “created”)

New format:

```
<order>
```

```
<items>
```

```
<item><product>Skis</product><quantity>20</quantity></item>
```

```
<item><product>Skis2</product><quantity>20</quantity></item>
```

```
</items>
```

```
</order>
```

Another Problem

- Also, we want to allow our customers to submit ordres using
 - Web Services
 - CSV files
-

Orders formats

- Inspect camel-standalone/orders directory
-

Camel in Standalone Java

- Inspect `pa165-camel-lab/pom.xml`
 - `mvn exec:java -Dexec.mainClass=cz.pa165.mains.StandaloneMain`
 - `StandaloneMain.java`
 - `CamelContext`
 - Watch out for following folders in camel-standalone
 - `new-input-orders`
 - `old-input-orders`
 - `outputDir`
 - `errored`
-

File Component

- Various parameters:
 - "?recursive=true&flatten=true"
- CamelFileName header

Validator

- Validates against XSD
 - orderSchema.xsd
 - Searches the XSD on classpath
-

JAXB

- JaxbDataFormat
 - .unmarshal(jaxb)
 - .marshal(jaxb)
-

Camel in Spring

- applicationContext-standalone.xml
 - SpringMain
 - We need to use Thread.sleep again!
 - @EndpointInject in Spring Bean to inject a Camel Endpoint
-

Camel in Web App

- inspect camel-web/pom.xml
- spring-ws-context.xml

```
<import resource="classpath:applicationContext-standalone.xml" />
```

Camel + Spring WS

- web.xml
 - spring-ws-context.xml
DefaultWsd11Definition
 - Contract for web service in:
src/main/webapp/WEB-INF/order.xsd
 - JAXB classes are used to map
request/response
-

What we need to do

- Convert request to our format (old or new xml format)
 - Route the converted message to our existing commonRoute
 - We need to return a sensible response to the Web Service Client
-

Making a Request

- tomcat7:run in camel-web
- modify request.xml

```
curl -H "Content-Type: text/xml" -X POST -d @request.xml http://localhost:8080/camel/order
```

Create a new CSV route

- This should be done in standalone Camel
- Assume we are getting orders in CSV format from yet another folder
- Use CSV camel component and convert the csv into XML format :-)

<product>, <quantity>
