Menschen. Innovationen. Lösungen.

Integration von Java Legacy Code in die Fusion Middleware 11 mittels des SOA Suite Spring Components

OPITZ CONSULTING

OPITZ CONSULTING

Menschen. Innovationen. Lösungen.

Java Legacy Code in der Fusion Middleware 11g

Einbindung mittels des Spring Components

4

Alexander Rüsberg, Berater

OPITZ CONSULTING Essen GmbH

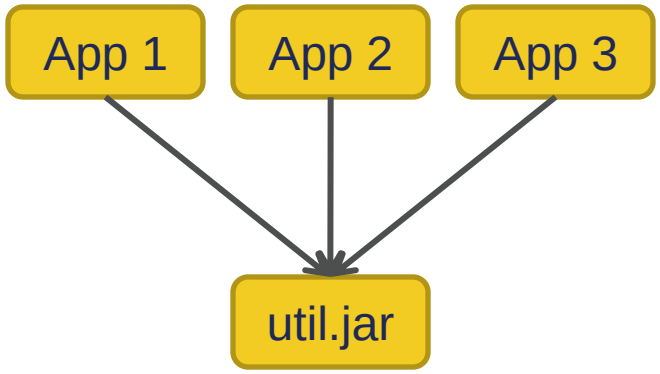
Essen, 11.03.2010

Agenda

1. **Java Code in der Enterprise Welt**
2. **Java in der Fusion Middleware 11g**
3. **Das Spring Component als Brücke zwischen Java Legacy Code und der Fusion Middleware 11g**
4. **Vorgehen anhand eines Beispiels**

Java Code in der Enterprise Welt

- **Unternehmensweite Java Bibliotheken, z.B. Validatoren**
- **Lose Kopplung mittels Spring**
- **Verwaltung der Bibliothek mittels Maven**
- **Automatische Tests**
- **Continous Integration**




```
3  public class AuftragsnummernGeneratorImpl implements AuftragsnummernGenerator {
4
5      public String generateAuftragsnummer(String vertriebskanalName) {
6          if (vertriebskanalName == null || vertriebskanalName.length() == 0) {
7              throw new IllegalArgumentException(
8                      "Der Name für den Vertriebskanal ist 'null' oder leer!");
9          }
10         if (!Vertriebskanal.isVertriebskanal(vertriebskanalName)) {
11             throw new IllegalArgumentException("Der Name \""
12                     + vertriebskanalName
13                     + "\" für den  Vertriebskanal existiert nicht!");
14         }
15         String alias = Vertriebskanal.getAliasForName(vertriebskanalName);
16         return alias + "-" + System.currentTimeMillis();
17     }
18
```

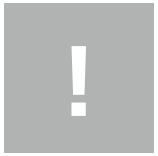
Java in der Fusion Middleware 11g

- **Mediator**
  - Callout
  - Interface muss implementiert werden
  - Allgemeines Datenformat
- **BPEL**
  - Embedding
  - Nur Code Schnipsel

```
<bpelx:exec name="Java_Embedding_1" version="1.5" language="java">
    <![CDATA[System.out.println("Hello, World");]]>
</bpelx:exec>
```

```
55
56      public boolean postRouting(CalloutMediatorMessage calloutMediatorMessage,
57                                 CalloutMediatorMessage calloutMediatorMessage1,
58                                 Throwable throwable) {
59
60          String sPayload = "null";
61          for (Iterator msgIt =
62               calloutMediatorMessage1.getPayload().entrySet().iterator();
63               msgIt.hasNext(); ) {
64              Map.Entry msgEntry = (Map.Entry)msgIt.next();
65              Object msgKey = msgEntry.getKey();
66              if (msgKey.equals(MYKEY)) {
67                  Object msgValue = msgEntry.getValue();
68                  sPayload = XmlUtils.convertDomNodeToString((Node)msgValue);
69                  try {
70                      XMLDocument changedoc;
71                      changedoc = XmlUtils.getXmlDocument(sPayload);
72                      Node node =
73                          changedoc.selectSingleNode("//ns2:orderResponse",
74                                                     new LocalNamespaceResolver());
75                      org.w3c.dom.Element auftragsnummerElement =
76                          changedoc.createElementNS(NAMESPACE,
77                                                    "ns2:auftragsnummer");
78                      auftragsnummerElement.appendChild(
79                          changedoc.createTextNode(
80                              generateAuftragsnummer(
81                                      getVertriebskanal(calloutMediatorMessage))));
82                      node.appendChild(auftragsnummerElement);
83                      calloutMediatorMessage1.addPayload(MYKEY,
84                                                         changedoc.getFirstChild());
85                  } catch (Exception f) {
86                      System.out.println(f);
87                  }
88              }
89          }
90          return true;
91      }
92
93      private String getVertriebskanal(CalloutMediatorMessage calloutMediatorMessage) {
94          for (Iterator msgIt =
95               calloutMediatorMessage.getPayload().entrySet().iterator();
96               msgIt.hasNext(); ) {
97              Map.Entry msgEntry = (Map.Entry)msgIt.next();
98              Object msgKey = msgEntry.getKey();
99              if (msgKey.equals(MYKEY)) {
100                 Object msgValue = msgEntry.getValue();
101                 String sPayload =
102                     XmlUtils.convertDomNodeToString((Node)msgValue);
103                 try {
```

Das Spring Component als Brücke zwischen Java und der Fusion Middleware 11g



Überblick

- Integration von Spring Komponenten in SOA Composites
- Bereitstellen von Java Klassen als Services
- Feature Preview im Patchset 1
- Standardmäßig deaktiviert

Vorteile

- **Einfache Integration bestehender Funktionalität**
- **Dependency Injection / Lose Kopplung**
- **Automatische Erstellung von WSDLs auf Basis der Java Klasse**
- **Erlaubt den Import bestehender Java Archive mit Spring-Definitionen (ApplicationContext)**

Einsatzmöglichkeiten

- **Wiederverwendung bestehender Funktionalität**
  - Konvertierung
  - Validierung
  - Mapping
- **Logging**
- **Aufruf von REST Services**

Vorgehen

- **Import eines JARs mit ApplicationContext**
  - Definition der JARs als Library
  - Anpassen des Deployment Profile damit das JAR mit deployed wird
- **Erstellen eines Spring Context Components**
  - Import des im JAR enthaltenen Contexts
  - Bereitstellen von Spring Beans als Service
- **Integration der neuen Services in den Workflow**

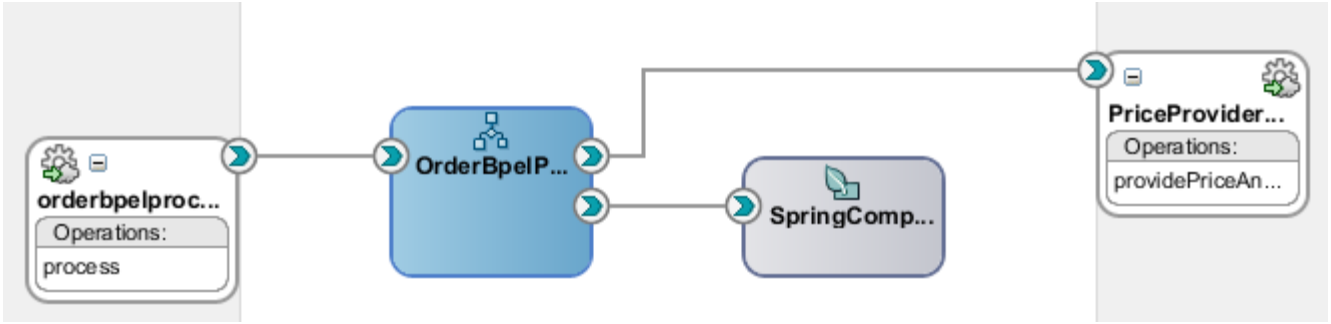
Spring Context

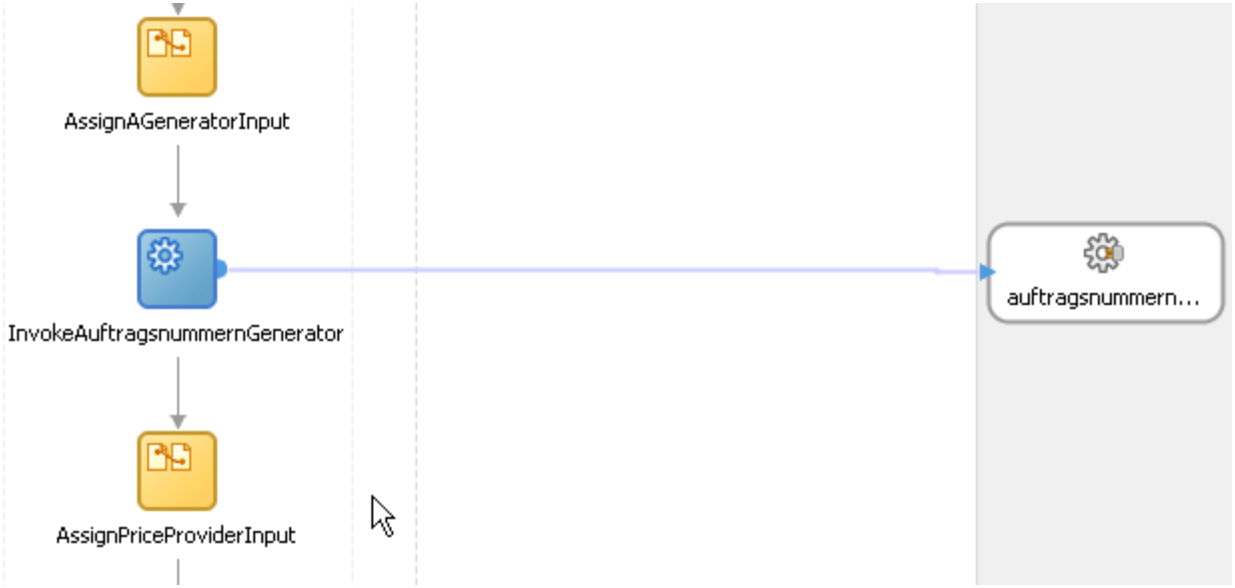
```
<?xml version = '1.0' encoding = 'UTF-8'?>
<beans xmlns=http://www.springframework.org/schema/beans  …. >

   <import resource="classpath:/application-context.xml"/>

  <sca:service name="auftragsnummernGeneratorService"
         target="auftragsnummernGenerator"
         type="util.AuftragsnummernGenerator"/>
</beans>
```

Composite & BPEL Process





Fragen und Antworten

Kontakt

Alexander Rüsberg
Berater

OPITZ CONSULTING Essen GmbH
Altendorfer Straße 3 ■ 45127 Essen
Tel. +49 (201) 892994 - 1721
alexander.ruesberg@opitz-consulting.com

Nuhad Shaabani
Berater

OPITZ CONSULTING Essen GmbH
Altendorfer Straße 3 ■ 45127 Essen
Tel. +49 (201) 892994 - 1720
nuhad.shaabani@opitz-consulting.com