

Checkliste

Was macht einen „guten“ Service aus?

1. Fachliche Aspekte

Ein fachlich guter Service setzt die Anforderungen der Fachseite adäquat um und ist so beschrieben, dass er in diversen Situationen aufrufbar ist. Wir sprechen in diesem Fall von sogenannten „Public Services“, meinen damit diejenigen Dienste, die fachlichen Nutzen bringen und deren Schnittstellen auf allgemeine Verwendung hin entworfen wurden. „Private Services“, also nicht zu exponierende technische Konstrukte, werden niemals wiederverwendet und müssen daher nicht gesondert betrachtet werden.

1.1 Servicedesign:

- Funktionalität entspricht Anforderungen
- Starker fachlicher Name des Services
- Auffindbarkeit und Verwendung durch Namenskonventionen und Beschreibung
- Verstehbarkeit durch gute Kohärenz
- Verwendung Kanonischer Datenformate

1.2 Service Klassifizierung:

- Einordnung in Service Kategorisierung

1.3 Fehlerbehandlung:

- Beschreibung der möglichen fachlichen Exceptions
- Bereitstellung einer fachlichen Testsuite

2. Technische Aspekte:

Die Aspekte, die uns zu technisch „guten“ Services führen, lassen sich weniger aus den funktionalen Anforderungen ableiten, sondern ergeben sich eher aus der Sichtbarkeit der Services und dem Gebot der Losen Kopplung. Dabei sind für jeden Service eine Reihe von Anforderungen zu prüfen:

2.1 Technische Compliance:

- Grad der Zustandslosigkeit
- Partizipation an Kompensationen
- Existenz von Versionierungskonzepten

- Eigenschaft der Idempotenz
- Struktureller Aufbau der WSDL-Datei
- Einhaltung von WS-* Standards
- Einhaltung von Projektstandards

2.2 Interoperabilität:

- WS-I Konformität
- Integrierbarkeit in verschiedene Security-Konzepte
- Keine Exposition interner Generiken

2.3 Kommunikation:

- Entkopplung vom Kommunikationsprotokoll
- Falls sinnvoll: Asynchrone Kommunikationsmuster
- QoS
- Policy

2.4 Fehlerbehandlung:

- Beschreibung der möglichen technischen Exceptions
- Bereitstellung einer technischen Testsuite

Kontakt:

Torsten Winterberg, OPITZ CONSULTING GmbH
Director Strategy and Innovation - Head of Competence Center SOA - Oracle ACE Director
Kirchstr. 6 - 51647 Gummersbach
Phone +49 2261 6001-0 - Fax -4200 - Mobile +49 173 5479302
torsten.winterberg@opitz-consulting.com - <http://www.opitz-consulting.com>

Diese Checkliste bietet einen Auszug aus der Artikelserie „SOA aus dem wahren Leben“, erschienen im Javamagazin (2009). Autoren: Berthold Maier, Bernd Trops, Hajo Normann, Clemens Utschig-Utschig, Torsten Winterberg.

Checkliste

Strukturvorschläge zum Aufbau einer WSDL-Datei

Der strukturelle Aufbau eines WSDL entscheidet über dessen Güte und Wiederverwendungsmöglichkeit. Es haben sich einige Best Practices herausgestellt, wie eine effiziente WSDL aufgebaut sein sollte:

1. Die XSD-Schemadefinition innerhalb des WSDL hat nur für Messages (ServiceOperationen) ohne Versionsnummer zu erfolgen. Basistypen, die möglichst einem Common Object Model folgen, sind durch Schema-Imports und hier mit der entsprechenden Versionsnummer im Namespace zu importieren. Beispiel:

```
<types>

<schema targetNamespace=http://www.myCompany.com/domäne/anwendung/MyService-v1"
xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:cust="http://www.myCompany.com/domäne/xsd/Customer-v1/2008/12">

<import namespace="http://www.myCompany.com/domäne/xsd/Customer-v1/2008/12"
schemaLocation="./Customer.xsd"/>

<element name="addAddressMessage">

  <complexType>

    <sequence>

      <element name="address" type="cust:Contact"/>

    </sequence>

  </complexType>

</element>

<element name="addAddressResponse">

  <complexType>

    <sequence>

      <element name="cid" type="string"/>

    </sequence>

  </complexType>

</element>

</schema>

</types>
```

2. Der WSDL-Namespace muss eindeutig lesbar sein und dabei neben der Firmendomäne die Anwendungsdomäne und die fachliche Zugehörigkeit ausdrücken. Weiter ist als Versionsnummer die Major-Versionsnummer mit in den TargetNamespace aufzunehmen.

```
<definitions name="MyService-V1.wsdl"
```

```
targetNamespace=http://www.myCompany.com/domäne/anwendung/MyService-V1
```

3. Versionsnummern mit Minor-Nummer sollten auch für die einzelnen Porttypes, Bindings und Services implementiert werden und für jede Version ein separater Service definiert werden:

```
<portType name=" MyService _v1_1 ">
```

```
<operation name=" doAnything">
```

```
<input message="tns: doAnythingRequestMessageType" name=" doAnythingRequest"/>
```

```
<output message="tns: doAnythingResponseMessageType" name=" doAnythingResponse"/>
```

```
</operation>
```

```
</portType>
```

```
<service name=" MyService _v1_1">
```

```
<port binding="tns: doAnything_v1_1_SOAP11Binding " name="DoAnything-SOAP11OverHTTP">
```

```
<soap:address location="http://host:8888/doAnything"/>
```

```
</port>
```

```
</service>
```

4. Annotationen mit semantischen Informationen sollten in der recht technischen WSDL-Markupsprache ebenfalls nicht fehlen. Hier werden Aussagen über Kategorisierungen, Fachlichkeit, usw. dokumentiert. Als Standard existiert dazu „Semantic Annotations for WSDL and XML Schema“.
5. Ohne MDA/MDSO halten sich diverse WS-Generierungstools selten an diese Vereinbarungen und sollten deshalb vermieden oder alternativ korrekt konfiguriert werden.

Kontakt: Torsten Winterberg, OPITZ CONSULTING GmbH

Director Strategy and Innovation - Head of Competence Center SOA - Oracle ACE Director

Kirchstr. 6 - 51647 Gummersbach

Phone +49 2261 6001-0 - Fax -4200 - Mobile +49 173 5479302

torsten.winterberg@opitz-consulting.com - <http://www.opitz-consulting.com>

Diese Checkliste bietet einen Auszug aus der Artikelserie „SOA aus dem wahren Leben“, erschienen im Javamagazin (2009). Autoren: Berthold Maier, Bernd Trops, Hajo Normann, Clemens Utschig-Utschig, Torsten Winterberg.