

İstemci ve Sunucu Tarafında Kullanılan Ortak Bileşenler



Spring WS'in Ortak Bileşenleri

- Spring WS'in **hem istemci, hem de sunucu tarafında** yer alan bileşenleridir
- Web servis çağrılarının oluşturulmasında, sunucuya gönderilmesinde, cevabın oluşturulmasında ve istemciye dönülmesinde **görev alırlar**

WebServiceMessage

- **Protokolden bağımsız XML mesajını** temsil eder
- Mesaj **payload'ına erişim** sağlayan metotlar sunar
- `javax.xml.transform.Source` ve `Result` arayüzleri üzerinden mesaj içeriği okunabilir veya yazılabilir

WebService MessageFactory

- WebServiceMessage oluşturmayı sağlar
- **SaaJSoapMessageFactory**
 - SOAP With Attachments API for Java (SaaJ) kullanır, DOM tabanlıdır
- **AxiomSoapMessageFactory**
 - Streaming API for XML (StAX) ve Axis2 Object Model'i kullanır
 - XML mesajlarının okunmasında pull tabanlı yöntem uygulanır

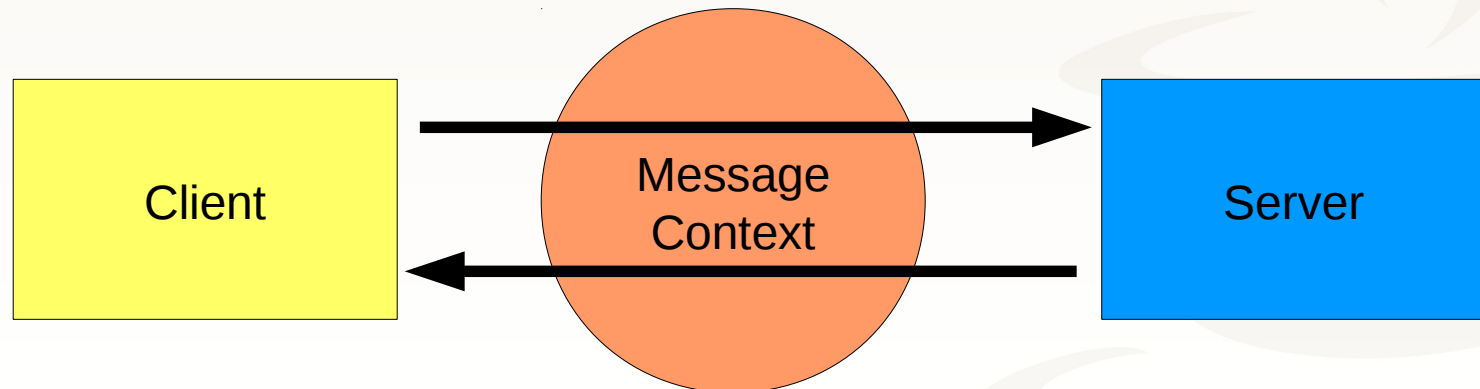
WebService MessageFactory

- Default olarak iki MessageFactory implemantasyonu da **SOAP 1.1**'i destekler
- İstenirse SOAP versiyonu **1.2**'ye aşağıdaki gibi değiştirilebilir

```
<bean id="messageFactory"  
      class="org.springframework.ws.soap.saaj.SaajSoapMessageFactory">  
  <property name="soapVersion">  
    <util:constant static-field=  
      "org.springframework.ws.soap.SoapVersion.SOAP_12"/>  
    </property>  
  </bean>
```

MessageContext

- Web servis **request-response** ikilisini ifade eder
- **WebServiceTemplate** ile istemci tarafında yaratılır
- Sunucu tarafında ise **transport spesifik stream**'den okunarak elde edilir



TransportContext

- SOAP transport bağımsız'dır
- **Mesaj-endpoint eşlemesinin** mesaj içeriğine göre yapılmasının nedeni budur
- Bundan dolayı mesajın geldiği **HTTP URL'ine göre** endpoint eşlemesi **yapılmaz**
- Ancak zaman zaman başka nedenlerle HTTP bağlantısına erişmek gerekebilir
- **TransportContext** bu amaçla **WebServiceConnection'a** erişimi mümkün kılar

TransportContext ile HTTP Bağlantısına Erişim

```
TransportContext context =  
    TransportContextHolder.getTransportContext();  
  
HttpServletConnection connection =  
    (HttpServletConnection )context.getConnection();  
  
HttpServletRequest request = connection.getHttpServletRequest();  
  
String ipAddress = request.getRemoteAddr();  
  
HttpServletResponse response = connection.getHttpServletResponse();  
  
int status = response.getStatus();
```


İletişim

- **Harezmi** Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- info@java-egitimleri.com

