selsoft
build better, deliver faster

Developing RESTful Web Services with Java

Chapter 3: Fundamentals Web Services

Eğitmen:

Akın Kaldıroğlu

Çevik Yazılım Geliştirme ve Java Uzmanı

Topics

- Web Service
- SOAP vs. RESTful WS

Web Service

selsoft

build better, deliver faster

Web Service - I

- Original definition of **Web service** by w3.org is:
- Web Service is a software application identified by a URI [IETF RFC 2396], whose interfaces and binding are capable of being defined, described and discovered by XML artifacts and supports direct interactions with other software applications using XML based messages via Internet-based protocols.
- https://www.w3.org/TR/2002/WD-ws-desc-reqs-20021028/
- This definition defines SOAP-based web services.

SOAP

- Web services historically started as SOAP-based web services.
- **SOAP** stands of **Simple Object Access Protocol**.
- SOAP-based web services uses SOAP as a protocol and XML for messages.
- SOAP-based web services are described in **WSDL** (Web Service Description language) documents.

RESTful

- **REST** stands for **REpresentational State Transfer**.
- It is introduced by Roy Fielding's in his doctoral thesis *Architectural Styles and the Design of Network-based Software Architecture* in 2000.
- In his thesis he mainly made a research regarding how to apply the architecture of web to applications.
- And he found some architectural styles that he called **REST**.

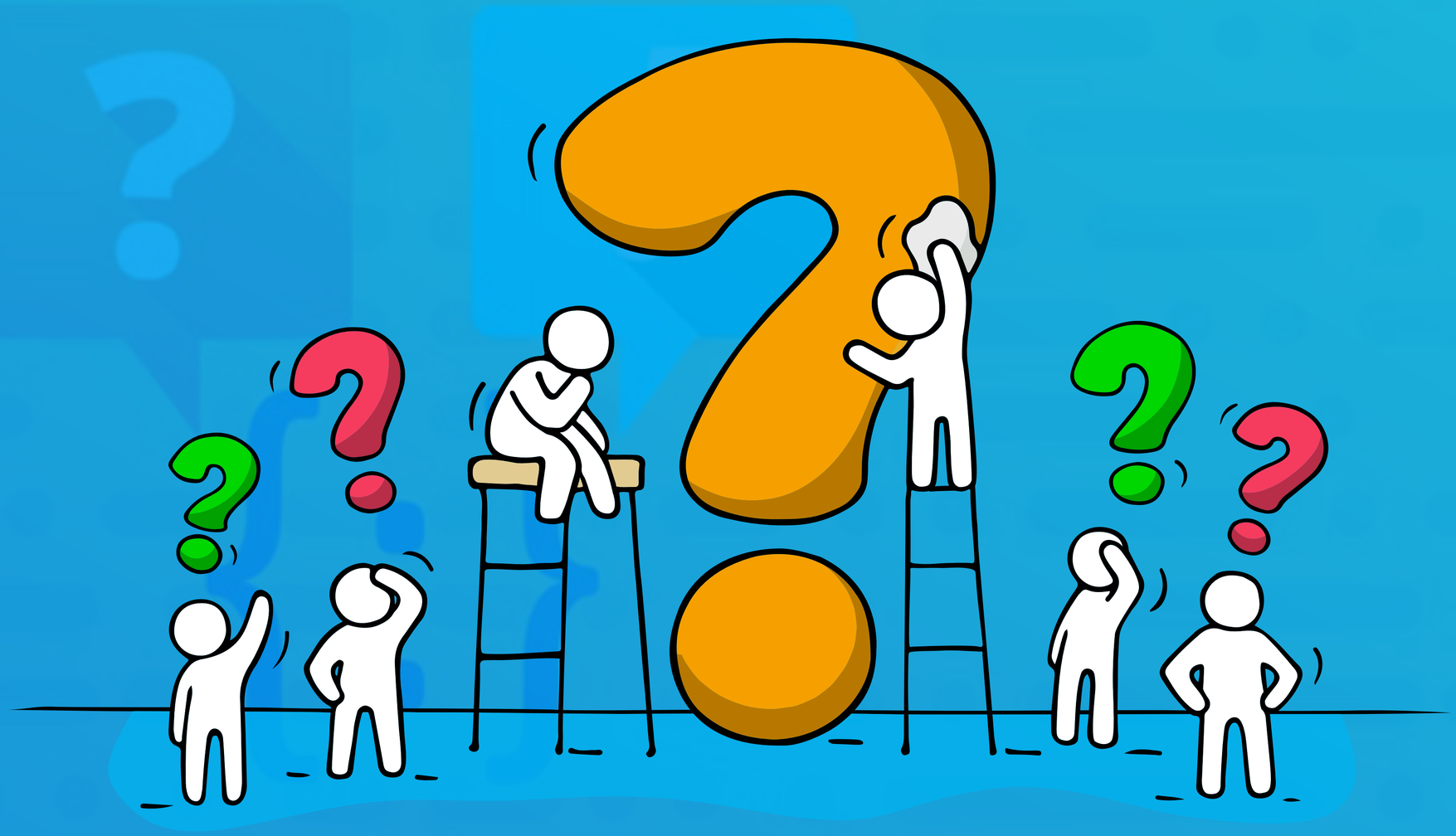
Characteristics - I

- Web services and web applications have following characteristics:
  - Machine-to-machine vs. user-to machine
  - Distributed vs. centralized, monolithic
  - Language-independent vs. language-dependent
  - Message-driven vs. method/function calls
    - Massages in XMl/JSON, etc. formats vs. language types

Characteristics - II

- Loosely-coupled vs. tightly coupled
- Mostly stateless vs. statefull
- Using different transportation vs. mostly using what langauge provides
- Unefficient processing vs. efficient processing

Time for questions!



selsoft
build better, deliver faster

info@selsoft.com.tr
selsoft.com.tr

SOAP & RESTful Web Service

selsoft
build better, deliver faster

SOAP vs. RESTful Web Services - I

- There are several aspects to compare SOAP-based web services with RESTful web services:
  - SOAP is a protocol whereas REST is an architectural pattern or style.
  - SOAP uses interfaces to expose service functionality while REST uses URLs to access resources.
    - So REST resourse-oriented.
  - SOAP only works with XML formats whereas REST can work with plain text, HTML , XML, and JSON.

SOAP vs. RESTful Web Services - II

- SOAP is more heavyweight than REST in terms of their overhead in request and response.
- SOAP cannot make use of REST whereas REST can make use of SOAP.

- So REST is more simple than SOAP,
- SOAP is more structured than REST.
- REST is easier to learn than SOAP.

URLs

- SOAP-based web service uses few URIs (nouns), many custom methods (verbs) described in WSDL.
  - They use HTTP as transport for SOAP messages.

```
SelamWSPort.selamSoyle(“Ali”)
```

- RESTful web service use many resources (nouns), few fixed methods (verbs)
  - They use HTTP as the protocol

```
GET /greet
```

Service vs. Resource

- SOAP-based web services provide interfaces for services such as greeting service or currency converter service, etc.

```
SelamWSPort.selamSoyle(“Ali”)
```

- REST is **Resource-Oriented Architecture** and provides resources such as greet or conversion.

- Resources are manipulated by HTTP methods

```
GET /greet
```

```
POST /greet/selam
```

Transportation

- SOAP is transport-independent so it can be used over any kind of transport (HTTP(S), JMS, SMTP).
- REST works only over HTTP(S).

Standards - I

- SOAP has a well-defined service contact via **WSDL**.
  - https://www.w3.org/TR/wsdl.html
- REST has **WADL** (Web Application Description language) which is not a standard yet.
  - https://www.w3.org/Submission/wadl/
- WSDL and WADL are both schema-based XML standards.

Standards - II

- SOAP has well-defined standards regarding the protocol itself and security, transaction, error-handling, etc.
- REST mainly lacks these standards.

```
POST http://localhost:6060/CurrencyConverterSOAP/converter HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: text/xml;charset=UTF-8
SOAPAction: "get-all-currencies"
Content-Length: 218
Host: localhost:6060
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.5.5 (Java/12.0.1)

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:jav="http://www.javaturk.org">
   <soapenv:Header/>
   <soapenv:Body>
      <jav:all-currencies/>
   </soapenv:Body>
</soapenv:Envelope>
```

```
HTTP/1.1 200
Content-Type: text/xml;charset=utf-8
Transfer-Encoding: chunked
Date: Wed, 09 Dec 2020 14:16:02 GMT
Keep-Alive: timeout=20
Connection: keep-alive

<?xml version='1.0' encoding='UTF-8'?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body><ns2:all-currenciesResponse xmlns:ns2="http://www.javaturk.org">
<Currency-List>TRY</Currency-List>
<Currency-List>USD</Currency-List>
<Currency-List>EUR</Currency-List>
</ns2:all-currenciesResponse>
</S:Body>
</S:Envelope>
```

```
GET http://localhost:6060/CurrencyConverter/resources/conversions/
currencies HTTP/1.1
Accept-Encoding: gzip,deflate
Host: localhost:6060
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.5.5 (Java/12.0.1)
```

```
HTTP/1.1 200
Content-Type: application/json
Content-Length: 19
Date: Wed, 09 Dec 2020 14:31:10 GMT
Keep-Alive: timeout=20
Connection: keep-alive

["TRY","USD","EUR"]
```

ConverterService

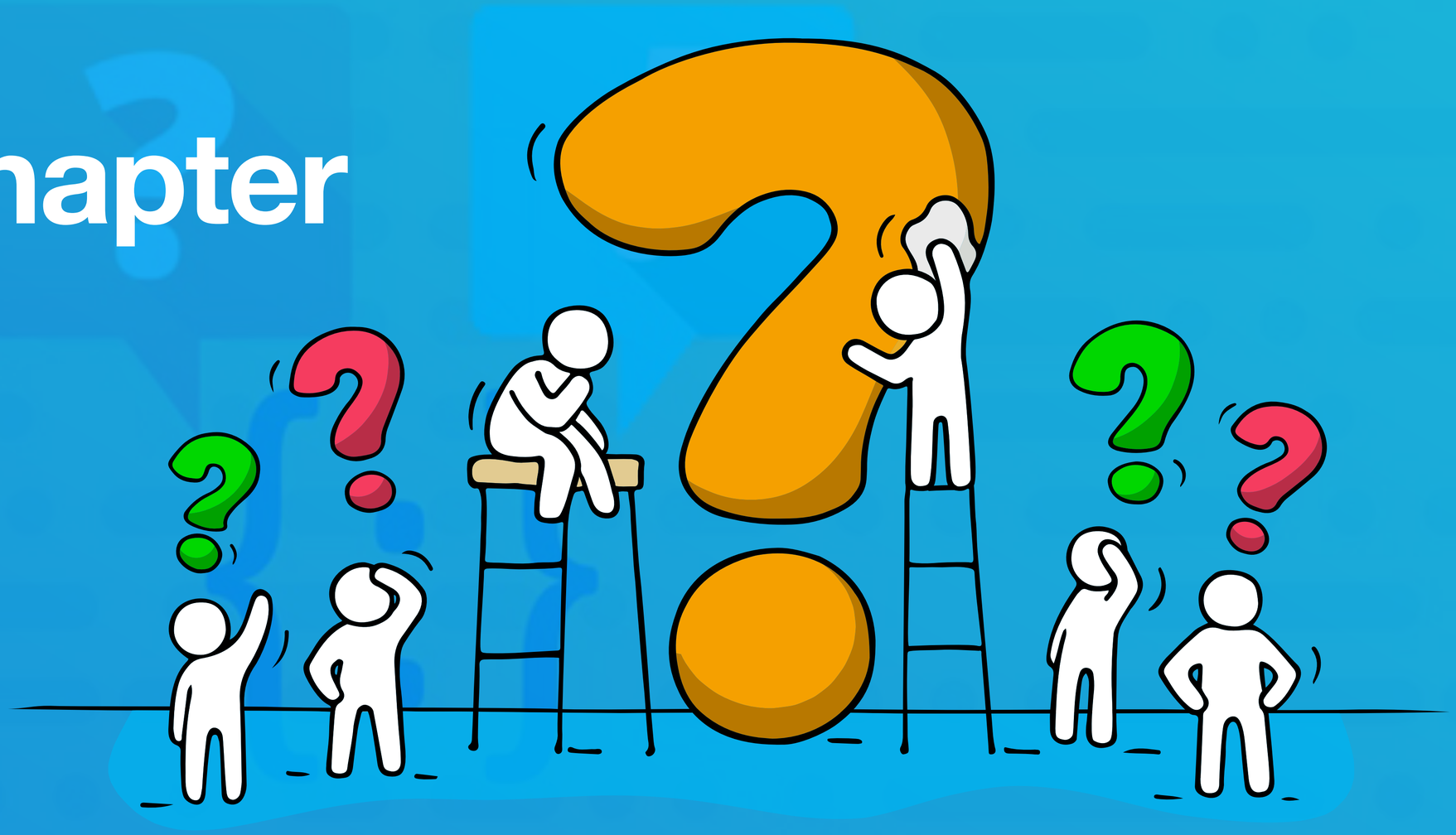
- **ConverterService** project.
- Run it and test it using Postman and SOAPUI.

ConverterServiceSOAP

- **ConverterServiceSOAP** project.
- Run it and test it using Postman and SOAPUI.

End of Chapter

Time for Questions!



selsoft
build better, deliver faster

info@selsoft.com.tr
selsoft.com.tr