

Rozproszone systemy internetowe

Web serwisy

JAX-WS (SOAP web services)

cz.3 SOAP Handlers, Exception ...

Sources and help:

[Web services \(javaTpoint\)](#)

[Java Web Services - Book online](#)

[JAX-WS Tutorial by mkyong](#)

[Java EE Tutorial -Web services](#)

[JAX-WS Release Documentation](#)

[Metro User Guide](#)

Ćwiczenie 1. SOAP Handlers

Zaimplementować web serwis i klienta z użyciem **SOAP Handlers**

SOAP handler is a SOAP message **interceptor**, which is able to intercept incoming or outgoing SOAP message and manipulate its values.

For example, attach a SOAP handler in client side, which will inject client's computer MAC address into the SOAP header block for every outgoing SOAP message that is send by the client.

In server side, attach another SOAP handler, to retrieve back the client's MAC address in SOAP header block from every incoming SOAP message. So that the server side is able to determine which computer is allow to access the published service.

Tutorial: [SOAP handler](#)

Handlers in JAX-WS Documentation: [Handlers](#)

Ćwiczenie 2. Application level authentication

Zaimplementować klienta, który wyśle w **nagłówku HTTP** zapytania-request username i password.

Zaimplementować serwis, który odbierze z nagłówka HTTP zapytania-request username i password.

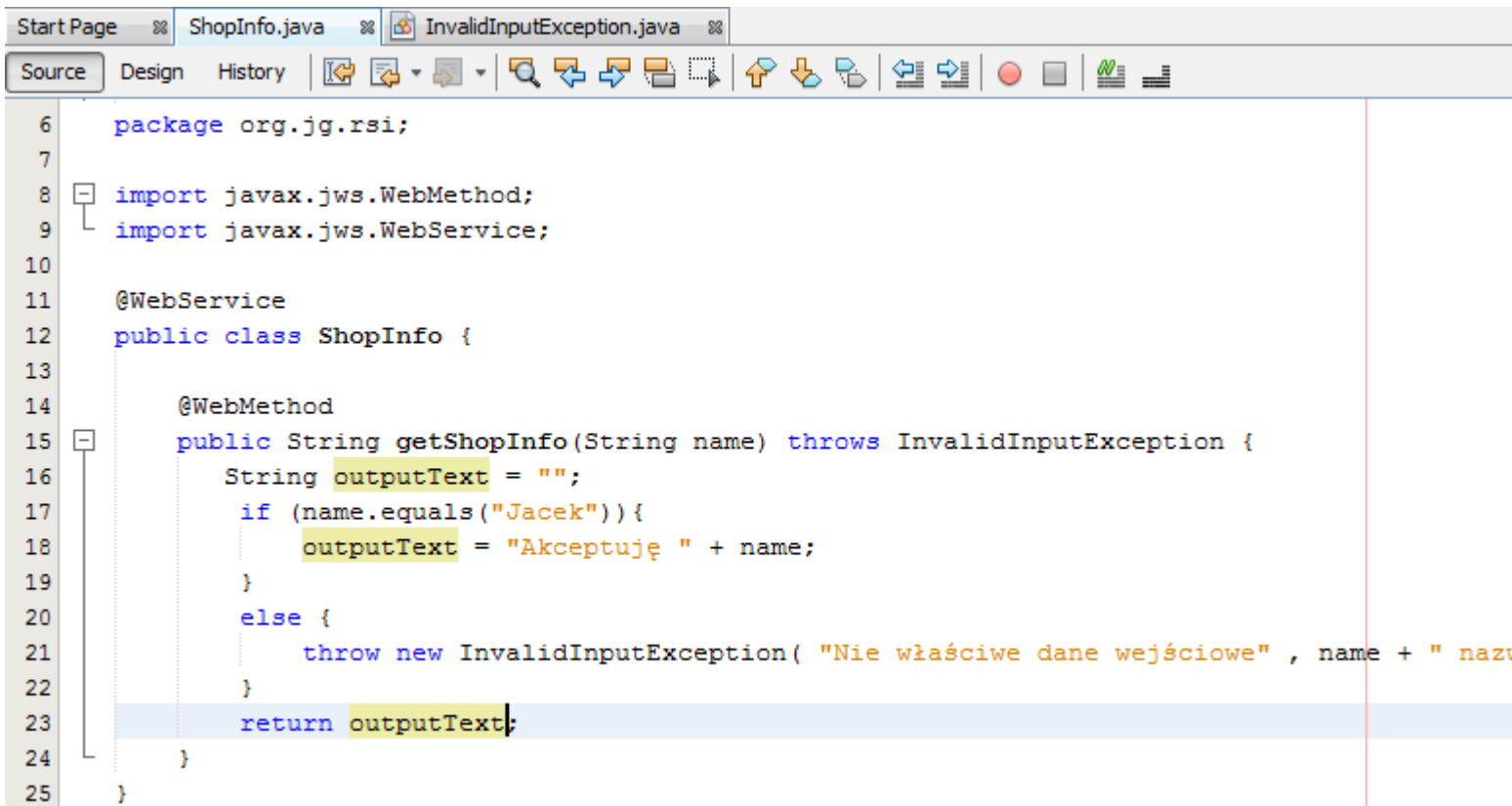
Tutorial: Authentication with JAX-WS [link](#)

- **Monitoruj przesyłane komunikaty SOAP, i nagłówki HTTP**

Ćwiczenie 3. Generowanie exception

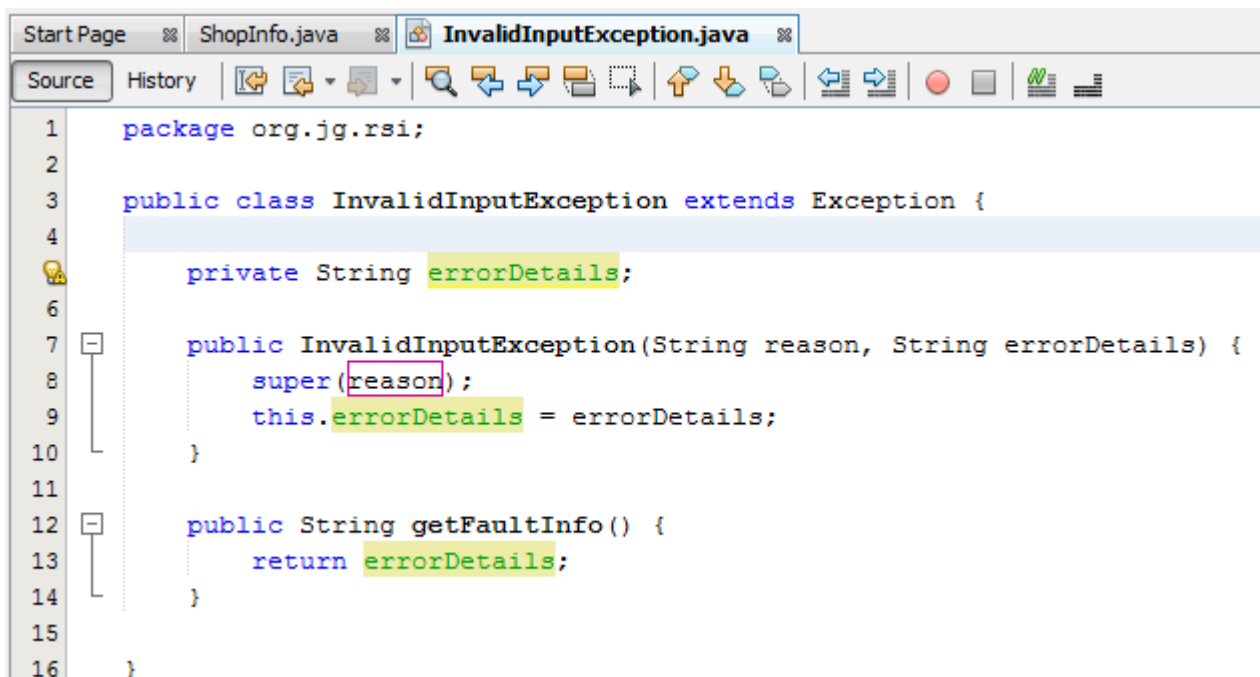
1. Zaimplementować web serwis, który przy pewnym parametrze (założmy, że błędnym) wywołania zgłosi **wyjątek**.
2. Przetestuj jego działanie przy użyciu SoapUI.
3. Zaimplementuj klienta, który połączy się serwisem i obsłuży wyjątki.

Przykład:



```
6 package org.jg.rsi;
7
8 import javax.jws.WebMethod;
9 import javax.jws.WebService;
10
11 @WebService
12 public class ShopInfo {
13
14     @WebMethod
15     public String getShopInfo(String name) throws InvalidInputException {
16         String outputText = "";
17         if (name.equals("Jacek")) {
18             outputText = "Akceptuję " + name;
19         }
20         else {
21             throw new InvalidInputException( "Nie właściwe dane wejściowe" , name + " nazwa" );
22         }
23         return outputText;
24     }
25 }
```

Exception:



```
1 package org.jg.rsi;
2
3 public class InvalidInputException extends Exception {
4
5     private String errorDetails;
6
7     public InvalidInputException(String reason, String errorDetails) {
8         super(reason);
9         this.errorDetails = errorDetails;
10    }
11
12    public String getFaultInfo() {
13        return errorDetails;
14    }
15
16 }
```

Testowanie serwisu - wywołanie błędu:

SOAP Request 1

http://localhost:8080/WebServicesGlass/ShopInfoService

XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <rsi:getShopInfo>
      <!--Optional:-->
      <arg0>Zły parametr</arg0>
    </rsi:getShopInfo>
  </soapenv:Body>
</soapenv:Envelope>
```

XML

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <S:Fault xmlns:ns4="http://www.w3.org/2003/05/soap-envelope":>
      <faultcode>S:Server</faultcode>
      <faultstring>Nie właściwe dane wejściowe</faultstring>
      <detail>
        <ns2:InvalidInputException xmlns:ns2="http://rsi.jg.org/">
          <faultInfo>Zły parametr nazwa jest niewłaściwa.</faultInfo>
          <message>Nie właściwe dane wejściowe</message>
        </ns2:InvalidInputException>
      </detail>
    </S:Fault>
  </S:Body>
</S:Envelope>
```

Testowanie serwisu - bez wywoływania błędu:

SOAP Request 1

http://localhost:8080/WebServicesGlass/ShopInfoService

XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <rsi:getShopInfo>
      <!--Optional:-->
      <arg0>Jacek</arg0>
    </rsi:getShopInfo>
  </soapenv:Body>
</soapenv:Envelope>
```

XML

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getShopInfoResponse xmlns:ns2="http://rsi.jg.org/">
      <return>Akceptuję Jacek</return>
    </ns2:getShopInfoResponse>
  </S:Body>
</S:Envelope>
```

<http://localhost:8080/WebServicesGlass/ShopInfoService?xsd=1>

```
- <xs:complexType name="getShopInfoResponse">
  - <xs:sequence>
    <xs:element name="return" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="InvalidInputException">
  - <xs:sequence>
    <xs:element name="faultInfo" type="xs:string" minOccurs="0"/>
    <xs:element name="message" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

<http://localhost:8080/WebServicesGlass/ShopInfoService?WSDL>

```
</message>
- <portType name="ShopInfo">
  - <operation name="getShopInfo">
    <input wsam:Action="http://rsi.jg.org/ShopInfo/getShopInfoRequest" message="tns:getShopInfo"/>
    <output wsam:Action="http://rsi.jg.org/ShopInfo/getShopInfoResponse" message="tns:getShopInfoResponse"/>
    <fault message="tns:InvalidInputException" name="InvalidInputException" wsam:Action="http://rsi.jg.org/ShopInfo/getS
  </operation>
</portType>
```