

01 Hands On Exercise - Services

Inhaltsverzeichnis

Introduction	3
System Requirements	3
Overview	3
Getting Started	3
Starting the Client	3
Starting the Webservice	4
Requests	4
SOAP	5
Add Customer	5
Add Order	6
Delete Order	6
Delete Customer	7
Get Orders	8
Get Customers	9
WSDL	10
REST	13
Add Customer	13
Add Order	13
Delete Order	14
Delete Customer	14
Get Orders	15
Get Customers	15
REST vs. SOAP	16

Introduction

Welcome to the help of **"01 Hands On Exercise - Services"**

Created with the Personal Edition of HelpNDoc: [Easily create HTML Help documents](#)

System Requirements

- .NET Framework 4.5.2

Created with the Personal Edition of HelpNDoc: [Single source CHM, PDF, DOC and HTML Help creation](#)

Overview

Webservices

Both webservises are base on NodeJS and use the same interaction logic and data structure.

Libraries used for the REST webservice:

- express - <http://expressjs.com/>

ExpressJS is used because it is a lightweight web application framework which helps you organize web applications. This library was taken because NodeJS has no build in REST function.

Libraries used for the SOAP webservice:

- soap-server - <http://github.com/tan-tan-kanarek/node-soap-server>

Soap-Server is used because NodeJS has no build in SOAP function.

Client

The client is a simple console client based on .NET.

The decision was made because a .NET console client is easy and fast to implement.

Created with the Personal Edition of HelpNDoc: [Free EPub and documentation generator](#)

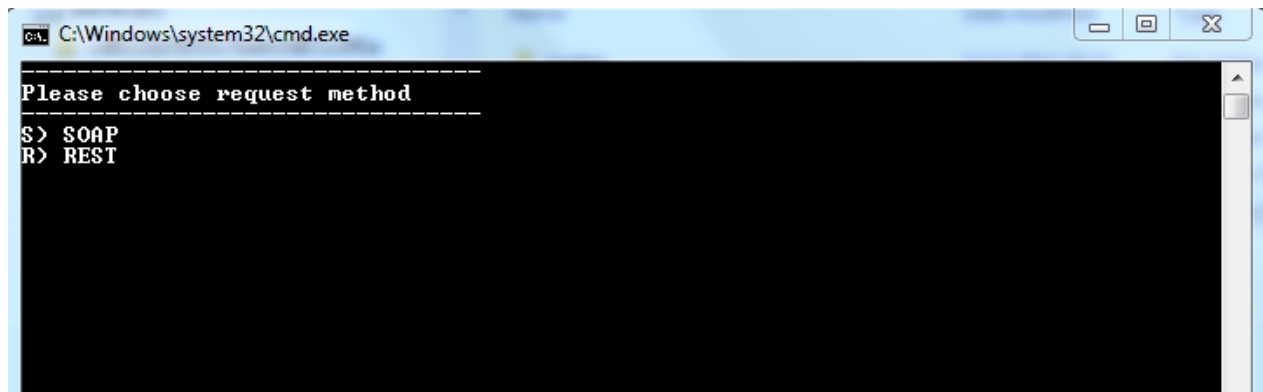
Getting Started

Created with the Personal Edition of HelpNDoc: [Easily create EPub books](#)

Starting the Client

Just use the "start soap-rest client.bat" file to start the client.

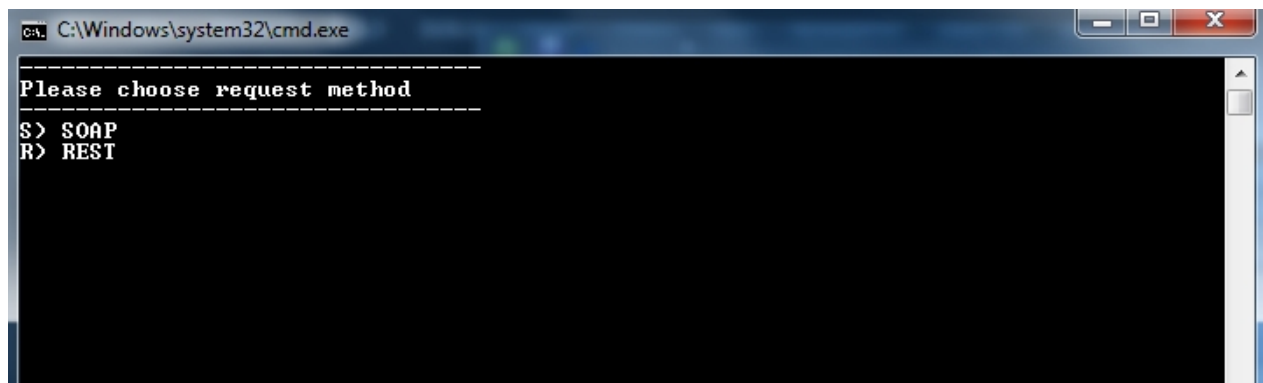
This will give you a basic interface where you can choose your request type.



```
C:\Windows\system32\cmd.exe

Please choose request method
-----
S> SOAP
R> REST
```

On the next screen you can choose your request method.



```
C:\Windows\system32\cmd.exe

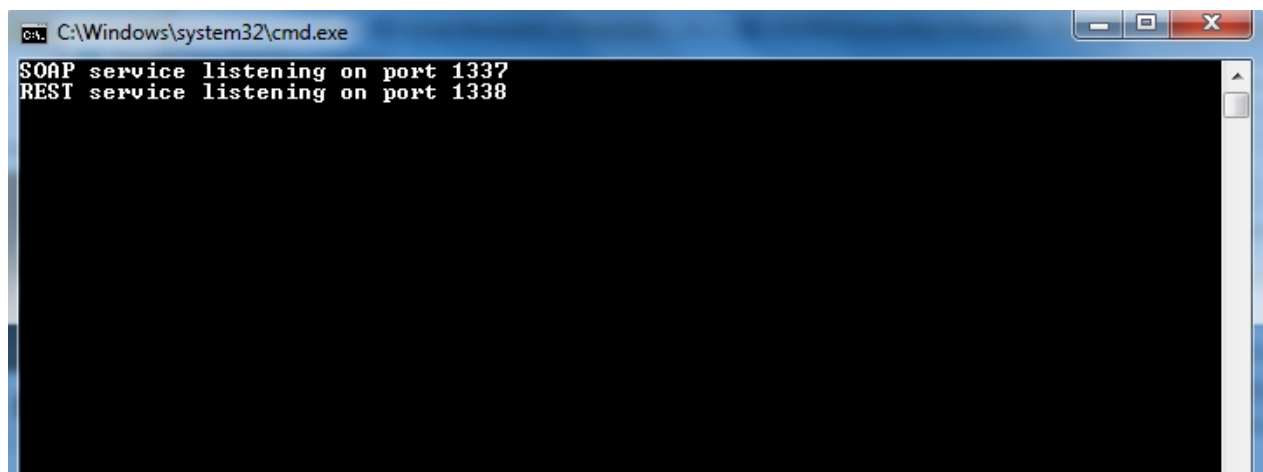
Please choose request method
-----
S> SOAP
R> REST
```

Created with the Personal Edition of HelpNDoc: [Easily create iPhone documentation](#)

Starting the Webservice

Just use the "start rest-soap service.bat" file to start the client.

This will start both webservices in the console on different ports.



```
C:\Windows\system32\cmd.exe

SOAP service listening on port 1337
REST service listening on port 1338
```

Created with the Personal Edition of HelpNDoc: [Easy EPub and documentation editor](#)

Requests

This chapter describes the two different requests and their methods

- SOAP
- REST

SOAP

Simple Object Access Protocol

The following requests are available if you choose SOAP as request method:

- Add Customer
- Add Order
- Delete Order
- Delete Customer
- Get Orders
- Get Customers

Add Customer

A customer can be added via the "Add Customer" request by specifying the customer name.

Request

POST /SOAPWebService HTTP/1.1
SOAPAction: addCustomer
Content-Type: text/xml;charset="utf-8"
Accept: text/xml
Host: localhost:1337
Content-Length: 264
Connection: Close

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <addCustomer>
      <customerName>Scherr</customerName>
    </addCustomer>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

HTTP/1.1 200 OK
Content-Type: text/xml
Date: Thu, 18 Dec 2014 20:31:23 GMT
Connection: close
Transfer-Encoding: chunked

```
115
<soap:Envelope
.xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
.xmlns:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  <soap:Body>
    ..<addCustomerResponse>
    ...<return>
```

```
....<Success>true</Success>
...</return>
..</addCustomerResponse>
.</soap:Body>
</soap:Envelope>
```

Created with the Personal Edition of HelpNDoc: [Easily create HTML Help documents](#)

Add Order

An order can be added via the "Add Order" request by specifying the customer name.
The order name will be generated automatically.

Request

```
POST /SOAPWebService HTTP/1.1
SOAPAction: addOrder
Content-Type: text/xml;charset="utf-8"
Accept: text/xml
Host: localhost:1337
Content-Length: 297
Connection: Close
```

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <addOrder>
      <customerName>Scherr</customerName>
      <orderName>Smarties</orderName>
    </addOrder>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
HTTP/1.1 200 OK
Content-Type: text/xml
Date: Thu, 18 Dec 2014 20:31:31 GMT
Connection: close
Transfer-Encoding: chunked
```

```
10f
<soap:Envelope
.xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
.soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
.<soap:Body>
..<addOrderResponse>
...<return>
....<Success>true</Success>
...</return>
..</addOrderResponse>
.</soap:Body>
</soap:Envelope>
```

Created with the Personal Edition of HelpNDoc: [Write eBooks for the Kindle](#)

Delete Order

All orders can be deleted with the "Delete Order" request by specifying the customer name.

Request

```
POST /SOAPWebService HTTP/1.1
SOAPAction: deleteOrder
Content-Type: text/xml;charset="utf-8"
Accept: text/xml
Host: localhost:1337
Content-Length: 303
Connection: Close
```

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <deleteOrder>
      <customerName>Scherr</customerName>
      <orderId>Smarties</orderId>
    </deleteOrder>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
HTTP/1.1 200 OK
Content-Type: text/xml
Date: Thu, 18 Dec 2014 20:31:51 GMT
Connection: close
Transfer-Encoding: chunked
```

```
115
<soap:Envelope
  xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
  soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  <soap:Body>
    ..<deleteOrderResponse>
      ...<return>
      ....<Success>true</Success>
      ...</return>
    ..</deleteOrderResponse>
  </soap:Body>
</soap:Envelope>
```

Created with the Personal Edition of HelpNDoc: [Produce electronic books easily](#)

Delete Customer

A customer can be deleted with the "Delete Customer" request by specifying the customer name.

Request

```
POST /SOAPWebService HTTP/1.1
SOAPAction: deleteCustomer
Content-Type: text/xml;charset="utf-8"
Accept: text/xml
Host: localhost:1337
```

Content-Length: 270

Connection: Close

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <deleteCustomer>
      <customerName>Scherr</customerName>
    </deleteCustomer>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

HTTP/1.1 200 OK

Content-Type: text/xml

Date: Thu, 18 Dec 2014 20:31:56 GMT

Connection: close

Transfer-Encoding: chunked

11b

```
<soap:Envelope
.xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
.soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
.<soap:Body>
..<deleteCustomerResponse>
...<return>
....<Success>true</Success>
...</return>
..</deleteCustomerResponse>
.</soap:Body>
</soap:Envelope>
```

Created with the Personal Edition of HelpNDoc: [Free help authoring tool](#)

Get Orders

All orders for a specific customer are retrieved with the "Get Orders" request by specifying the customer name.

Request

POST /SOAPWebService HTTP/1.1

SOAPAction: getOrders

Content-Type: text/xml; charset="utf-8"

Accept: text/xml

Host: localhost:1337

Content-Length: 260

Connection: Close

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <getOrders>
      <customerName>Scherr</customerName>
    </getOrders>
  </SOAP-ENV:Body>
```



```
</SOAP-ENV:Envelope>
```

Response

```
HTTP/1.1 200 OK
Content-Type: text/xml
Date: Thu, 18 Dec 2014 20:31:42 GMT
Connection: close
Transfer-Encoding: chunked
```

```
12f
<soap:Envelope
.xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
.soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
.<soap:Body>
..<getOrdersResponse>
...<return>
....<Scherr>
.....<order0>Smarties</order0>
....</Scherr>
...</return>
..</getOrdersResponse>
.</soap:Body>
</soap:Envelope>
```

Created with the Personal Edition of HelpNDoc: [Full-featured Documentation generator](#)

Get Customers

All customers can be retrieved with the "Get Customers" request.

Request

```
POST /SOAPWebService HTTP/1.1
SOAPAction: getCustomers
Content-Type: text/xml;charset="utf-8"
Accept: text/xml
Host: localhost:1337
Content-Length: 250
Connection: Close
```

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <getCustomers>
      <all>
      </all>
    </getCustomers>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
HTTP/1.1 200 OK
Content-Type: text/xml
Date: Thu, 18 Dec 2014 20:31:37 GMT
Connection: close
```

Transfer-Encoding: chunked

16f

```
<soap:Envelope
.xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
.soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
.<soap:Body>
..<getCustomersResponse>
...<return>
....<Murrent>
.....<order0>Coffee</order0>
....</Murrent>
....<Scherr>
.....<order0>Smarties</order0>
....</Scherr>
...</return>
..</getCustomersResponse>
.</soap:Body>
</soap:Envelope>
```

Created with the Personal Edition of HelpNDoc: [Single source CHM, PDF, DOC and HTML Help creation](#)

WSDL

```
<definitions
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
  xmlns:wsp="http://www.w3.org/ns/ws-policy"
  xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:tns="http://server.soap.com/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  targetNamespace="http://server.soap.com/"
  name="SOAPWebServiceService">
  <types>
    <xsd:schema version="1.0" targetNamespace="http://server.soap.com/">
      </xsd:schema>
    </types>
    <message name="getCustomers">
      <part name="all" element="xsd:string" />
    </message>
    <message name="getCustomersResponse">
      <part name="return" element="xsd:string" />
    </message>
    <message name="getOrders">
      <part name="customerName" element="xsd:string" />
    </message>
    <message name="getOrdersResponse">
      <part name="return" element="xsd:string" />
    </message>
    <message name="addOrder">
      <part name="customerName" element="xsd:string" />
      <part name="orderName" element="xsd:string" />
    </message>
    <message name="addOrderResponse">
      <part name="return" element="xsd:string" />
    </message>
    <message name="addCustomer">
      <part name="customerName" element="xsd:string" />
```

```

</message>
<message name="addCustomerResponse">
  <part name="return" element="xsd:string" />
</message>
<message name="deleteOrder">
  <part name="customerName" element="xsd:string" />
  <part name="orderName" element="xsd:string" />
</message>
<message name="deleteOrderResponse">
  <part name="return" element="xsd:string" />
</message>
<message name="deleteCustomer">
  <part name="customerName" element="xsd:string" />
</message>
<message name="deleteCustomerResponse">
  <part name="return" element="xsd:string" />
</message>
<portType name="SOAPWebService">
  <operation name="getCustomers">
    <input wsam:Action="getCustomers" message="tns:getCustomers" />
    <output wsam:Action="http://server.soap.com/SOAPWebService/
getCustomersResponse"
      message="tns:getCustomersResponse" />
    </operation>
    <operation name="getOrders">
      <input wsam:Action="getOrders" message="tns:getOrders" />
      <output wsam:Action="http://server.soap.com/SOAPWebService/
getOrdersResponse"
        message="tns:getOrdersResponse" />
      </operation>
      <operation name="addOrder">
        <input wsam:Action="addOrder" message="tns:addOrder" />
        <output wsam:Action="http://server.soap.com/SOAPWebService/
addOrderResponse"
          message="tns:addOrderResponse" />
        </operation>
        <operation name="addCustomer">
          <input wsam:Action="addCustomer" message="tns:addCustomer" />
          <output wsam:Action="http://server.soap.com/SOAPWebService/
addCustomerResponse"
            message="tns:addCustomerResponse" />
          </operation>
          <operation name="deleteOrder">
            <input wsam:Action="deleteOrder" message="tns:deleteOrder" />
            <output wsam:Action="http://server.soap.com/SOAPWebService/
deleteOrderResponse"
              message="tns:deleteOrderResponse" />
            </operation>
            <operation name="deleteCustomer">
              <input wsam:Action="deleteCustomer" message="tns:deleteCustomer" />
              <output wsam:Action="http://server.soap.com/SOAPWebService/
deleteCustomerResponse"
                message="tns:deleteCustomerResponse" />
              </operation>
            </portType>
            <binding name="SOAPWebServicePortBinding" type="tns:SOAPWebService">
              <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
                style="rpc" />
              <operation name="getCustomers">
                <soap:operation soapAction="getCustomers" />

```

```

        <input name="getCustomers">
            <soap:body use="literal" />
        </input>
        <output name="getCustomersResponse">
            <soap:body use="literal" />
        </output>
    </operation>
    <operation name="getOrders">
        <soap:operation soapAction="getOrders" />
        <input name="getOrders">
            <soap:body use="literal" />
        </input>
        <output name="getOrdersResponse">
            <soap:body use="literal" />
        </output>
    </operation>
    <operation name="addOrder">
        <soap:operation soapAction="addOrder" />
        <input name="addOrder">
            <soap:body use="literal" />
        </input>
        <output name="addOrderResponse">
            <soap:body use="literal" />
        </output>
    </operation>
    <operation name="addCustomer">
        <soap:operation soapAction="addCustomer" />
        <input name="addCustomer">
            <soap:body use="literal" />
        </input>
        <output name="addCustomerResponse">
            <soap:body use="literal" />
        </output>
    </operation>
    <operation name="deleteOrder">
        <soap:operation soapAction="deleteOrder" />
        <input name="deleteOrder">
            <soap:body use="literal" />
        </input>
        <output name="deleteOrderResponse">
            <soap:body use="literal" />
        </output>
    </operation>
    <operation name="deleteCustomer">
        <soap:operation soapAction="deleteCustomer" />
        <input name="deleteCustomer">
            <soap:body use="literal" />
        </input>
        <output name="deleteCustomerResponse">
            <soap:body use="literal" />
        </output>
    </operation>
</binding>
<service name="SOAPWebServiceService">
    <port name="SOAPWebServicePort" binding="tns:SOAPWebServicePortBinding">
        <soap:address location="http://localhost:1337/SOAPWebService" />
    </port>
</service>
</definitions>

```

REST

Representational State Transfer

The following requests are available if you choose REST as request method:

- Add Customer
- Add Order
- Delete Order
- Delete Customer
- Get Orders
- Get Customers

Add Customer

A customer can be added via the "Add Customer" request by specifying the customer name.

Request

```
PUT /customer/add/Murrent HTTP/1.1
Accept: application/json, application/xml, text/json, text/x-json, text/javascript, text/xml
User-Agent: RestSharp/105.0.1.0
Host: localhost:1338
Content-Length: 0
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
```

Response

```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 16
Date: Thu, 18 Dec 2014 20:13:29 GMT
Connection: keep-alive
```

```
{"Success":true}
```

Add Order

An order can be added via the "Add Order" request by specifying the customer name.
The order name will be generated automatically.

Request

```
PUT /order/add/Murrent/Coffee HTTP/1.1
Accept: application/json, application/xml, text/json, text/x-json, text/javascript, text/xml
User-Agent: RestSharp/105.0.1.0
Host: localhost:1338
Content-Length: 0
```

Accept-Encoding: gzip, deflate

Response

HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 16
Date: Thu, 18 Dec 2014 20:13:43 GMT
Connection: keep-alive

```
{"Success":true}
```

Created with the Personal Edition of HelpNDoc: [Easily create iPhone documentation](#)

Delete Order

All orders can be deleted with the "Delete Order" request by specifying the customer name.

Request

DELETE /order/delete/Murrent/Coffee HTTP/1.1
Accept: application/json, application/xml, text/json, text/x-json, text/javascript, text/xml
User-Agent: RestSharp/105.0.1.0
Host: localhost:1338
Content-Length: 0
Accept-Encoding: gzip, deflate

Response

HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 16
Date: Thu, 18 Dec 2014 20:13:58 GMT
Connection: keep-alive

```
{"Success":true}
```

Created with the Personal Edition of HelpNDoc: [Free Kindle producer](#)

Delete Customer

A customer can be deleted with the "Delete Customer" request by specifying the customer name.

Request

DELETE /customer/delete/Murrent HTTP/1.1
Accept: application/json, application/xml, text/json, text/x-json, text/javascript, text/xml
User-Agent: RestSharp/105.0.1.0
Host: localhost:1338
Content-Length: 0
Accept-Encoding: gzip, deflate

Response

HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 16
Date: Thu, 18 Dec 2014 20:14:04 GMT
Connection: keep-alive

```
{"Success":true}
```

Created with the Personal Edition of HelpNDoc: [Full-featured EPub generator](#)

Get Orders

All orders for a specific customer are retrieved with the "Get Orders" request by specifying the customer name.

Request

GET /order/Murrent HTTP/1.1
Accept: application/json, application/xml, text/json, text/x-json, text/javascript, text/xml
User-Agent: RestSharp/105.0.1.0
Host: localhost:1338
Accept-Encoding: gzip, deflate

Response

HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 10
ETag: W/"a-ecc64dd7"
Date: Thu, 18 Dec 2014 20:25:58 GMT
Connection: keep-alive

```
["Coffee"]
```

Created with the Personal Edition of HelpNDoc: [Easily create CHM Help documents](#)

Get Customers

All customers can be retrieved with the "Get Customers" request.

Request

GET /customers HTTP/1.1
Accept: application/json, application/xml, text/json, text/x-json, text/javascript, text/xml
User-Agent: RestSharp/105.0.1.0
Host: localhost:1338
Accept-Encoding: gzip, deflate

Response

HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 40
ETag: W/"28-258d088c"

Date: Thu, 18 Dec 2014 20:13:46 GMT

Connection: keep-alive

```
[{"Name":"Murrent","Orders":["Coffee"]}]
```

Created with the Personal Edition of HelpNDoc: [Easily create HTML Help documents](#)

REST vs. SOAP

SOAP

SOAP is actually agnostic of the underlying transport protocol and can be sent over almost any protocol such as HTTP, SMTP, TCP, or JMS. SOAP message itself must be XML-formatted. The root element in this case is the Envelope element. This contains two required elements: the Header and the Body. The rest of the elements in this message are described by the WSDL.

REST

REST is an architectural style, unlike SOAP which is a standardized protocol. REST makes use of existing and widely adopted technologies, specifically HTTP, and does not create any new standards.

Comparison

One of the major benefits of SOAP is that you have a WSDL service description. You can pretty much discover the service automatically and generate a usable client proxies from that service description.

RESTful services are lightweight, because they don't have a lot of extra XML markup in comparison to SOAP. To invoke RESTful API all you need is a browser or HTTP stack and pretty much every device or machine connected to a network has that.

Created with the Personal Edition of HelpNDoc: [Easily create EPub books](#)
