

SOA PROJE ÖDEVİ

İSTERLER

Database:

Veriler:

DB Browser for SQLite - C:\Users\mrtag\Desktop\NodejsSOA\NodejsApi\Db\Products.db

Dosya Düzenle Görünüm Araçlar Yardım

Yeni Veritabanı Veritabanı Aç Değişiklikleri Kaydet Değişiklikleri Geri Al Proje Aç Projeyi Kaydet Veritabanı Ekle Veritabanı Kapat

Veritabanı Yapısı Veriyi Görüntüle SQL kodunu yürüt Pragmaları Düzenle

Tablo: Orders

	Id	Title	Quantity	Message	City
	Filtre	Filtre	Filtre	Filtre	Filtre
1	1	PostmanTitle2	PostmanQuantity2	PostmanMessage2	PostmanCity2
2	2	kjhaskjd	ajksdlaskj	dlaksdjlja	djalsk
3	6	PostmanTitle2	PostmanQuantity2	PostmanMessage2	PostmanCity2

Veritabanı Yapısı:

İsim	Tip	Şema
Tablolar (2)		
Orders		CREATE TABLE "Orders" ("Id" INTEGER NOT NULL UNIQUE, "Title" TEXT, "Quantity" TEXT, "Message" TEXT, "City" TEXT, PRIMARY KEY("Id" AUTOINCREMENT))
Id	INTEGER	"Id" INTEGER NOT NULL UNIQUE
Title	TEXT	"Title" TEXT
Quantity	TEXT	"Quantity" TEXT
Message	TEXT	"Message" TEXT

Backend:

```
NodejsApi > CRUD > dboperations.js > GetOrders > db.serialize() callback > db.all("SELECT * from Orders") callback
1 const sqlite3 = require("sqlite3").verbose();
2 var order = require("../Model/Order");
3
4 let db = new sqlite3.Database(
5   "../Db/Products.db",
6   sqlite3.OPEN_READWRITE,
7   (err) => {
8     if (err) {
9       console.error(err.message);
10    }
11    console.log("Connected to the chinook database.");
12  }
13 );
14
15 async function GetOrders() {
16   db.serialize(() => {
17     db.all("SELECT * from Orders", function (err, rows) {
18       if (err) {
19         console.log(err);
20       } else {
21         console.log(rows);
22       }
23     });
24   });
25 }
26
27 async function GetOrder(orderId) {
28   db.serialize(() => {
29     db.each("SELECT * FROM Orders WHERE id =?", [orderId], function (err, row) {
30       if (err) {
31         res.send("Error encountered while displaying");
32         return console.error(err.message);
33       } else {
34         console.log(row);
35       }
36     });
37   });
38 }
39
40 async function AddOrder(order) {
```

- **Dboperations.js :**

```
const sqlite3 = require("sqlite3").verbose();
```

```
var order = require("../Model/Order");
```

```
let db = new sqlite3.Database(  
  "./Db/Products.db",  
  sqlite3.OPEN_READWRITE,  
  (err) => {  
    if (err) {  
      console.error(err.message);  
    }  
    console.log("Connected to the chinook database.");  
  }  
);
```

```
async function GetOrders() {  
  db.serialize(() => {  
    db.all("SELECT * from Orders", function (err, rows) {  
      if (err) {  
        console.log(err);  
      } else {  
        console.log(rows);  
      }  
    });  
  });  
}
```

```
async function GetOrder(orderId) {  
  db.serialize(() => {  
    db.each("SELECT * FROM Orders WHERE id =?", [orderId], function (err, row)  
    {  
      if (err) {
```

```

        res.send("Error encountered while displaying");
        return console.error(err.message);
    } else {
        console.log(row);
    }
});
});
}

async function AddOrder(order) {
    db.serialize(() => {
        db.run(
            "INSERT INTO Orders (Title,Quantity,Message,City) VALUES(?,?,?,?)",
            [order.Title, order.Quantity, order.Message, order.City],
            function (err) {
                if (err) {
                    return console.log(err.message);
                }
                console.log("Entry added");
            }
        );
    });
}

async function DeleteOrder(orderId) {
    db.serialize(() => {
        db.run("DELETE FROM Orders WHERE id =?", [orderId], function (err) {
            if (err) {
                return console.log(err.message);
            }
            console.log("Entry deleted");
        });
    });
}

```

```

}

async function UpdateOrder(orderId, order) {
  db.serialize(() => {
    db.run(
      "UPDATE Orders SET Title = ?, Quantity = ?, Message = ?, City = ? WHERE
id = ?",
      [order.Title, order.Quantity, order.Message, order.City, orderId],
      function (err) {
        if (err) {
          return console.log(err.message);
        }
        console.log("Entry updated");
      }
    );
  });
}

async function splitter_function(args) {
  console.log("splitter_function");
  var splitter = args.splitter;
  var splitted_msg = args.message.split(splitter);
  var result = [];
  for (var i = 0; i < splitted_msg.length; i++) {
    result.push(splitted_msg[i]);
  }
  return {
    result: result,
  };
}

```

```

module.exports = {
  GetOrders: GetOrders,

```

```

AddOrder: AddOrder,

GetOrder: GetOrder,

DeleteOrder: DeleteOrder,

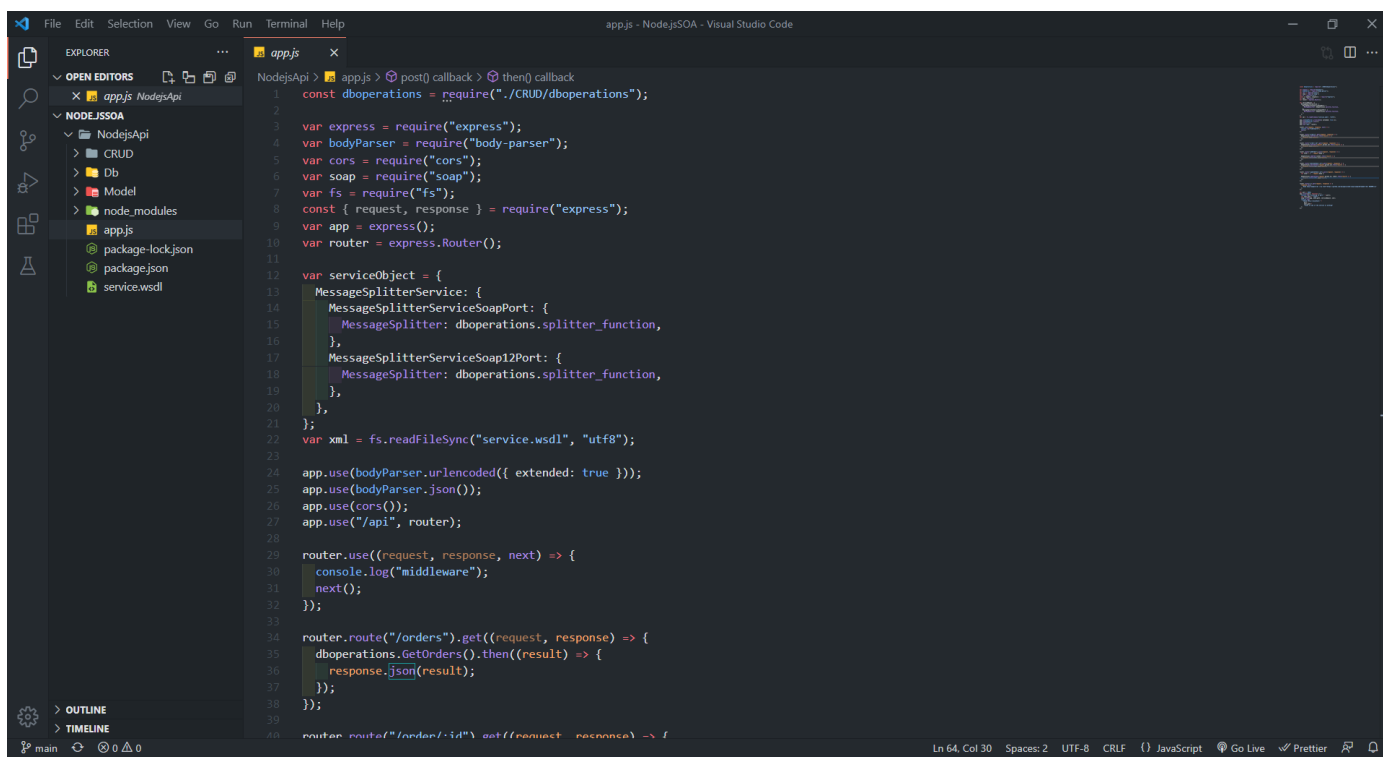
UpdateOrder: UpdateOrder,

splitter_function: splitter_function,

};

```

API VE SOAP/WSDL:



- **App.js:**

```

const dboperations = require("./CRUD/dboperations");

var express = require("express");

var bodyParser = require("body-parser");

var cors = require("cors");

var soap = require("soap");

var fs = require("fs");

const { request, response } = require("express");

var app = express();

```

```

var router = express.Router();

var serviceObject = {
  MessageSplitterService: {
    MessageSplitterServiceSoapPort: {
      MessageSplitter: dboperations.splitter_function,
    },
    MessageSplitterServiceSoap12Port: {
      MessageSplitter: dboperations.splitter_function,
    },
  },
};

var xml = fs.readFileSync("service.wsdl", "utf8");

app.use(bodyParser.urlencoded({ extended: true }));
app.use(bodyParser.json());
app.use(cors());
app.use("/api", router);

router.use((request, response, next) => {
  console.log("middleware");
  next();
});

router.route("/orders").get((request, response) => {
  dboperations.GetOrders().then((result) => {
    response.json(result);
  });
});

router.route("/order/:id").get((request, response) => {

```

```

    dboperations.GetOrder(request.params.id).then((result) => {
        response.json(result);
    });
});
router.route("/addOrder").post((request, response) => {
    let order = { ...request.body };

    dboperations.AddOrder(order).then((result) => {
        response.status(201).json(result);
    });
});
router.route("/deleteOrder/:id").get((request, response) => {
    dboperations.DeleteOrder(request.params.id).then((result) => {
        response.status(200).json(result);
    });
});
router.route("/updateOrder/:id").post((request, response) => {
    let order = { ...request.body };

    dboperations.UpdateOrder(request.params.id, order).then((result) => {
        response.status(200).json(result);
    });
});
router.route("/").get((request, response) => {
    response.send(
        'Node Soap Example!<br /><a href="https://github.com/macogala/node-soap-example#readme">Git README</a>'
    );
});

```

```

var port = 8090;

app.listen(port, function () {

    console.log("Listening on port " + port);

    var wsdl_path = "/wsdl";

    soap.listen(app, wsdl_path, serviceObject, xml);

    console.log(

        "Check http://localhost: " +

        port +

        wsdl_path +

        "?wsdl to see if the service is working"

    );

});

```

Service.wsdl :

This XML file does not appear to have any style information associated with it. The document tree is shown below:

```

<!-- definitions must be the root of the WSDL document -->
<wsdl:definitions xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://tempuri.org/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tm="http://microsoft.com/wsdl/tns" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://tempuri.org/">
  <!-- WSDL TYPES: definition of the data types that are used in the web service -->
  <wsdl:types>
    <xs:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
      <xs:element name="MessageSplitterRequest">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="1" maxOccurs="1" name="message" type="s:string"/>
            <xs:element minOccurs="1" maxOccurs="1" name="splitter" type="s:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="MessageSplitterResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="1" maxOccurs="unbounded" name="result" type="s:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
  </wsdl:types>
  <!-- MESSAGES: defines the data being exchanged between the service and client -->
  <wsdl:message name="MessageSplitterSoapIn">
    <wsdl:part name="parameters" element="tns:MessageSplitterRequest"/>
  </wsdl:message>
  <wsdl:message name="MessageSplitterSoapOut">
    <wsdl:part name="parameters" element="tns:MessageSplitterResponse"/>
  </wsdl:message>
  <!-- PORT TYPES: defines the complete communication operation (one way/round trip) -->
  <wsdl:portType name="MessageSplitterSoapPort">
    <!-- The operation name must be the same as the one specified in the service object -->
    <wsdl:operation name="MessageSplitter">
      <wsdl:input message="tns:MessageSplitterSoapIn"/>
      <wsdl:output message="tns:MessageSplitterSoapOut"/>
    </wsdl:operation>
  </wsdl:portType>
  <!-- BINDINGS: provides details on how a portType operation will actually be transmitted -->
  <wsdl:binding name="MessageSplitterServiceSoapBinding" type="tns:MessageSplitterSoapPort">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="MessageSplitter">
      <soap:operation soapAction="MessageSplitter" style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:binding name="MessageSplitterServiceSoap12Binding" type="tns:MessageSplitterSoapPort">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="MessageSplitter">
      <soap12:operation soapAction="MessageSplitter" style="document"/>
      <wsdl:input>
        <soap12:body use="literal"/>
      </wsdl:input>
      <wsdl:output>

```

ÇALIŞTIRMA

GetOrders:

Postman

GET http://localhost:8090...

POST http://localhost:8090...

POST http://localhost:8090...

GET http://localhost:8090...

GET http://localhost:8090...

+ ...

No Environment

http://localhost:8090/api/orders

Save

</>

GET

http://localhost:8090/api/orders

Send

Params

Authorization

Headers (6)BodyPre-request ScriptTestsSettings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Bulk Edit

BodyCookiesHeaders (7)Test Results

Status: 200 OKTime: 19 msSize: 224 BSave Response

Wireshark

Apply a display filter: <Ctrl-F>

No.	Time	Source	Destination	Protocol	Length	Info
363	144.466219	:::1	:::1	TCP	64	8090 → 62623 [ACK] Seq=1 Ack=2 Win=2160640 Len=0
364	144.469466	:::1	:::1	TCP	64	8090 → 62623 [FIN, ACK] Seq=1 Ack=2 Win=2160640 Len=0
365	144.468562	:::1	:::1	TCP	64	62623 → 8090 [ACK] Seq=2 Ack=2 Win=2160640 Len=0
366	144.468681	:::1	:::1	HTTP	275	GET /api/orders HTTP/1.1
367	144.468614	:::1	:::1	TCP	64	8090 → 62624 [ACK] Seq=1 Ack=212 Win=2160384 Len=0
368	144.471235	:::1	:::1	HTTP	288	HTTP/1.1 200 OK
369	144.471257	:::1	:::1	TCP	64	62624 → 8090 [ACK] Seq=212 Ack=225 Win=2160384 Len=0
370	144.898649	fe80::f895:f2ce:e9a...	ff02::1:3	L2MR	74	Standard query 0x2c9e A wpad
371	144.898788	192.168.1.64	224.0.0.252	L2MR	54	Standard query 0x2c9e A wpad
372	145.215813	192.168.1.64	192.168.1.255	NDIS	82	Name query NB WPA0<00>
373	145.477884	:::1	:::1	TCP	65	[TCP Keep-Alive] 62624 → 8090 [ACK] Seq=211 Ack=225 Win=2160384 Len=1
374	145.477833	:::1	:::1	TCP	76	[TCP Keep-Alive ACK] 8090 → 62624 [ACK] Seq=225 Ack=212 Win=2160384 Len=0 SLE=211 SRE=212

> Frame 366: 275 bytes on wire (2200 bits), 275 bytes captured (2200 bits) on interface \Device\NPF_{...}, id 0

> Null/Loopback

> Internet Protocol Version 6, Src: :::1, Dst: :::1

> Transmission Control Protocol, Src Port: 62624, Dst Port: 8090, Seq: 1, Ack: 1, Len: 211

> Hypertext Transfer Protocol

> GET /api/orders HTTP/1.1\r\nUser-Agent: PostmanRuntime/7.28.4\r\nAccept: */*\r\nPostman-Token: ff0b74ad-2da5-419b-a21f-ab2b0469fd0c\r\nHost: localhost:8090\r\nAccept-Encoding: gzip, deflate, br\r\nConnection: keep-alive\r\n\r\n[Full request URI: http://localhost:8090/api/orders]

[HTTP request 1/1]

[Response in frame 368]

0000 18 00 00 00 00 01 ec 50 00 07 06 00 00 00 00 00 00P.....

0010 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00

0020 00 00 00 00 00 00 00 00 00 00 01 f4 a0 1f 9a

0030 1d 6d d0 97 52 02 92 69 50 18 20 f8 61 58 00 00 m-R-i P aX

0040 47 45 54 20 2f 61 70 69 2f 6f 72 64 65 72 73 20 GET /api/orders

0050 48 54 54 50 2f 31 2e 31 0d 0a 55 73 65 72 2d 41 HTTP/1.1 User-A

0060 67 65 6e 74 3a 20 50 6f 73 74 6d 61 6e 52 75 6e gent: Po steamRun

0070 74 69 6d 65 2f 37 2e 32 38 2e 34 0d 0a 41 63 63 time/7.2 8.4 Acc

0080 65 70 74 3a 20 2a 2f 2a 0d 0a 50 6f 73 74 6d 61 ept: */* Postnea

0090 6e 2d 54 6f 6b 65 6e 3a 20 66 60 30 62 37 34 61 n-Token: ff0b74a

00a0 64 2d 32 64 61 35 2d 34 21 39 62 2d 61 32 31 66 d-2da5-4 19b-a21f

00b0 2d 61 62 32 62 30 34 36 39 66 64 30 63 0d 0a 48 -ab2b046 9fd0c H

Adapter for loopback traffic capture: <live capture in progress>

Packets: 425 · Displayed: 425 (100.0%)

Profile: Default

Çıktı

```
C:\Users\mrtag\AppData\Roam... x + v - □ ×
C:\Users\mrtag\Desktop\Node.jsSOA\NodejsApi>node app.js
Listening on port 8090
Check http://localhost:8090/wsdl?wsdl to see if the service is working
Connected to the chinook database.
middleware
[
  {
    Id: 1,
    Title: 'PostmanTitle2',
    Quantity: 'PostmanQuantity2',
    Message: 'PostmanMessage2',
    City: 'PostmanCity2'
  },
  {
    Id: 2,
    Title: 'kjhaskjd',
    Quantity: 'ajsdslaskj',
    Message: 'dlaksdjla',
    City: 'djalsk'
  },
  {
    Id: 6,
    Title: 'PostmanTitle2',
    Quantity: 'PostmanQuantity2',
    Message: 'PostmanMessage2',
    City: 'PostmanCity2'
  }
]
```

GetOrder:

Postman

GET http://localhost:80... POST http://localhost:8... POST http://localhost:8... GET http://localhost:80... GET http://localhost:80... + ... No Environment

http://localhost:8090/api/order/1 Save Edit </>

GET http://localhost:8090/api/order/1 Send Cookies

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION	... Bulk Edit
Key	Value	Description	

Body Cookies Headers (7) Test Results Status: 200 OK Time: 15 ms Size: 224 B Save Response

Wireshark

Apply a display filter -> <Ctrl-F>

No.	Time	Source	Destination	Protocol	Length	Info
11	2.968516	:::1	:::1	TCP	76	62734 -> 8090 [SYN] Seq=0 Win=65535 Len=0 MSS=65475 WS=256 SACK_PERM=1
12	2.968584	:::1	:::1	TCP	76	8090 -> 62734 [SYN, ACK] Seq=0 Ack=1 Min=65535 Len=0 MSS=65475 WS=256 SACK_PERM=1
13	2.968611	:::1	:::1	TCP	64	62734 -> 8090 [ACK] Seq=1 Ack=1 Win=327168 Len=0
14	2.969416	:::1	:::1	TCP	76	62735 -> 8090 [SYN] Seq=0 Win=65535 Len=0 MSS=65475 WS=256 SACK_PERM=1
15	2.969451	:::1	:::1	TCP	76	8090 -> 62735 [SYN, ACK] Seq=0 Ack=1 Min=65535 Len=0 MSS=65475 WS=256 SACK_PERM=1
16	2.969467	:::1	:::1	TCP	64	62735 -> 8090 [ACK] Seq=1 Ack=1 Win=327168 Len=0
17	2.969515	:::1	:::1	TCP	64	62734 -> 8090 [FIN, ACK] Seq=1 Ack=1 Win=327168 Len=0
18	2.969524	:::1	:::1	TCP	64	8090 -> 62734 [ACK] Seq=1 Ack=2 Win=2160640 Len=0
19	2.969600	:::1	:::1	HTTP	276	GET /api/order/1 HTTP/1.1
20	2.969695	:::1	:::1	TCP	64	8090 -> 62735 [ACK] Seq=1 Ack=213 Win=2160384 Len=0
21	2.971139	:::1	:::1	TCP	64	8090 -> 62734 [FIN, ACK] Seq=1 Ack=2 Win=2160640 Len=0
22	2.971155	:::1	:::1	TCP	64	62734 -> 8090 [ACK] Seq=1 Ack=2 Win=327168 Len=0
23	2.979473	:::1	:::1	HTTP	288	HTTP/1.1 200 OK
24	2.979494	:::1	:::1	TCP	64	62735 -> 8090 [ACK] Seq=213 Ack=225 Win=326912 Len=0
25	3.997892	:::1	:::1	TCP	65	[TCP Keep-Alive] 62735 -> 8090 [ACK] Seq=212 Ack=225 Win=326912 Len=1

> Frame 19: 276 bytes on wire (2208 bits), 276 bytes captured (2208 bits) on interface \Device\NPF_{...}_Loopback, id 0

> Null/Loopback

> Internet Protocol Version 6, Src: ::1, Dst: ::1

> Transmission Control Protocol, Src Port: 62735, Dst Port: 8090, Seq: 1, Ack: 1, Len: 212

> Hypertext Transfer Protocol

> GET /api/order/1 HTTP/1.1\r\nUser-Agent: PostmanRuntime/7.28.4\r\nAccept: */*\r\nPostman-Token: 687b311f-e28c-4617-bbe8-4b7c6fb8f44f\r\nHost: localhost:8090\r\nAccept-Encoding: gzip, deflate, br\r\nConnection: keep-alive\r\n\r\n[Full request URI: http://localhost:8090/api/order/1][HTTP request 1/1][Response in frame: 23]

0000 18 00 00 00 60 08 83 54 00 e8 06 80 00 00 00 00T.....

0010 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00

0020 00 00 00 00 00 00 00 00 00 00 00 01 f5 0f 1f 9a

0030 00 75 8a d1 22 62 e4 ee 50 18 04 fe 94 70 00 00 :g.*D..P.....

0040 47 45 54 20 2f 61 70 69 2f 6f 72 64 65 72 2f 31 GET /api /order/1

0050 20 48 54 54 50 2f 31 2e 31 0d 0a 55 73 65 72 2d HTTP/1.1:User-

0060 41 67 65 6e 74 3a 20 50 6f 73 74 6d 61 6e 52 75 Agent: PostmanRu

0070 6e 74 69 6d 65 2f 37 2e 32 38 2e 34 0d 0a 41 63 ntime/7. 28.4:Ac

0080 65 65 70 74 3a 20 2a 2f 2a 0d 0a 50 6f 73 74 6d cept: */ *.Postm

0090 61 6e 2d 54 6f 6b 65 6e 3a 20 36 30 37 62 33 31 an-Token : 687b31

00a0 31 66 2d 65 32 30 63 2d 34 36 31 37 2d 62 62 65 1f-e28c- 4617-bbe

00b0 38 2d 34 62 37 63 36 66 62 38 66 64 34 66 0d 0a 8-4b7c6f b8f44f...

Adapter for loopback traffic capture: <live capture in progress>

Packets: 74 · Displayed: 74 (100.0%)

Profile: Default

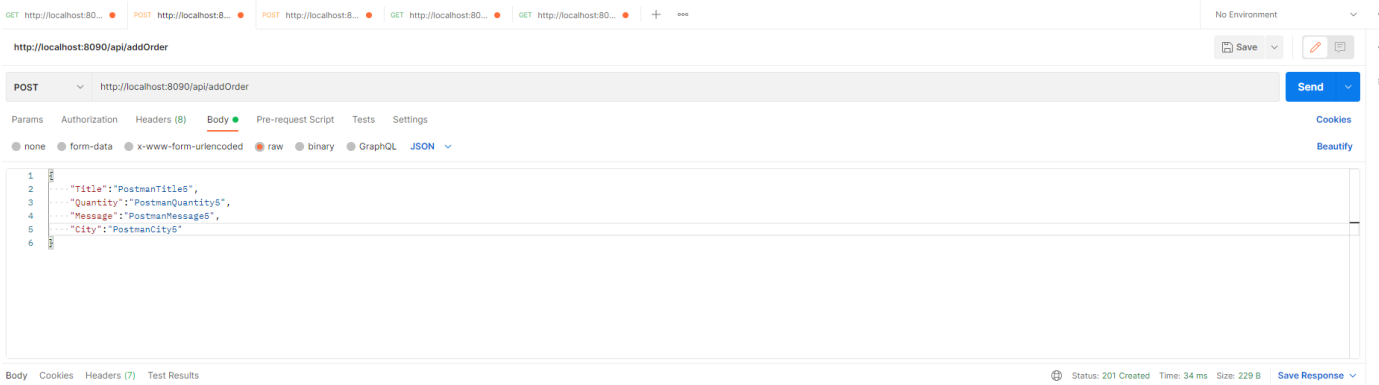
Çıktı

```
C:\Users\mrtag\AppData\Roami x + v - □ ×

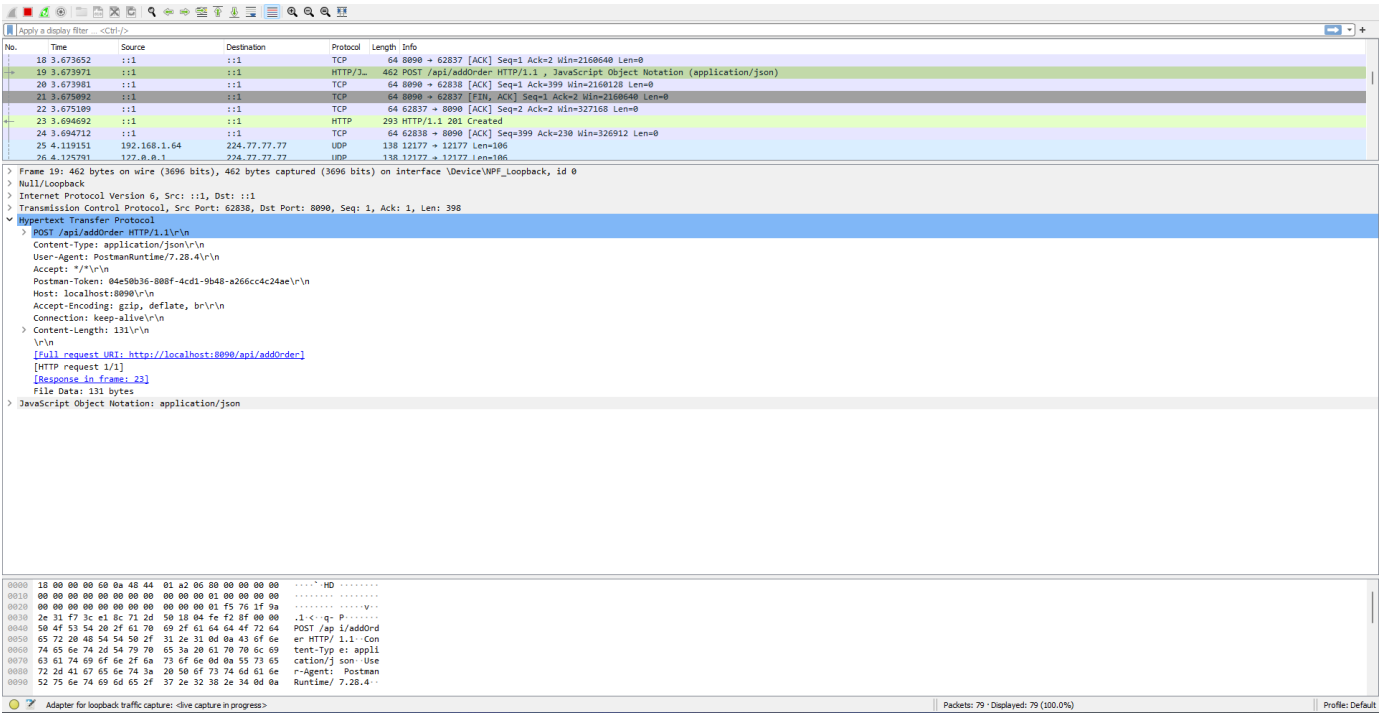
C:\Users\mrtag\Desktop\Node.jsSOA\NodejsApi>node app.js
Listening on port 8090
Check http://localhost:8090/wsdl?wsdl to see if the service is working
Connected to the chinook database.
middleware
{
  Id: 1,
  Title: 'PostmanTitle2',
  Quantity: 'PostmanQuantity2',
  Message: 'PostmanMessage2',
  City: 'PostmanCity2'
}
```

AddOrder:

Postman



Wireshark



Çıktı

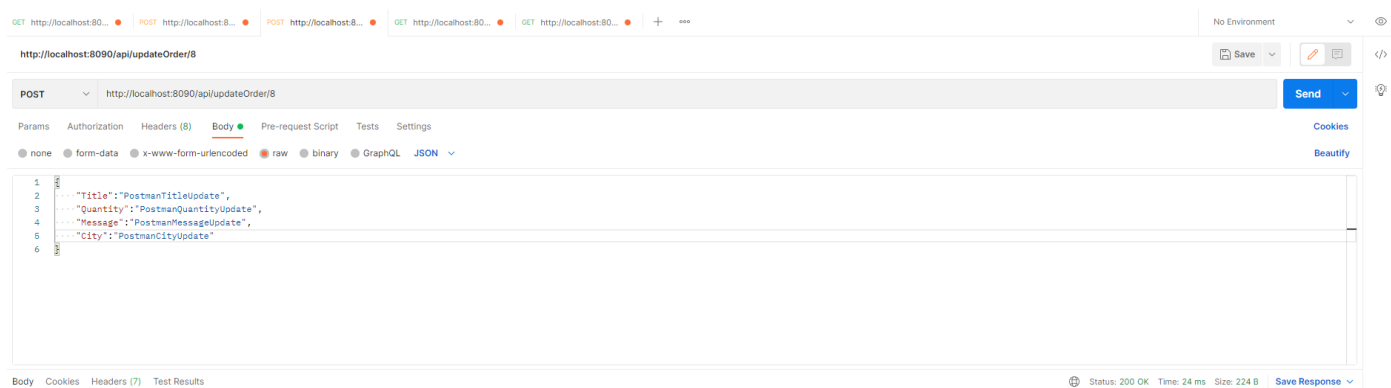
```
C:\Users\mrtag\AppData\Roam\ > + - □ ×
```

```
C:\Users\mrtag\Desktop\Node.jsSOA\NodejsApi>node app.js
Listening on port 8090
Check http://localhost:8090/wsdl?wsdl to see if the service is working
Connected to the chinook database.
middleware
Entry added
```

	Id		Title	Quantity	Message	City
	Filtre	Filtre		Filtre	Filtre	Filtre
1	1	PostmanTitle2		PostmanQuantity2	PostmanMessage2	PostmanCity2
2	2	kjhas kjd		ajsd laskj	dlaksdjla	djalsk
3	6	PostmanTitle2		PostmanQuantity2	PostmanMessage2	PostmanCity2
4	8	PostmanTitle5		PostmanQuantity5	PostmanMessage5	PostmanCity5

UpdateOrder:

Postman



Wireshark

Apply a display filter<Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
11	2.6655802	:::1	:::1	TCP	76	62927 → 8090 [SYN] Seq=0 Win=65535 Len=0 MSS=65475 WS=256 SACK_PERM=1
12	2.6655832	:::1	:::1	TCP	76	8090 → 62927 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65475 WS=256 SACK_PERM=1
13	2.6655544	:::1	:::1	TCP	64	62927 → 8090 [ACK] Seq=1 Ack=1 Win=327168 Len=0
14	2.6655592	:::1	:::1	TCP	64	62926 → 8090 [FIN, ACK] Seq=1 Ack=1 Win=327168 Len=0
15	2.6656000	:::1	:::1	TCP	64	8090 → 62926 [ACK] Seq=1 Ack=2 Win=2160640 Len=0
16	2.6656812	:::1	:::1	HTTP/2	487	POST /api/updateOrder/8 HTTP/1.1, JavaScript Object Notation (application/json)
17	2.6658022	:::1	:::1	TCP	64	8090 → 62927 [ACK] Seq=1 Ack=424 Win=2160128 Len=0
18	2.667389	:::1	:::1	TCP	64	8090 → 62926 [FIN, ACK] Seq=1 Ack=2 Win=2160640 Len=0
19	2.667325	:::1	:::1	TCP	64	62926 → 8090 [ACK] Seq=2 Ack=2 Win=327168 Len=0
20	2.684597	:::1	:::1	HTTP	288	HTTP/1.1 200 OK
21	2.684622	:::1	:::1	TCP	64	62927 → 8090 [ACK] Seq=424 Ack=225 Win=326912 Len=0
22	2.736484	127.0.0.1	127.0.0.1	TCP	52	17945 → 52825 [PSH, ACK] Seq=281 Ack=1 Win=8440 Len=8
23	2.736498	127.0.0.1	127.0.0.1	TCP	44	52825 → 17945 [ACK] Seq=1 Ack=289 Win=480 Len=0
24	2.736589	127.0.0.1	127.0.0.1	TCP	348	17945 → 52825 [PSH, ACK] Seq=289 Ack=1 Win=8440 Len=304

> Frame 16: 487 bytes on wire (3896 bits), 487 bytes captured (3896 bits) on interface \Device\NPF_{...}_id 0

> Null/Loopback

> Internet Protocol Version 6, Src: ::1, Dst: ::1

> Transmission Control Protocol, Src Port: 62927, Dst Port: 8090, Seq: 1, Ack: 1, Len: 423

> Hypertext Transfer Protocol

> POST /api/updateOrder/8 HTTP/1.1\r\nContent-Type: application/json\r\nUser-Agent: PostmanRuntime/7.28.4\r\nAccept: */*\r\nPostman-Token: a646a7d-e65a-410c-bb11-5eb0f35eeda3\r\nHost: localhost:8090\r\nAccept-Encoding: gzip, deflate, br\r\nConnection: keep-alive\r\nContent-Length: 151\r\n\r\n[Full request URI: http://localhost:8090/api/updateOrder/8][HTTP request 1/1][Response in frame 20]

> JavaScript Object Notation: application/json

> Object

> Member Key: Title

> Member Key: Quantity

> Member Key: Message

> Member Key: City

0000 18 00 00 00 60 08 40 0e 01 bb 06 80 00 00 00 00@.....

0010 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00

0020 00 00 00 00 00 00 00 00 00 00 01 f5 cf 1f 9a

0030 6a de a2 02 a5 a4 aa 2c 50 18 04 fe 92 0b 00 00j.....P.....

0040 50 4f 53 54 20 2f 61 70 69 2f 75 70 64 61 74 65 POST /ap i/update

0050 4f 72 64 65 72 2f 38 20 48 54 54 50 2f 31 2e 31 Order/8 HTTP/1.1

0060 0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20 ..Conten t-type:

0070 61 70 70 6c 69 63 61 74 69 6f 6e 2f 6a 73 6f 6e applicat ion/json

0080 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 50 6f ..User-a gent: Po

0090 73 74 6d 61 6e 52 75 6e 74 69 6d 65 2f 37 2e 32 stmanRun time/7.2

Adapter for loopback traffic capture: <live capture in progress>

Packets: 78 / Displayed: 78 (100.0%)

Profile: Default

Çıktı

C:\Users\mrtag\AppData\Roami

C:\Users\mrtag\Desktop\Node.jsSOA\NodejsApi>node app.js

Listening on port 8090

Check http://localhost:8090/wsdl?wsdl to see if the service is working

Connected to the chinook database.

middleware

Entry updated

Table: Orders

	ID	Title	Quantity	Message	City
1	1	PostmanTitle2	PostmanQuantity2	PostmanMessage2	PostmanCity2
2	1	kjhaskjd	ajsdaskj	diaksdjla	djalsk
3	6	PostmanTitle2	PostmanQuantity2	PostmanMessage2	PostmanCity2
4	8	PostmanTitleUpdate	PostmanQuantityUpdate	PostmanMessageUpdate	PostmanCityUpdate

Postman

GET http://localhost:80...

POST http://localhost:80...

POST http://localhost:80...

GET http://localhost:80...

GET http://localhost:80...

+ ...

No Environment

Save

Send

GET http://localhost:8090/api/deleteOrder/8

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTestsSettings

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

BodyCookiesHeaders (7)Test Results

Status: 200 OKTime: 19 msSize: 224 BSave Response

Wireshark

Wireshark Network Traffic Analysis

Packet List

No.	Time	Source	Destination	Protocol	Length	Info
19	2.791512	:::	:::	TCP	64	53142 → 8090 [ACK] Seq=1 Ack=1 Win=327168 Len=0
19	2.791560	:::	:::	TCP	64	53141 → 8090 [FIN, ACK] Seq=1 Ack=1 Win=327168 Len=0
20	2.791569	:::	:::	TCP	64	8090 → 53141 [ACK] Seq=1 Ack=2 Win=2160640 Len=0
21	2.791893	:::	:::	HTTP	282	GET /api/deleteOrder/8 HTTP/1.1
22	2.791904	:::	:::	TCP	64	8090 → 53142 [ACK] Seq=1 Ack=219 Win=2160384 Len=0
23	2.792653	:::	:::	TCP	64	8090 → 53141 [FIN, ACK] Seq=1 Ack=2 Win=2160640 Len=0
24	2.792649	:::	:::	TCP	64	53141 → 8090 [ACK] Seq=2 Ack=2 Win=327168 Len=0
25	2.800714	:::	:::	HTTP	288	HTTP/1.1 200 OK
26	2.800734	:::	:::	TCP	64	53142 → 8090 [ACK] Seq=219 Ack=225 Win=326912 Len=0
27	2.832171	192.168.1.64	224.77.77.77	UDP	138	12177 → 12177 Len=186
28	2.832928	127.0.0.1	224.77.77.77	UDP	138	12177 → 12177 Len=186
29	3.820810	:::	:::	TCP	65	[TCP Keep-Alive] 53142 → 8090 [ACK] Seq=218 Ack=225 Win=326912 Len=1
30	3.820822	:::	:::	TCP	76	[TCP Keep-Alive ACK] 8090 → 53142 [ACK] Seq=225 Ack=219 Win=2160384 Len=0 SLE=218 SRE=219
31	4.140805	127.0.0.1	127.0.0.1	TCP	52	17945 → 52825 [PSH, ACK] Seq=503 Ack=1 Win=8448 Len=0

Packet Details

- Frame 21: 282 bytes on wire (2256 bits), 282 bytes captured (2256 bits) on interface DeviceWPF_Loopback, id 0
 - Null Loopback
 - Internet Protocol Version 6, Src: :::, Dst: :::
 - Transmission Control Protocol, Src Port: 53142, Dst Port: 8090, Seq: 1, Ack: 1, Len: 218
 - Hypertext Transfer Protocol
 - GET /api/deleteOrder/8 HTTP/1.1\r\n
 - [Expert Info (Chat/Sequence): GET /api/deleteOrder/8 HTTP/1.1\r\n]
 - Request Method: GET
 - Request URI: /api/deleteOrder/8
 - Request Version: HTTP/1.1
 - User-Agent: PostmanRuntime/7.28.4\r\n
 - Accept: */*\r\n
 - Postman-Token: 416799ca-3c5a-4863-a2ba-6bf64f64dc5\r\n
 - Host: localhost:8090\r\n
 - Accept-Encoding: gzip, deflate, br\r\n
 - Connection: keep-alive\r\n
 - \r\n
 - [Full request URI: http://localhost:8090/api/deleteOrder/8]
 - [HTTP request 1/1]
 - [Response in frame 25]

Packet Bytes

```

0000  18 00 00 00 60 0a d5 5b 00 ee 06 00 00 00 00 00 .....[ .....
0010  00 00 00 00 00 00 00 00 00 00 00 00 01 cf 96 1f 5a .....
0020  00 00 00 00 00 00 00 00 00 00 00 00 e1 cf 96 1f 5a .....
0030  bcd 71 bd 41 c1 cf 2f 50 18 04 fa fe 1e 25 00 00 ..<D>>P~-&
0040  47 45 54 20 2f 61 70 69 2f 64 65 6c 65 74 65 4f GET /api /deleteO
0050  72 64 65 72 2f 30 20 40 54 54 50 2f 31 2e 31 0d rder/8 H TTP/1.1.
0060  0a 55 73 65 72 2d 41 67 65 64 74 3a 20 50 0f 73 User-Ag ent: Pos
0070  74 dd 61 6a 65 75 6e 74 69 6d 65 2f 37 2e 32 38 tmanRunT ime/7.28
0080  2e 34 0d 0a 41 63 63 65 70 74 3a 20 2a 2f 2a 0d .4 -Acce pt: /*
0090  0a 50 6f 73 74 6d 61 6e 2d 54 6f 6b 65 6e 6a 20 *Postman -Token:
    
```

Çıktı

The terminal window shows the following output:

```
C:\Users\mrtag\Desktop\Node.jsSOA\NodejsApi>node app.js
Listening on port 8090
Check http://localhost:8090/wsdl?wsdl to see if the service is working
Connected to the chinook database.
middleware
Entry deleted
```

The web browser shows the results of a GET request to `http://localhost:8090/wsdl?wsdl`. The response is an XML document with the following structure:

Id	Title	Quantity	Message	City
1	PostmanTitle2	PostmanQuantity2	PostmanMessage2	PostmanCity2
2	kjhasjkd	ajsdiasjk	diasdjlja	djalsk
3	6 PostmanTitle2	PostmanQuantity2	PostmanMessage2	PostmanCity2

SOAP/WSDL:

Postman

The screenshot shows the Postman application interface. The request is a GET request to `http://localhost:8090/wsdl?wsdl`. The response is an XML document with the following structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- <definitions> must be the root of the WSDL document -->
<wsdl:definitions targetNamespace="http://tempuri.org/"
  xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:tns="http://tempuri.org/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <!-- WSDL TYPES: definition of the data types that are used in the web service -->
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
      <s:element name="MessageSplitterRequest">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="message" type="s:string"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </wsdl:types>
</wsdl:definitions>
```


Wireshark

The image shows a Wireshark packet capture of an HTTP GET request and response. The packet list on the left shows a GET request from 192.168.1.64 to 224.77.77.77. The packet details pane shows the request structure, including the URI path, query, and headers. The packet bytes pane shows the raw data of the request.

Frame 19: 274 bytes on wire (2192 bits), 274 bytes captured (2192 bits) on interface \Device\NPF_{...} id 0

Null/Loopback

Internet Protocol Version 6, Src: ::1, Dst: ::1

Transmission Control Protocol, Src Port: 53186, Dst Port: 8090, Seq: 1, Ack: 1, Len: 210

Hypertext Transfer Protocol

GET /wsdl?wsdl HTTP/1.1

[Expert Info (Chat/Sequence): GET /wsdl?wsdl HTTP/1.1]

Request Method: GET

Request URI: /wsdl?wsdl

Request URI Path: /wsdl

Request URI Query: wsdl

Request Version: HTTP/1.1

User-Agent: PostmanRuntime/7.28.4

Accept: */*

Postman-Token: ead5206d-45d6-4480-9dca-c13f74487644

Host: localhost:8090

Accept-Encoding: gzip, deflate, br

Connection: keep-alive

[Full request URI: http://localhost:8090/wsdl?wsdl]

[HTTP request 1/1]

[Response in frame 23]

Çıktı

The image shows a web browser displaying an XML file. The XML content is a WSDL document for a service named "MessageSplitter". The document defines the service's interface, including the "MessageSplitterRequest" and "MessageSplitterResponse" messages, and the "MessageSplitter" port type. The XML is displayed in a monospace font, with the root element being the "definitions" tag.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!-- definitions must be the root of the WSDL document -->
<?xml:definitions xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://tempuri.org/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tm="http://microsoft.com/wsdl/mime/textmatching/" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://tempuri.org/"
  <!-- WSDL TYPES: Definition of the data types that are used in the web service -->
  <wsdl:types>
    <xs:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
      <xs:element name="MessageSplitterRequest">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="1" maxOccurs="1" name="message" type="s:string"/>
            <xs:element minOccurs="1" maxOccurs="1" name="splitter" type="s:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="MessageSplitterResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="1" maxOccurs="unbounded" name="result" type="s:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
  </wsdl:types>
  <!-- MESSAGES: defines the data being exchanged between the service and client -->
  <wsdl:message name="MessageSplitterSoapIn">
    <wsdl:part name="parameters" element="tns:MessageSplitterRequest"/>
  </wsdl:message>
  <wsdl:message name="MessageSplitterSoapOut">
    <wsdl:part name="parameters" element="tns:MessageSplitterResponse"/>
  </wsdl:message>
  <!-- PORT TYPES: defines the complete communication operation (one way/round trip) -->
  <wsdl:portType name="MessageSplitterSoapPort">
    <!-- The operation name must be the same as the one specified in the service object -->
    <wsdl:operation name="MessageSplitter">
      <wsdl:input message="tns:MessageSplitterSoapIn"/>
      <wsdl:output message="tns:MessageSplitterSoapOut"/>
    </wsdl:operation>
  </wsdl:portType>
  <!-- BINDING: provides details on how a portType operation will actually be transmitted -->
  <wsdl:binding name="MessageSplitterServiceSoapBinding" type="tns:MessageSplitterSoapPort">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="MessageSplitter">
      <soap:operation soapAction="MessageSplitter" style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:binding name="MessageSplitterServiceSoap12Binding" type="tns:MessageSplitterSoapPort">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="MessageSplitter">
      <soap12:operation soapAction="MessageSplitter" style="document"/>
      <wsdl:input>
        <soap12:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
```