Enabling Embedded Systems to access Internet Resources
Enabling Embedded Systems to access Internet Resources
Embedded Internet Book

The Embedded Internet (with CD)
TCP/IP Basics, Implementation and Applications
Sergio Scaglia

Feb 2007, Paperback, 632 pages

ISBN13: 9780321306388
ISBN10: 0321306384

www.EmbeddedInternet.org
Agenda

- Enabling Embedded Systems to access Internet Resources: RATIONALE
- Web Services: INTRODUCTION
- HTTP Protocol: REVIEW
- HTTP Protocol Bindings
- Testing a Web Service
- Application Case: Using Web Services for DNS Resolution
- Implementing and Testing the Web Service
- Implementing and Testing the Embedded Application
Enabling Embedded Systems to access Internet Resources

RATIONALE

- Embedded Systems have limited resources compared with PCs.
- Some applications may require large memory space and high processing power.
Enabling Embedded Systems to access Internet Resources

ALTERNATIVE SOLUTION

- Adding a TCP/IP Stack, Embedded Systems will have internet-connectivity which allows them access Internet Resources.
In this scenario, complex tasks could be resolved remotely in external servers.
Enabling Embedded Systems to access Internet Resources

ALTERNATIVE SOLUTION

- In other words, Embedded Systems could greatly benefit from the “external intelligence” provided by Desktop servers connected to the Internet, without the need of additional resources.
Enabling Embedded Systems to access Internet Resources

IMPLEMENTATION

- A Remote Procedure could be called in order to resolve complex tasks, such as complex algorithms, CPU-intensive calculations, or retrieve records from a Database.

Ok. But how to implement this? Using Web Services!
Web Services
Introduction
What are Web Services?

- They are a Standardized way to call a remote procedure over the Internet.
- They allow a distributing computing schema to work independently from the technology, language, and device.
- Servers expose a piece of functionality through a Web Interface.
- Clients consume this functionality from its application, using standard Internet protocols (HTTP, SOAP, XML).
- HTTP is used as the transport protocol, to move messages between Clients and Servers - for secure transmissions, HTTPS can be used - (Advantage: most firewalls allow HTTP traffic).
- These messages formats are defined according to the SOAP protocol. The SOAP messages are encoded using XML.
Enabling Embedded Systems to access Internet Resources

WEB SERVICES – Typical Scenario
Enabling Embedded Systems to access Internet Resources
WEB SERVICES – Standards

- **UDDI (Universal Description, Discovery, and Integration):** The UDDI database is a central repository of available Web Services. Developers can access the UDDI registry to search for a Web Service functionality.

- **DISCO (Discovery Protocol):** It allows dynamic discovery of all Web Services located on a particular web site.

- **WSDL (Web Service Description Language):** It allows specifying into a WSDL document, each method of a Web Service and the parameters it accepts and returns. That is, it specifies the interface of the Web Services. This document is considered like a contract that specifies the SOAP messages to send to the Web Service and the messages to expect in return.
Enabling Embedded Systems to access Internet Resources
WEB SERVICES – Standards (Cont.)
Enabling Embedded Systems to access Internet Resources
WEB SERVICES – Transport Protocol: HTTP
HTTP Protocol: Review
Enabling Embedded Systems to access Internet Resources

HTTP session example
Enabling Embedded Systems to access Internet Resources
HTTP Messages Format
Enabling Embedded Systems to access Internet Resources
HTTP Messages Examples

Request:

GET /book/webservices/DnsService.asmx/DnsResolve?dn=string HTTP/1.1
Host: localhost

Response:

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://EmbeddedInternet.org/Book/WebServices/DnsService">string</string>
HTTP Protocol Bindings
Enabling Embedded Systems to access Internet Resources
WEB SERVICES – Protocol Bindings

- SOAP (versions 1.1 and 1.2)
- HTTP POST
- HTTP GET (simplest - recommended for Embedded systems)
Enabling Embedded Systems to access Internet Resources

WEB SERVICES – SOAP 1.1 Binding

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```xml
POST /book/webservices/DnsService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://EmbeddedInternet.org/Book/WebServices/DnsService/DnsResolve"

<?xml version="1.0" encoding="utf-8"?>
<soap:Body>
  <DnsResolve xmlns="http://EmbeddedInternet.org/Book/WebServices/DnsService">
    <dns:string>dns</dns:string>
  </DnsResolve>
</soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Body>
  <DnsResolveResponse xmlns="http://EmbeddedInternet.org/Book/WebServices/DnsService">
    <DnsResolveResult><dns:string>dns</dns:string></DnsResolveResult>
  </DnsResolveResponse>
</soap:Body>
</soap:Envelope>
```
Enabling Embedded Systems to access Internet Resources
WEB SERVICES – SOAP 1.2 Binding

SOAP 1.2
The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

POST /book/webservices/DnsService.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
  <soap12:Body>
    <DnsResolve xmlns="http://EmbeddedInternet.org/Book/WebServices/DnsService">
      <dns:string></dns:string>
    </DnsResolve>
  </soap12:Body>
</soap12:Envelope>

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
  <soap12:Body>
    <DnsResolveResponse xmlns="http://EmbeddedInternet.org/Book/WebServices/DnsService">
      <DnsResolveResult>string</DnsResolveResult>
    </DnsResolveResponse>
  </soap12:Body>
</soap12:Envelope>
Enabling Embedded Systems to access Internet Resources
WEB SERVICES – POST Binding

HTTP POST

The following is a sample HTTP POST request and response. The placeholders shown need to be replaced with actual values.

```
POST /book/webservices/DnsService.asmx/DnsResolve HTTP/1.1
Host: localhost
Content-Type: application/x-www-form-urlencoded
Content-Length: length

dn=string

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://EmbeddedInternet.org/Book/WebServices/DnsService">string</string>
```
Enabling Embedded Systems to access Internet Resources
WEB SERVICES – GET Binding

HTTP GET

The following is a sample HTTP GET request and response. The placeholders shown need to be replaced with actual values.

```
GET /book/webservices/DnsService.asmx/DnsResolve?dn=string HTTP/1.1
Host: localhost

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://EmbeddedInternet.org/Book/WebServices/DnsService">string</string>
```
Enabling Embedded Systems to access Internet Resources

WEB SERVICES – Binding Configurations (web.config)

```xml
<configuration>
    ............
    <system.web>
        ............
        <webServices>
            <protocols>
                <clear/>
                <add name="HttpGet"/>
                <add name="HttpPost"/>
                <add name="HttpSoap"/>
                <add name="HttpSoap12"/>
                <add name="Documentation"/>
            </protocols>
        </webServices>
    </system.web>
</configuration>
```
Testing a Web Service
Enabling Embedded Systems to access Internet Resources
WEB SERVICES – Testing the Web Service (1/3)

The following operations are supported. For a formal definition, please review the Service Description.

- Add
- Divide
- Multiply
- Subtract
Enabling Embedded Systems to access Internet Resources

WEB SERVICES – Testing the Web Service (2/3)

Click [here](http://www.embeddedinternet.org/Book/WebServices/MathService.asmx?op=Add) for a complete list of operations.

**Add**

**Test**

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>a:</td>
<td></td>
</tr>
<tr>
<td>b:</td>
<td></td>
</tr>
</tbody>
</table>

[Invoke]
Enabling Embedded Systems to access Internet Resources
WEB SERVICES – Testing the Web Service (3/3)

```xml
<?xml version="1.0" encoding="utf-8" ?>
<float xmlns="http://EmbeddedInternet.org/Book/WebServices/MathService">3</float>
```
Using the Web Service through the HTTP URL syntax

http://www.embeddedinternet.org/Book/WebServices/MathService.asmx/Add?a=1&b=2
Application Case: Using Web Services for DNS Resolution
Enabling Embedded Systems to access Internet Resources
Implementing the “DnsService” Web Service

```csharp
<%@ WebService Language="C#" Class="DnsService" %>
using System;
using System.Web.Services;
using System.Net;

[WebService(Namespace = "http://EmbeddedInternet.org/Book/WebServices/DnsService")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class DnsService {

    [WebMethod]
    public string DnsResolve(string dn) {
        try {
            IPEndPoint iphost = Dns.GetHostEntry(dn);
            return iphost.AddressList[0].ToString();
        } catch (Exception ex) {
            return ex.Message;
        }
    }
}
```
Enabling Embedded Systems to access Internet Resources
Testing the “DnsService” Web Service

Enabling Embedded Systems to access Internet Resources
Consuming the “DnsService” Web Service from an Embedded System

- Create a TCP socket and open a connection to Port 80 (IP address of the server where the Web Service is published)
- Once the connection is established, create and send the Web Service Request
- When the Response is received, process the XML and extract the result
- Close the connection and the TCP socket
Enabling Embedded Systems to access Internet Resources
Testing the Embedded Application - Web Service Request

Lab 13: Consuming Web Services
Press h to see the Console Commands help...
Input the domain name (press Enter to end): www.hotmail.com
Resolving...
TCP Socket 1 created, port 1025
Connection Established with IP: 65.182.101.227 - Port: 80
Web Service Request:
GET /Book/WebServices/DnsService.asmx/DnsResolve?dn=www.hotmail.com
Host: www.embeddedinternet.org

The WebService Request was sent!
Enabling Embedded Systems to access Internet Resources
Testing the Embedded Application - Web Service Response

Event: Data Available from IP: 65.182.101.227 - Port: 80

Web Service Response:
HTTP/1.1 200 OK
Date: Thu, 23 Mar 2006 19:01:38 GMT
Server: Microsoft-IIS/6.0
Set-Cookie: n/a
MicrosoftOfficeWebServer: 5.0_Pub
X-AspNet-Version: 2.0.50727
Cache-Control: private, max-age=0
Content-Type: text/xml; charset=utf-8
Content-Length: 132

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://EmbeddedInternet.org/Book/WebServices/DnsService">64.70.45.46</string>

IP address: 64.70.45.46
Socket 1 closed