COMMON Europe
Enterprise Modernization Tour

Open your World Wide Windows and Supply Real-Time Information To Your Applications

About The Speaker

With an IT career spanning over 30 years, Charles Guarino has been a consultant for most of them. Since 1995 he has been founder and President of Central Park Data Systems, Inc., a New York area based IBM midrange consulting company. In addition to being a professional speaker, he is a frequent contributor of technical and strategic articles and webcasts for the IT community. He is a proud member of COMMON's Speaker Excellence Hall of Fame and also Long Island Software and Technology Network's Twenty Top Techies of 2009. Charles currently serves as a member of COMMON's Strategic Education Team (SET) and is also Immediate Past President and monthly Q&A host of LISUG, a Long Island IBM i User's Group www.lisug.org.

Charles can be reached at cguarino@centralparkdata.com.
LinkedIn - http://www.linkedin.com/in/guarinocharles
Twitter - @charlieguarino

"640 KB Ought To Be Enough For Anybody"

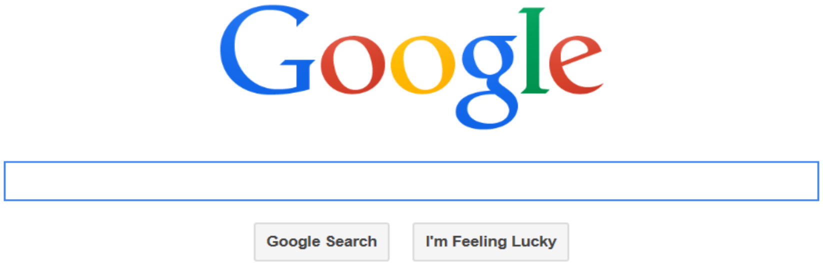
- Bill Gates, 1981

"… the value of the Internet for communications in general isn't very high… I think it's grossly over-promoted and there's a great deal of hyperbole surrounding it."

-Clifford Stohl, 1995
Scientist and author

April 1998

$100,000



Private / Public / Hybrid Cloud

SaaS

PaaS

IaaS

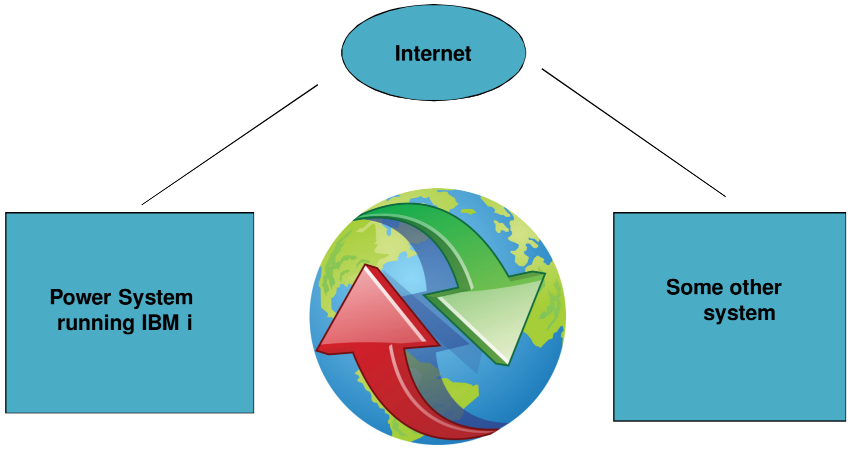
Cloud Storage / Backup

Web Services or APIs

What We’ll Cover …

- Web Services 101
- Consuming a REST Web Service
- Consuming a SOAP Web Service
- Providing a SOAP Web Service
- Wrap-up

WEB SERVICES



Internet

Power System running IBM i

Some other system

REST

(Representational State Transfer)

URI http://.....

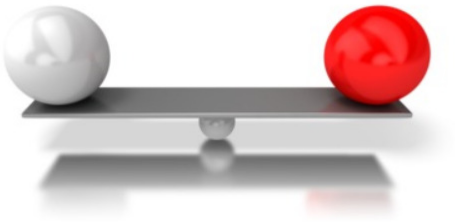
SOAP

(Simple Object Access Protocol)

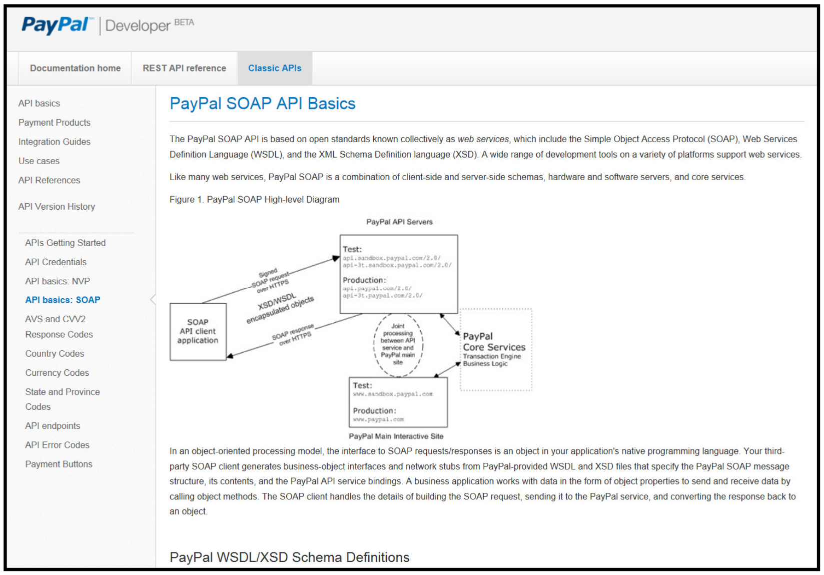
WSDL and messages

Web Service (or API) =

ILE Service Program



https://developer.paypal.com/docs/classic/api/PayPalSOAPAPIArchitecture/



PayPal SOAP API Basics

The PayPal SOAP API is based on open standards known collectively as *web services*, which include the Simple Object Access Protocol (SOAP), Web Services Definition Language (WSDL), and the XML Schema Definition language (XSD). A wide range of development tools on a variety of platforms support web services.

Like many web services, PayPal SOAP is a combination of client-side and server-side schemas, hardware and software servers, and core services.

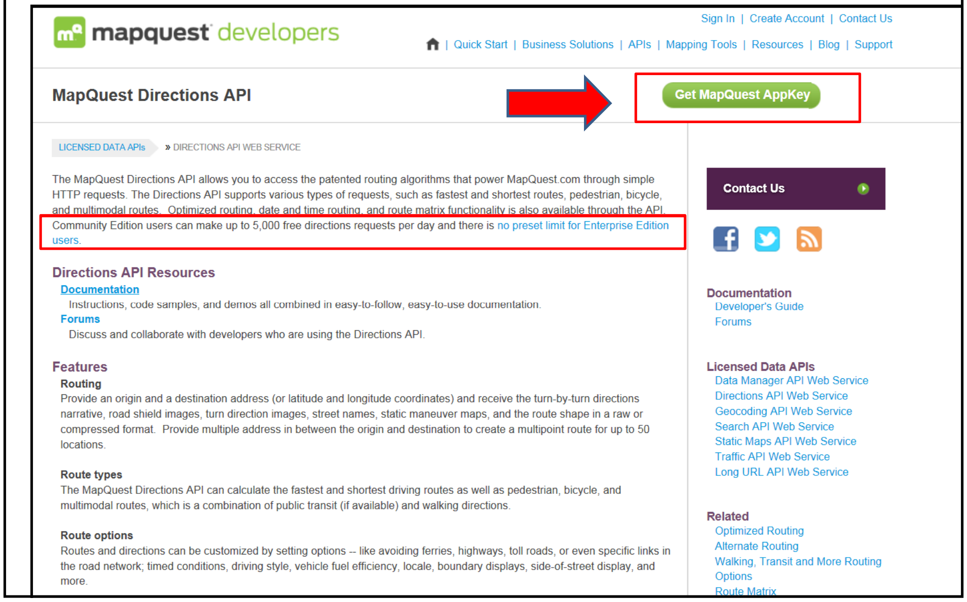
Figure 1. PayPal SOAP High-level Diagram

PayPal API Servers
Test:
api.sandbox.paypal.com/2.0/
api-3t.sandbox.paypal.com/2.0/
Production:
api.paypal.com/2.0/
api-3t.paypal.com/2.0/
Signed SOAP request over HTTPS
XSD/WSDL encapsulated objects
SOAP response over HTTPS
SOAP API client application
Joint processing between API service and PayPal main site
PayPal Core Services
Transaction Engine
Business Logic
Test:
www.sandbox.paypal.com
Production:
www.paypal.com
PayPal Main Interactive Site

In an object-oriented processing model, the interface to SOAP requests/responses is an object in your application's native programming language. Your third-party SOAP client generates business-object interfaces and network stubs from PayPal-provided WSDL and XSD files that specify the PayPal SOAP message structure, its contents, and the PayPal API service bindings. A business application works with data in the form of object properties to send and receive data by calling object methods. The SOAP client handles the details of building the SOAP request, sending it to the PayPal service, and converting the response back to an object.

PayPal WSDL/XSD Schema Definitions

https://developer.mapquest.com/web/products/dev-services/directions-ws


mapquest developers
Sign In | Create Account | Contact Us
| Quick Start | Business Solutions | APIs | Mapping Tools | Resources | Blog | Support
Get MapQuest AppKey
Contact Us

MapQuest Directions API

LICENSED DATA APIs » DIRECTIONS API WEB SERVICE

The MapQuest Directions API allows you to access the patented routing algorithms that power MapQuest.com through simple HTTP requests. The Directions API supports various types of requests, such as fastest and shortest routes, pedestrian, bicycle, and multimodal routes. Optimized routing, date and time routing, and route matrix functionality is also available through the API. Community Edition users can make up to 5,000 free directions requests per day and there is no preset limit for Enterprise Edition users.

Directions API Resources

Documentation
Instructions, code samples, and demos all combined in easy-to-follow, easy-to-use documentation.

Forums
Discuss and collaborate with developers who are using the Directions API.

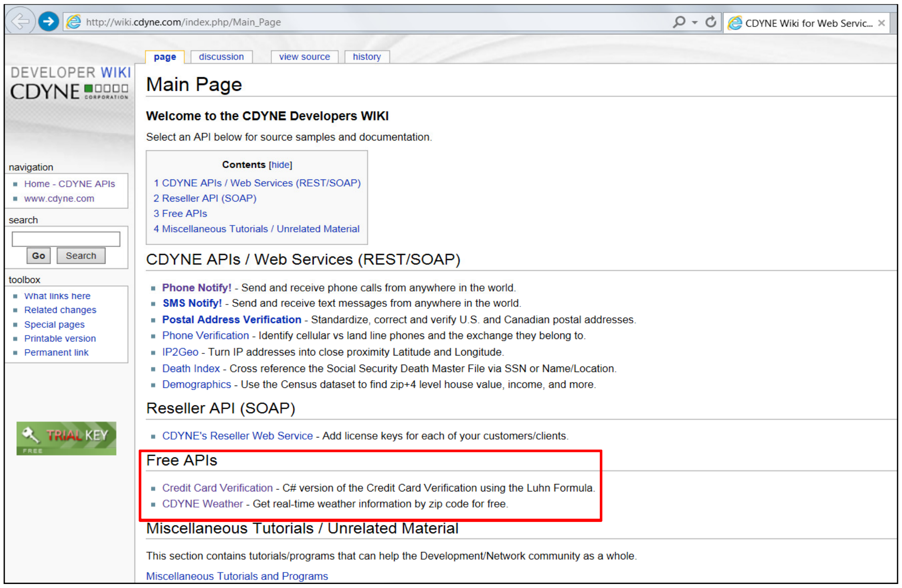
Features

Routing
Provide an origin and a destination address (or latitude and longitude coordinates) and receive the turn-by-turn directions narrative, road shield images, turn direction images, street names, static maneuver maps, and the route shape in a raw or compressed format. Provide multiple address in between the origin and destination to create a multipoint route for up to 50 locations.

Route types
The MapQuest Directions API can calculate the fastest and shortest driving routes as well as pedestrian, bicycle, and multimodal routes, which is a combination of public transit (if available) and walking directions.

Route options
Routes and directions can be customized by setting options -- like avoiding ferries, highways, toll roads, or even specific links in the road network; timed conditions, driving style, vehicle fuel efficiency, locale, boundary displays, side-of-street display, and more.

http://wiki.cdyne.com/index.php/Main_Page



Main Page

Welcome to the CDYNE Developers WIKI

Select an API below for source samples and documentation.

Contents [hide]

1 CDYNE APIs / Web Services (REST/SOAP)
2 Reseller API (SOAP)
3 Free APIs
4 Miscellaneous Tutorials / Unrelated Material

CDYNE APIs / Web Services (REST/SOAP)

- **Phone Notify!** - Send and receive phone calls from anywhere in the world.
- **SMS Notify!** - Send and receive text messages from anywhere in the world.
- **Postal Address Verification** - Standardize, correct and verify U.S. and Canadian postal addresses.
- Phone Verification - Identify cellular vs land line phones and the exchange they belong to.
- IP2Geo - Turn IP addresses into close proximity Latitude and Longitude.
- Death Index - Cross reference the Social Security Death Master File via SSN or Name/Location.
- Demographics - Use the Census dataset to find zip+4 level house value, income, and more.

Reseller API (SOAP)

- CDYNE's Reseller Web Service - Add license keys for each of your customers/clients.

Free APIs

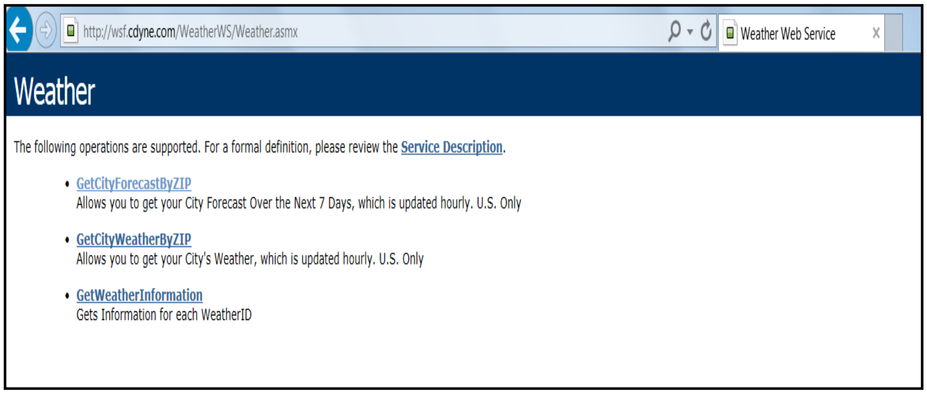
- Credit Card Verification - C# version of the Credit Card Verification using the Luhn Formula.
- CDYNE Weather - Get real-time weather information by zip code for free.

Miscellaneous Tutorials / Unrelated Material

This section contains tutorials/programs that can help the Development/Network community as a whole.

Miscellaneous Tutorials and Programs

http://wsf.cdyne.com/WeatherWS/Weather.asmx



Weather

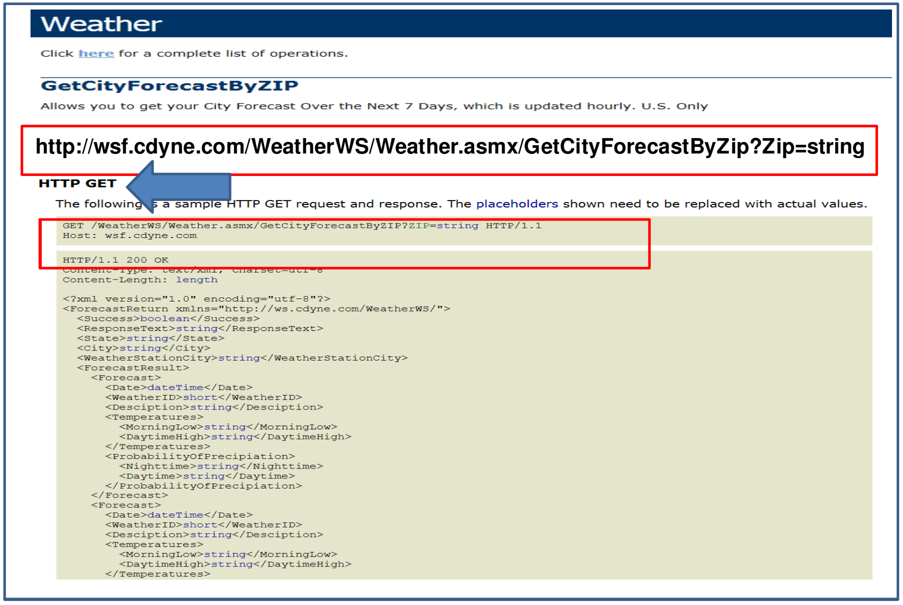
The following operations are supported. For a formal definition, please review the Service Description.

- GetCityForecastByZIP
  Allows you to get your City Forecast Over the Next 7 Days, which is updated hourly. U.S. Only
- GetCityWeatherByZIP
  Allows you to get your City's Weather, which is updated hourly. U.S. Only
- GetWeatherInformation
  Gets Information for each WeatherID

What We'll Cover …

- Web Services 101
- Consuming a REST Web Service
- Consuming a SOAP Web Service
- Providing a SOAP Web Service
- Wrap-up

http://wsf.cdyne.com/WeatherWS/Weather.asmx?op=GetCityForecastByZIP



Weather

Click here for a complete list of operations.

GetCityForecastByZIP

Allows you to get your City Forecast Over the Next 7 Days, which is updated hourly. U.S. Only

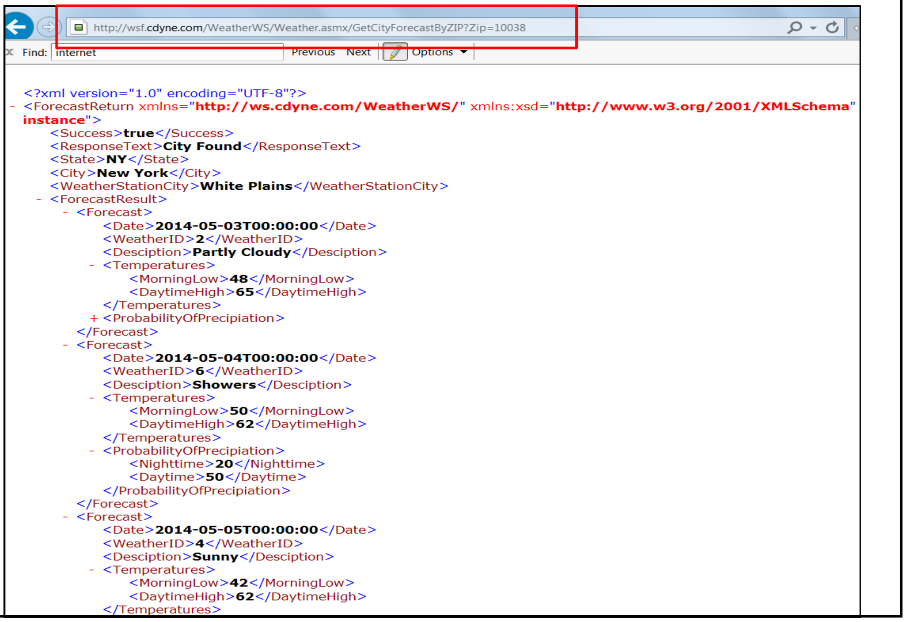
http://wsf.cdyne.com/WeatherWS/Weather.asmx/GetCityForecastByZip?Zip=string

HTTP GET

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

```
GET /WeatherWS/Weather.asmx/GetCityForecastByZIP?ZIP=string HTTP/1.1
Host: wsf.cdyne.com

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

<?xml version="1.0" encoding="utf-8"?>
<ForecastReturn xmlns="http://ws.cdyne.com/WeatherWS/">
  <Success>boolean</Success>
  <ResponseText>string</ResponseText>
  <State>string</State>
  <City>string</City>
  <WeatherStationCity>string</WeatherStationCity>
  <ForecastResult>
    <Forecast>
      <Date>dateTime</Date>
      <WeatherID>short</WeatherID>
      <Desciption>string</Desciption>
      <Temperatures>
        <MorningLow>string</MorningLow>
        <DaytimeHigh>string</DaytimeHigh>
      </Temperatures>
      <ProbabilityOfPrecipiation>
        <Nighttime>string</Nighttime>
        <Daytime>string</Daytime>
      </ProbabilityOfPrecipiation>
    </Forecast>
    <Forecast>
      <Date>dateTime</Date>
      <WeatherID>short</WeatherID>
      <Desciption>string</Desciption>
      <Temperatures>
        <MorningLow>string</MorningLow>
        <DaytimeHigh>string</DaytimeHigh>
      </Temperatures>
```

http://wsf.cdyne.com/WeatherWS/Weather.asmx/GetCityForecastByZIP?Zip=10038



http://wsf.cdyne.com/WeatherWS/Weather.asmx/GetCityForecastByZIP?Zip=10038
Find: internet
Previous
Next
Options

```
  <?xml version="1.0" encoding="UTF-8"?>
- <ForecastReturn xmlns="http://ws.cdyne.com/WeatherWS/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  instance">
      <Success>true</Success>
      <ResponseText>City Found</ResponseText>
      <State>NY</State>
      <City>New York</City>
      <WeatherStationCity>White Plains</WeatherStationCity>
    - <ForecastResult>
      - <Forecast>
            <Date>2014-05-03T00:00:00</Date>
            <WeatherID>2</WeatherID>
            <Desciption>Partly Cloudy</Desciption>
          - <Temperatures>
                <MorningLow>48</MorningLow>
                <DaytimeHigh>65</DaytimeHigh>
            </Temperatures>
          + <ProbabilityOfPrecipiation>
        </Forecast>
      - <Forecast>
            <Date>2014-05-04T00:00:00</Date>
            <WeatherID>6</WeatherID>
            <Desciption>Showers</Desciption>
          - <Temperatures>
                <MorningLow>50</MorningLow>
                <DaytimeHigh>62</DaytimeHigh>
            </Temperatures>
          - <ProbabilityOfPrecipiation>
                <Nighttime>20</Nighttime>
                <Daytime>50</Daytime>
            </ProbabilityOfPrecipiation>
        </Forecast>
      - <Forecast>
            <Date>2014-05-05T00:00:00</Date>
            <WeatherID>4</WeatherID>
            <Desciption>Sunny</Desciption>
          - <Temperatures>
                <MorningLow>42</MorningLow>
                <DaytimeHigh>62</DaytimeHigh>
            </Temperatures>
```

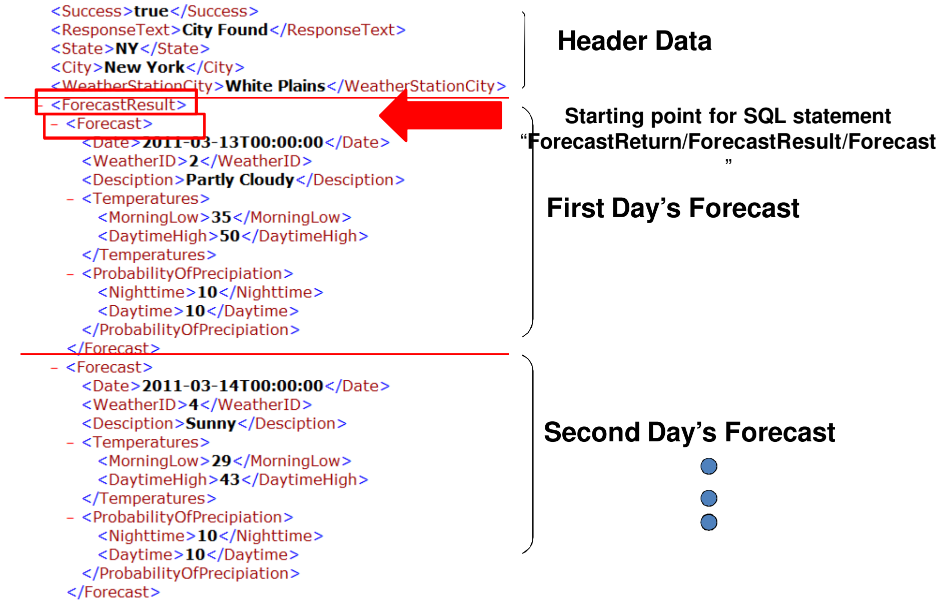
The Power of Using New Native SQL Tools

HTTPGETCLOB

XMLPARSE

XMLTABLE

A Detailed Review of the XML Data



```
<Success>true</Success>
<ResponseText>City Found</ResponseText>
<State>NY</State>
<City>New York</City>
<WeatherStationCity>White Plains</WeatherStationCity>
- <ForecastResult>
  - <Forecast>
      <Date>2011-03-13T00:00:00</Date>
      <WeatherID>2</WeatherID>
      <Desciption>Partly Cloudy</Desciption>
    - <Temperatures>
        <MorningLow>35</MorningLow>
        <DaytimeHigh>50</DaytimeHigh>
      </Temperatures>
    - <ProbabilityOfPrecipiation>
        <Nighttime>10</Nighttime>
        <Daytime>10</Daytime>
      </ProbabilityOfPrecipiation>
    </Forecast>
  - <Forecast>
      <Date>2011-03-14T00:00:00</Date>
      <WeatherID>4</WeatherID>
      <Desciption>Sunny</Desciption>
    - <Temperatures>
        <MorningLow>29</MorningLow>
        <DaytimeHigh>43</DaytimeHigh>
      </Temperatures>
    - <ProbabilityOfPrecipiation>
        <Nighttime>10</Nighttime>
        <Daytime>10</Daytime>
      </ProbabilityOfPrecipiation>
    </Forecast>
```

Header Data

Starting point for SQL statement “ForecastReturn/ForecastResult/Forecast ”

First Day’s Forecast

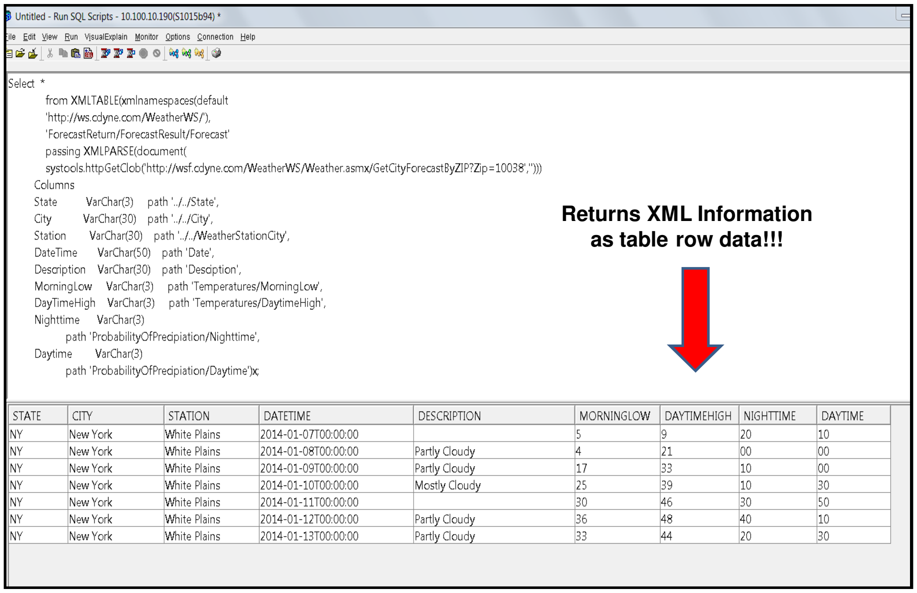
Second Day’s Forecast

●
●
●

The NEW Power of the SQL Select Statement

```
Select  *
          from XMLTABLE(xmlnamespaces(default
          'http://ws.cdyne.com/WeatherWS/'),
          'ForecastReturn/ForecastResult/Forecast'
          passing XMLPARSE(document(
systools.httpGetClob('http://wsf.cdyne.com/WeatherWS/Weather.asmx/Ge
tCityForecastByZIP?Zip=10038','')))
       Columns
       State          VarChar(3)      path '../../State',
       City           VarChar(30)     path '../../City',
       Station        VarChar(30)     path '../../WeatherStationCity',
       DateTime       VarChar(50)     path 'Date',
       Description    VarChar(30)     path 'Desciption',
       MorningLow     VarChar(3)      path 'Temperatures/MorningLow',
       DayTimeHigh    VarChar(3)      path 'Temperatures/DaytimeHigh',
       Nighttime      VarChar(3)
                path 'ProbabilityOfPrecipiation/Nighttime',
       Daytime        VarChar(3)
                path 'ProbabilityOfPrecipiation/Daytime')x;
```

IBM i Navigator Using i7.1 TR6

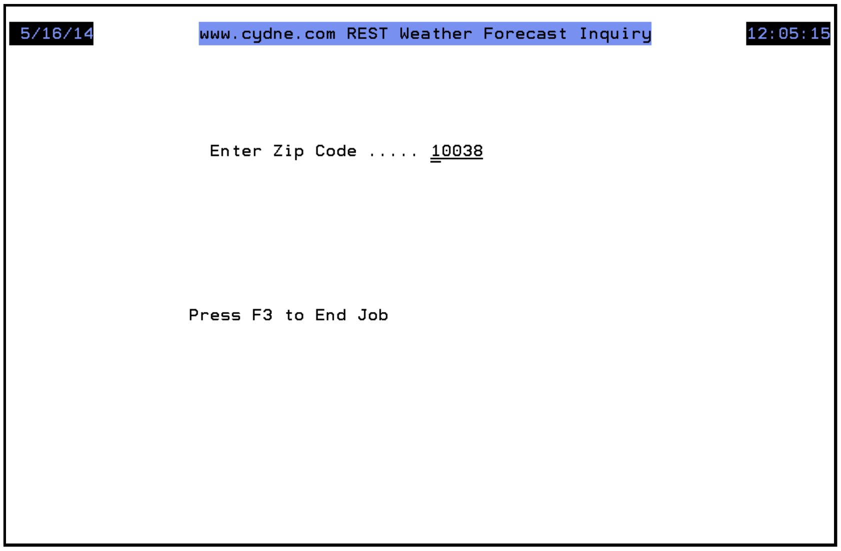

Untitled - Run SQL Scripts - 10.100.10.190(S1015b94) *
File Edit View Run VisualExplain Monitor Options Connection Help

```
Select  *
        from XMLTABLE(xmlnamespaces(default
        'http://ws.cdyne.com/WeatherWS/'),
        'ForecastReturn/ForecastResult/Forecast'
        passing XMLPARSE(document(
        systools.httpGetClob('http://wsf.cdyne.com/WeatherWS/Weather.asmx/GetCityForecastByZIP?Zip=10038','')))
     Columns
     State        VarChar(3)     path '../../State',
     City         VarChar(30)    path '../../City',
     Station      VarChar(30)    path '../../WeatherStationCity',
     DateTime     VarChar(50)    path 'Date',
     Description  VarChar(30)    path 'Desciption',
     MorningLow   VarChar(3)     path 'Temperatures/MorningLow',
     DayTimeHigh  VarChar(3)     path 'Temperatures/DaytimeHigh',
     Nighttime    VarChar(3)
           path 'ProbabilityOfPrecipiation/Nighttime',
     Daytime      VarChar(3)
           path 'ProbabilityOfPrecipiation/Daytime')x;
```

Returns XML Information as table row data!!!

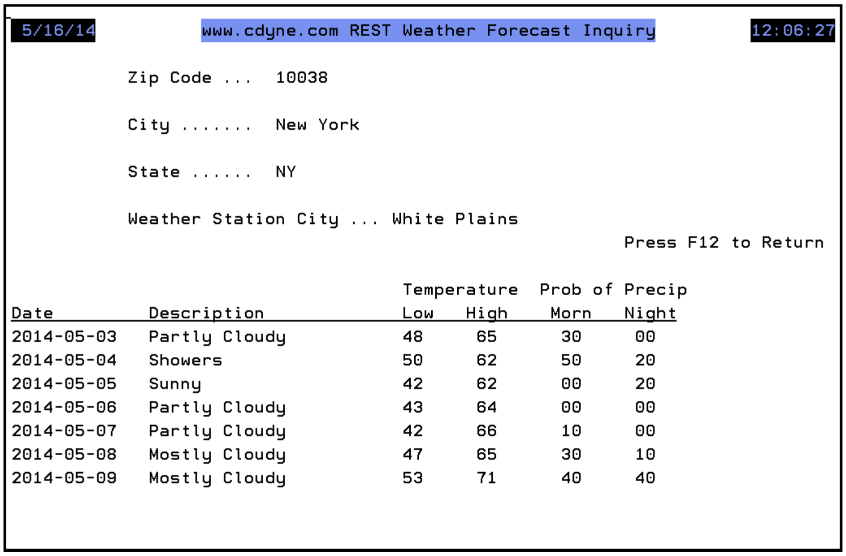
STATE	CITY	STATION	DATETIME	DESCRIPTION	MORNINGLOW	DAYTIMEHIGH	NIGHTTIME	DAYTIME
NY	New York	White Plains	2014-01-07T00:00:00		5	9	20	10
NY	New York	White Plains	2014-01-08T00:00:00	Partly Cloudy	4	21	00	00
NY	New York	White Plains	2014-01-09T00:00:00	Partly Cloudy	17	33	10	00
NY	New York	White Plains	2014-01-10T00:00:00	Mostly Cloudy	25	39	10	30
NY	New York	White Plains	2014-01-11T00:00:00		30	46	30	50
NY	New York	White Plains	2014-01-12T00:00:00	Partly Cloudy	36	48	40	10
NY	New York	White Plains	2014-01-13T00:00:00	Partly Cloudy	33	44	20	30

Simple Green Screen Application



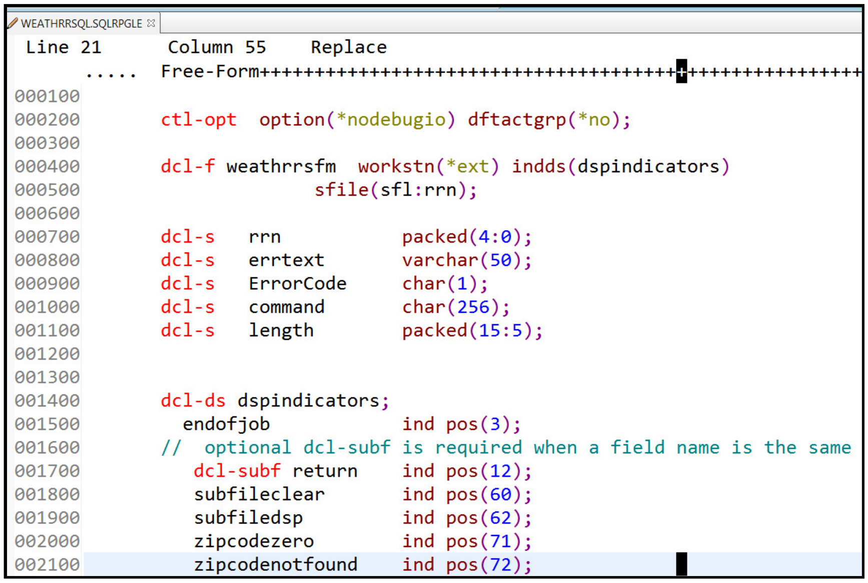
```
 5/16/14          www.cydne.com REST Weather Forecast Inquiry         12:05:15



                    Enter Zip Code ..... 10038




                  Press F3 to End Job
```

Simple Green Screen Application



```
 5/16/14          www.cdyne.com REST Weather Forecast Inquiry         12:06:27

           Zip Code ...  10038

           City .......  New York

           State ......  NY

           Weather Station City ... White Plains
                                                      Press F12 to Return

                                    Temperature  Prob of Precip
Date        Description             Low    High   Morn   Night
2014-05-03  Partly Cloudy           48     65     30     00
2014-05-04  Showers                 50     62     50     20
2014-05-05  Sunny                   42     62     00     20
2014-05-06  Partly Cloudy           43     64     00     00
2014-05-07  Partly Cloudy           42     66     10     00
2014-05-08  Mostly Cloudy           47     65     30     10
2014-05-09  Mostly Cloudy           53     71     40     40
```

RPG using Embedded SQL


WEATHRRSQL.SQLRPGLE
Line 21   Column 55   Replace
..... Free-Form++

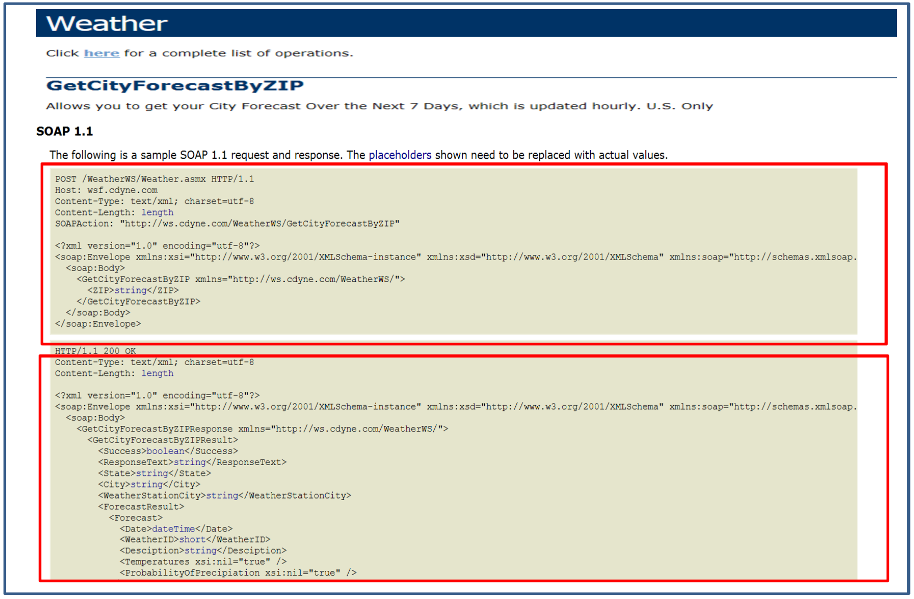
```

ctl-opt  option(*nodebugio) dftactgrp(*no);

dcl-f weathrrsfm  workstn(*ext) indds(dspindicators)
              sfile(sfl:rrn);

dcl-s   rrn           packed(4:0);
dcl-s   errtext       varchar(50);
dcl-s   ErrorCode     char(1);
dcl-s   command       char(256);
dcl-s   length        packed(15:5);


dcl-ds dspindicators;
  endofjob            ind pos(3);
//  optional dcl-subf is required when a field name is the same
   dcl-subf return    ind pos(12);
   subfileclear       ind pos(60);
   subfiledsp         ind pos(62);
   zipcodezero        ind pos(71);
   zipcodenotfound    ind pos(72);
```

What We'll Cover …

- Web Services 101
- Consuming a REST Web Service
- Consuming a SOAP Web Service
- Providing a SOAP Web Service
- Wrap-up

SOAP web services communicate with messages, like email

http://wsf.cdyne.com/WeatherWS/Weather.asmx?op=GetCityForecastByZIP



Weather

Click here for a complete list of operations.

GetCityForecastByZIP

Allows you to get your City Forecast Over the Next 7 Days, which is updated hourly. U.S. Only

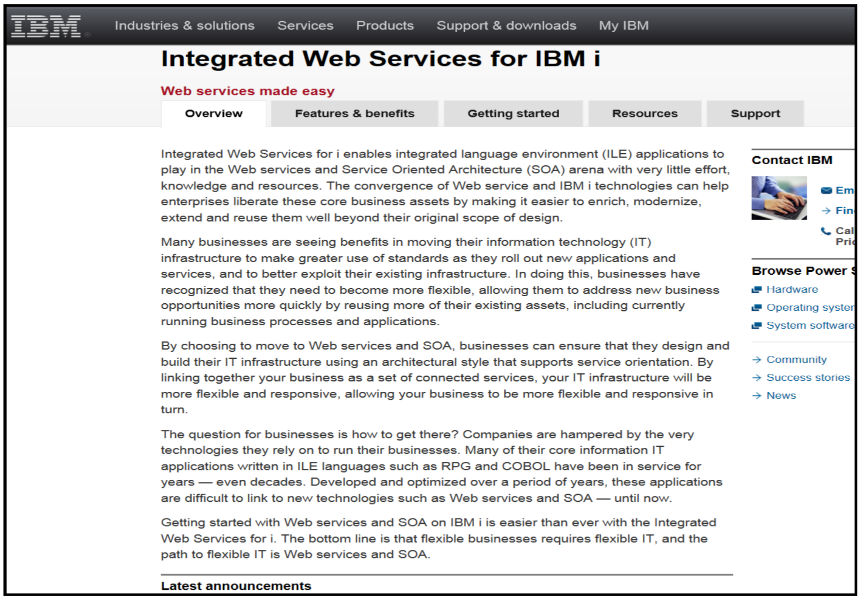
SOAP 1.1

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

```
POST /WeatherWS/Weather.asmx HTTP/1.1
Host: wsf.cdyne.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://ws.cdyne.com/WeatherWS/GetCityForecastByZIP"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.
  <soap:Body>
    <GetCityForecastByZIP xmlns="http://ws.cdyne.com/WeatherWS/">
      <ZIP>string</ZIP>
    </GetCityForecastByZIP>
  </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:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.
  <soap:Body>
    <GetCityForecastByZIPResponse xmlns="http://ws.cdyne.com/WeatherWS/">
      <GetCityForecastByZIPResult>
        <Success>boolean</Success>
        <ResponseText>string</ResponseText>
        <State>string</State>
        <City>string</City>
        <WeatherStationCity>string</WeatherStationCity>
        <ForecastResult>
          <Forecast>
            <Date>dateTime</Date>
            <WeatherID>short</WeatherID>
            <Desciption>string</Desciption>
            <Temperatures xsi:nil="true" />
            <ProbabilityOfPrecipiation xsi:nil="true" />
```

http://www-03.ibm.com/systems/i/software/iws/



Integrated Web Services for IBM i

Web services made easy

Integrated Web Services for i enables integrated language environment (ILE) applications to play in the Web services and Service Oriented Architecture (SOA) arena with very little effort, knowledge and resources. The convergence of Web service and IBM i technologies can help enterprises liberate these core business assets by making it easier to enrich, modernize, extend and reuse them well beyond their original scope of design.

Many businesses are seeing benefits in moving their information technology (IT) infrastructure to make greater use of standards as they roll out new applications and services, and to better exploit their existing infrastructure. In doing this, businesses have recognized that they need to become more flexible, allowing them to address new business opportunities more quickly by reusing more of their existing assets, including currently running business processes and applications.

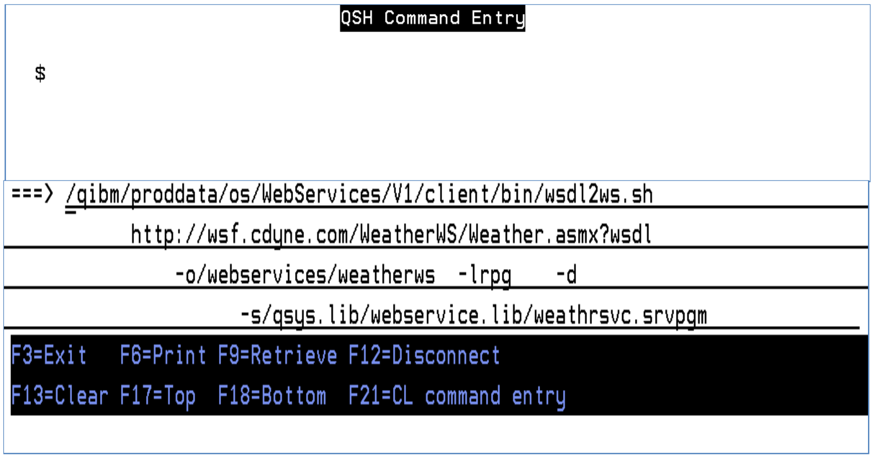
By choosing to move to Web services and SOA, businesses can ensure that they design and build their IT infrastructure using an architectural style that supports service orientation. By linking together your business as a set of connected services, your IT infrastructure will be more flexible and responsive, allowing your business to be more flexible and responsive in turn.

The question for businesses is how to get there? Companies are hampered by the very technologies they rely on to run their businesses. Many of their core information IT applications written in ILE languages such as RPG and COBOL have been in service for years — even decades. Developed and optimized over a period of years, these applications are difficult to link to new technologies such as Web services and SOA — until now.

Getting started with Web services and SOA on IBM i is easier than ever with the Integrated Web Services for i. The bottom line is that flexible businesses requires flexible IT, and the path to flexible IT is Web services and SOA.

Latest announcements

Use WSDL2WS in QSHELL

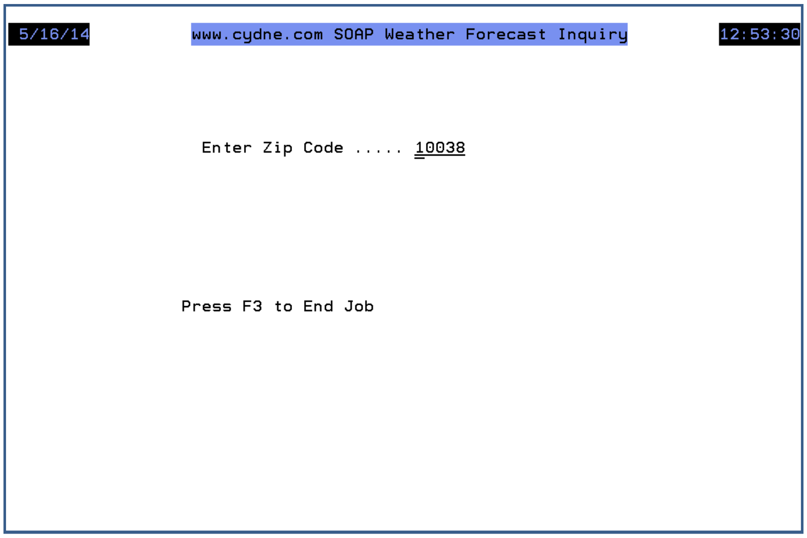


```
 $

===> /qibm/proddata/os/WebServices/V1/client/bin/wsdl2ws.sh
         http://wsf.cdyne.com/WeatherWS/Weather.asmx?wsdl
              -o/webservices/weatherws  -lrpg    -d
                      -s/qsys.lib/webservice.lib/weathrsvc.srvpgm
F3=Exit   F6=Print F9=Retrieve F12=Disconnect
F13=Clear F17=Top  F18=Bottom  F21=CL command entry
```

Lines

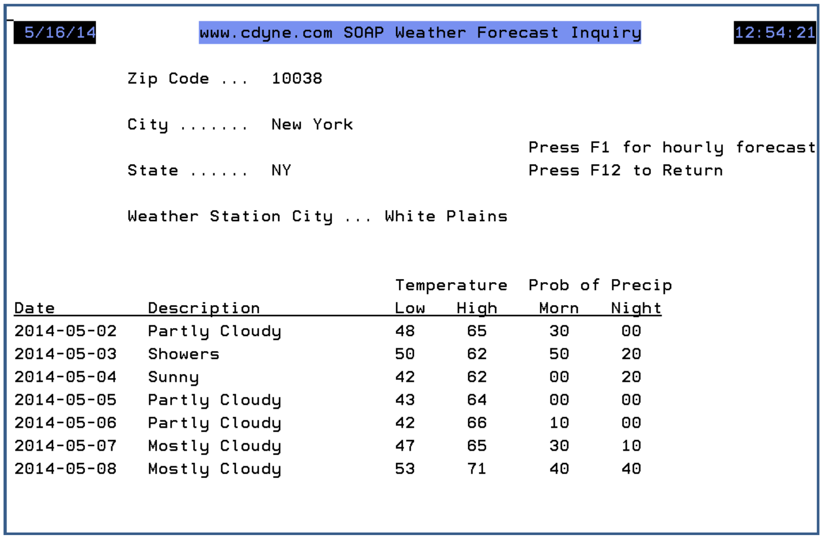
1) **Invokes the command**
2) **WSDL**
3) **Command processing options**
4) **Identifies service program to be created**

Simple Green Screen Application



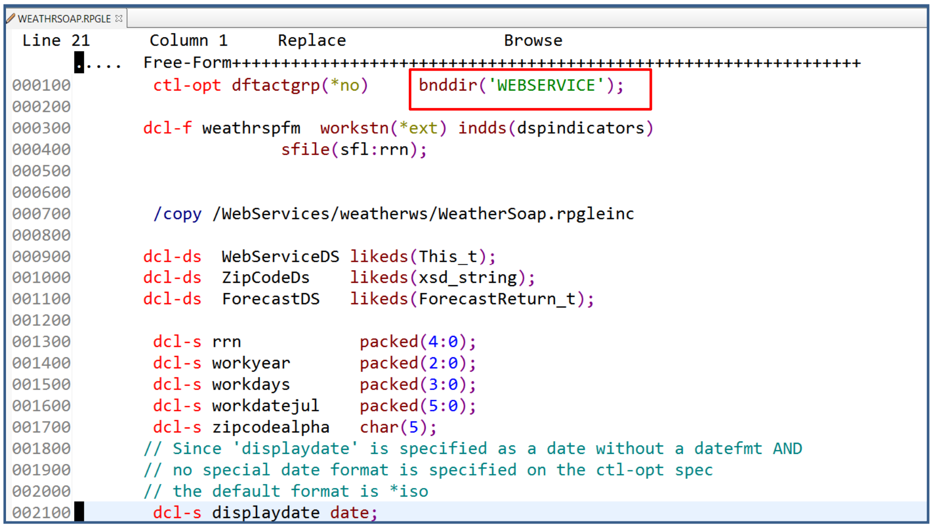
```
5/16/14          www.cydne.com SOAP Weather Forecast Inquiry          12:53:30



                       Enter Zip Code ..... 10038





                     Press F3 to End Job
```

Simple Green Screen Application



```
5/16/14          www.cdyne.com SOAP Weather Forecast Inquiry          12:54:21

           Zip Code ...   10038

           City .......   New York
                                                    Press F1 for hourly forecast
           State ......   NY                        Press F12 to Return

           Weather Station City ... White Plains


                                      Temperature  Prob of Precip
Date         Description              Low    High    Morn    Night
2014-05-02   Partly Cloudy            48     65       30      00
2014-05-03   Showers                  50     62       50      20
2014-05-04   Sunny                    42     62       00      20
2014-05-05   Partly Cloudy            43     64       00      00
2014-05-06   Partly Cloudy            42     66       10      00
2014-05-07   Mostly Cloudy            47     65       30      10
2014-05-08   Mostly Cloudy            53     71       40      40
```

This program calls service program procedures



```
 ctl-opt dftactgrp(*no)     bnddir('WEBSERVICE');

dcl-f weathrspfm  workstn(*ext) indds(dspindicators)
              sfile(sfl:rrn);


 /copy /WebServices/weatherws/WeatherSoap.rpgleinc

dcl-ds  WebServiceDS likeds(This_t);
dcl-ds  ZipCodeDs    likeds(xsd_string);
dcl-ds  ForecastDS   likeds(ForecastReturn_t);

 dcl-s rrn            packed(4:0);
 dcl-s workyear       packed(2:0);
 dcl-s workdays       packed(3:0);
 dcl-s workdatejul    packed(5:0);
 dcl-s zipcodealpha   char(5);
// Since 'displaydate' is specified as a date without a datefmt AND
// no special date format is specified on the ctl-opt spec
// the default format is *iso
 dcl-s displaydate date;
```

What We'll Cover …

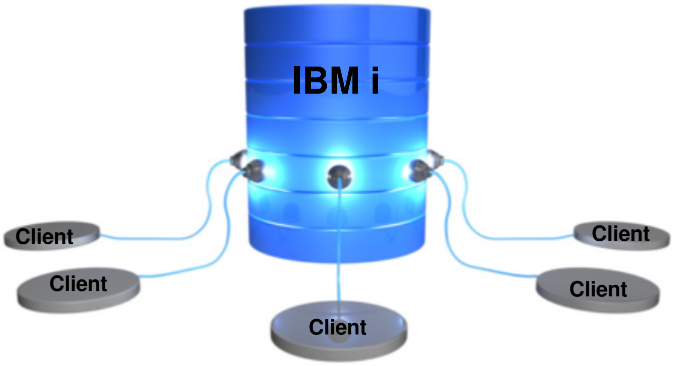
- Web Services 101
- Consuming a REST Web Service
- Consuming a SOAP Web Service
- Providing a SOAP Web Service
- Wrap-up

Need to Expose Our Item Information Across The Globe

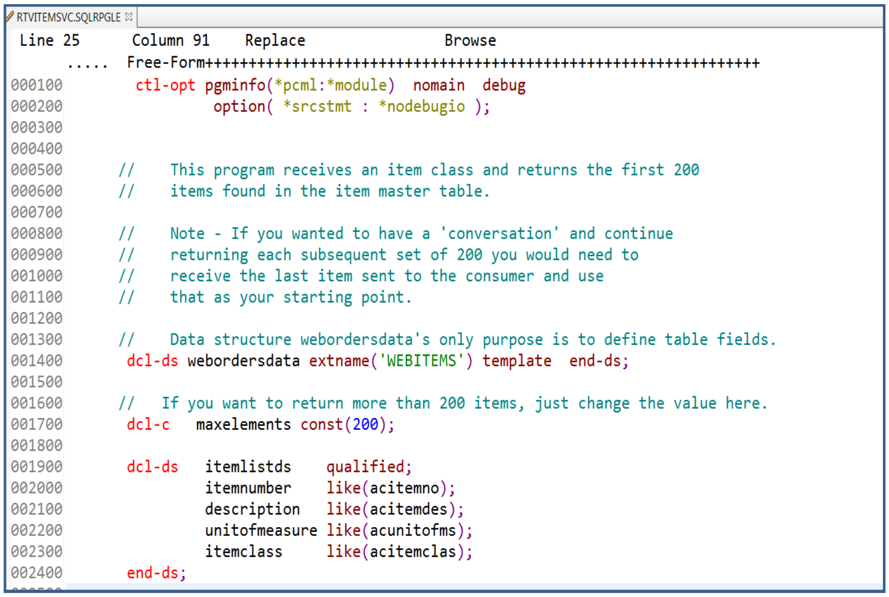
Receive a SOAP request for items for a certain item class

We will return to the client a list of items for the request item class

Web Services Server
Runs Program RTVITEMSVC



RPG Program used as a web service

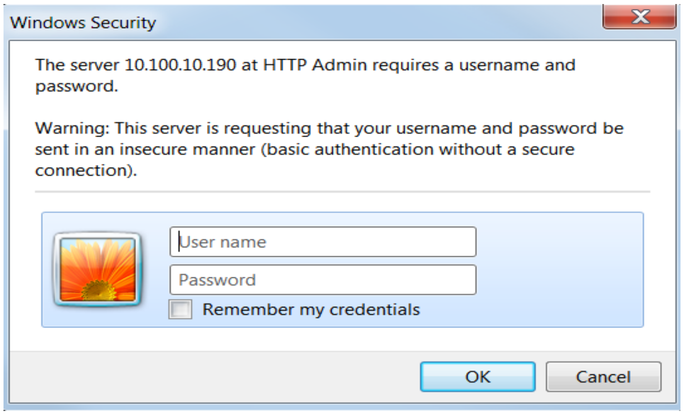

RTVITEMSVC.SQLRPGLE
Line 25   Column 91   Replace   Browse
..... Free-Form+++

```
  ctl-opt pgminfo(*pcml:*module)  nomain  debug
           option( *srcstmt : *nodebugio );


//    This program receives an item class and returns the first 200
//    items found in the item master table.

//    Note - If you wanted to have a 'conversation' and continue
//    returning each subsequent set of 200 you would need to
//    receive the last item sent to the consumer and use
//    that as your starting point.

//    Data structure webordersdata's only purpose is to define table fields.
 dcl-ds webordersdata extname('WEBITEMS') template  end-ds;

//   If you want to return more than 200 items, just change the value here.
 dcl-c   maxelements const(200);

 dcl-ds   itemlistds    qualified;
          itemnumber    like(acitemno);
          description   like(acitemdes);
          unitofmeasure like(acunitofms);
          itemclass     like(acitemclas);
 end-ds;
```

Create your own web services using your own programs

Step 1 – STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)

Step 2 – From a browser go to http://hostname:2001/HTTPAdmin

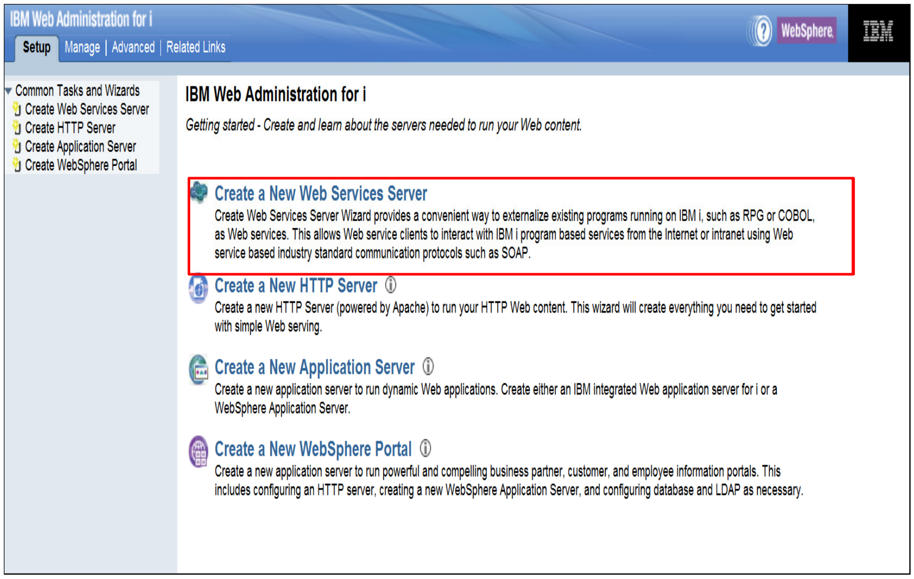


Windows Security

The server 10.100.10.190 at HTTP Admin requires a username and password.

Warning: This server is requesting that your username and password be sent in an insecure manner (basic authentication without a secure connection).

http://hostname:2001/HTTPAdmin



IBM Web Administration for i
Setup | Manage | Advanced | Related Links
WebSphere
IBM
Common Tasks and Wizards
Create Web Services Server
Create HTTP Server
Create Application Server
Create WebSphere Portal

IBM Web Administration for i

Getting started - Create and learn about the servers needed to run your Web content.

Create a New Web Services Server

Create Web Services Server Wizard provides a convenient way to externalize existing programs running on IBM i, such as RPG or COBOL, as Web services. This allows Web service clients to interact with IBM i program based services from the Internet or intranet using Web service based industry standard communication protocols such as SOAP.

Create a New HTTP Server

Create a new HTTP Server (powered by Apache) to run your HTTP Web content. This wizard will create everything you need to get started with simple Web serving.

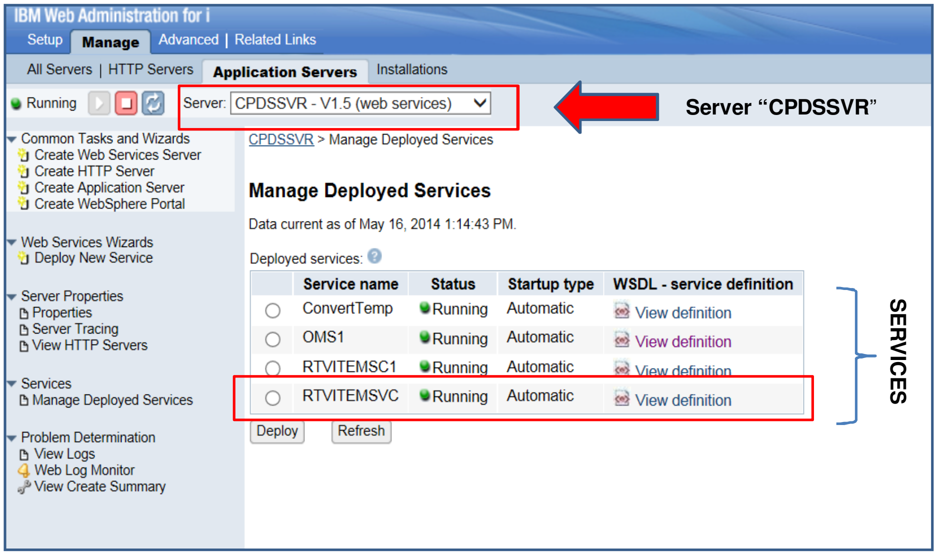
Create a New Application Server

Create a new application server to run dynamic Web applications. Create either an IBM integrated Web application server for i or a WebSphere Application Server.

Create a New WebSphere Portal

Create a new application server to run powerful and compelling business partner, customer, and employee information portals. This includes configuring an HTTP server, creating a new WebSphere Application Server, and configuring database and LDAP as necessary.

Manage Deployed Services



Server: CPDSSVR - V1.5 (web services)

Server “CPDSSVR”

CPDSSVR > Manage Deployed Services

Manage Deployed Services

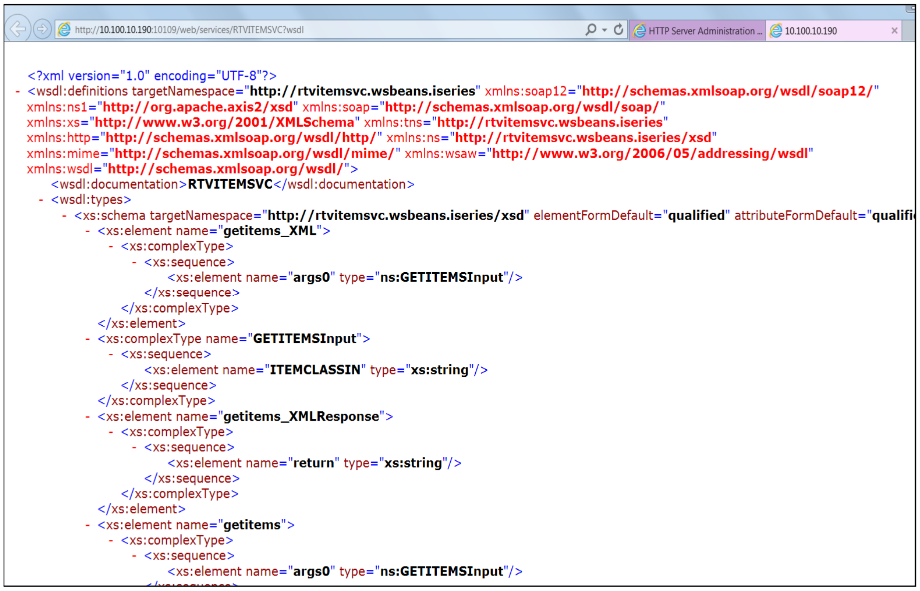
Data current as of May 16, 2014 1:14:43 PM.

Deployed services:

	Service name	Status	Startup type	WSDL - service definition
○	ConvertTemp	●Running	Automatic	View definition
○	OMS1	●Running	Automatic	View definition
○	RTVITEMSC1	●Running	Automatic	View definition
○	RTVITEMSVC	●Running	Automatic	View definition

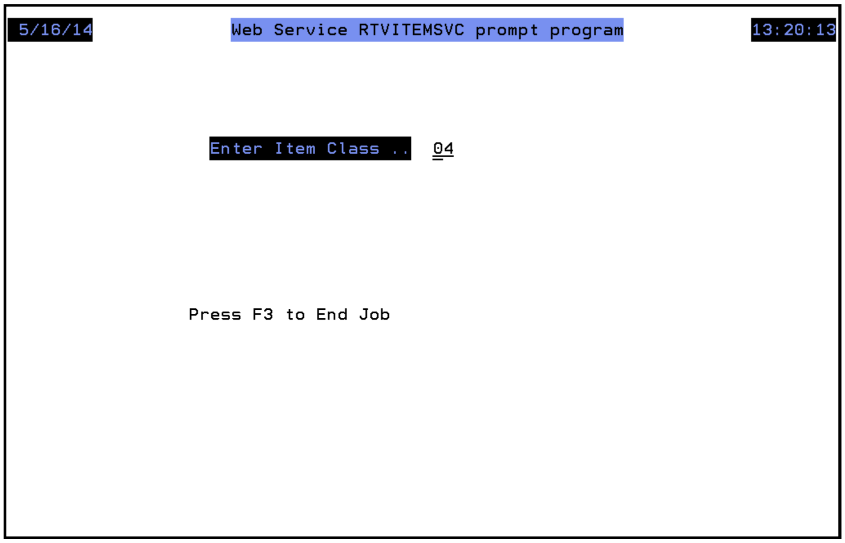
SERVICES

http://10.100.10.190:10109/web/services/RTVITEMSVC?wsdl


http://10.100.10.190:10109/web/services/RTVITEMSVC?wsdl
HTTP Server Administration ...
10.100.10.190

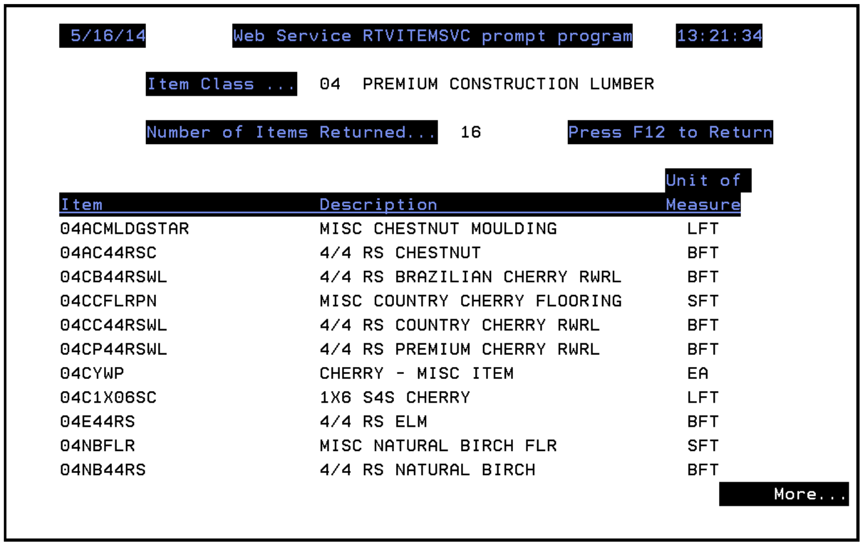
```
  <?xml version="1.0" encoding="UTF-8"?>
- <wsdl:definitions targetNamespace="http://rtvitemsvc.wsbeans.iseries" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:ns1="http://org.apache.axis2/xsd" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://rtvitemsvc.wsbeans.iseries"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:ns="http://rtvitemsvc.wsbeans.iseries/xsd"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
      <wsdl:documentation>RTVITEMSVC</wsdl:documentation>
    - <wsdl:types>
      - <xs:schema targetNamespace="http://rtvitemsvc.wsbeans.iseries/xsd" elementFormDefault="qualified" attributeFormDefault="qualifi
        - <xs:element name="getitems_XML">
          - <xs:complexType>
            - <xs:sequence>
                <xs:element name="args0" type="ns:GETITEMSInput"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        - <xs:complexType name="GETITEMSInput">
          - <xs:sequence>
              <xs:element name="ITEMCLASSIN" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
        - <xs:element name="getitems_XMLResponse">
          - <xs:complexType>
            - <xs:sequence>
                <xs:element name="return" type="xs:string"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        - <xs:element name="getitems">
          - <xs:complexType>
            - <xs:sequence>
                <xs:element name="args0" type="ns:GETITEMSInput"/>
              </xs:sequence>
```

Simple Green Screen Application



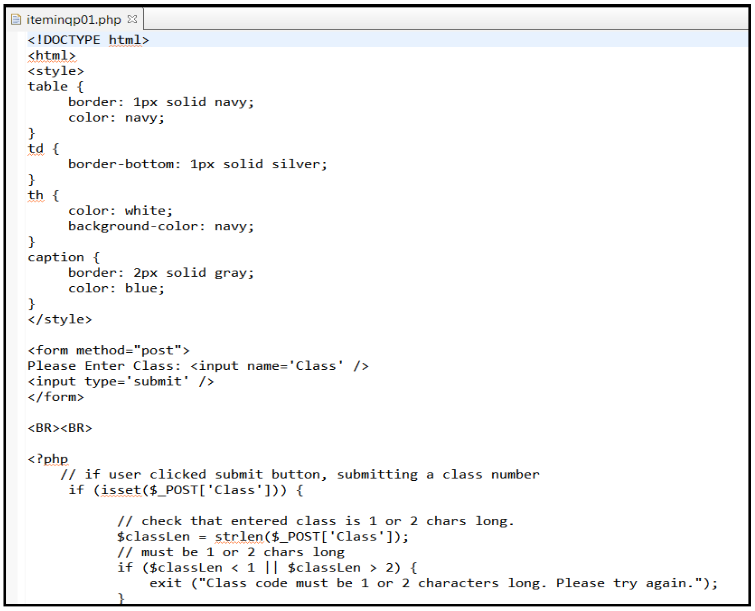
```
5/16/14            Web Service RTVITEMSVC prompt program            13:20:13



                   Enter Item Class ..  04




                 Press F3 to End Job
```

Simple Green Screen Application



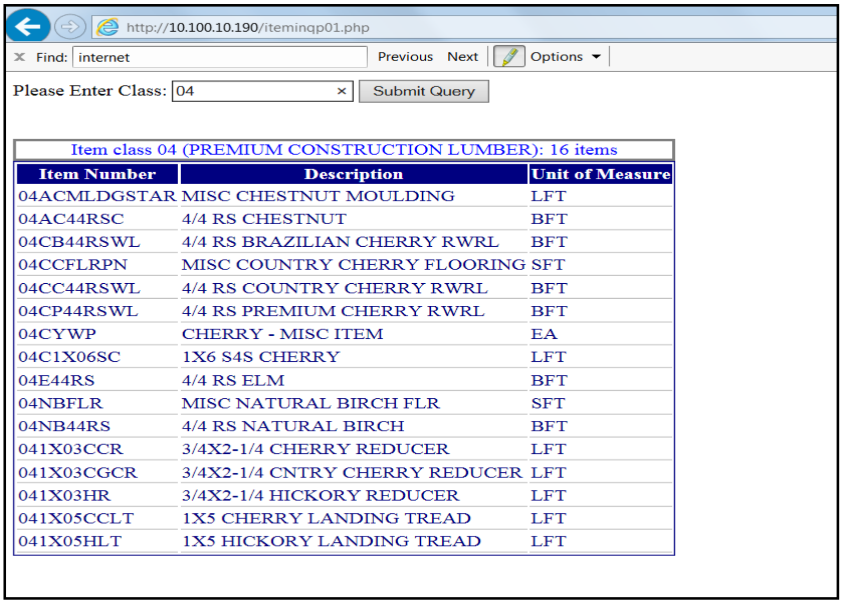
```
5/16/14       Web Service RTVITEMSVC prompt program    13:21:34

        Item Class ...  04  PREMIUM CONSTRUCTION LUMBER

        Number of Items Returned...  16      Press F12 to Return

                                                          Unit of
Item                    Description                       Measure
04ACMLDGSTAR            MISC CHESTNUT MOULDING              LFT
04AC44RSC               4/4 RS CHESTNUT                     BFT
04CB44RSWL              4/4 RS BRAZILIAN CHERRY RWRL        BFT
04CCFLRPN               MISC COUNTRY CHERRY FLOORING        SFT
04CC44RSWL              4/4 RS COUNTRY CHERRY RWRL          BFT
04CP44RSWL              4/4 RS PREMIUM CHERRY RWRL          BFT
04CYWP                  CHERRY - MISC ITEM                  EA
04C1X06SC               1X6 S4S CHERRY                      LFT
04E44RS                 4/4 RS ELM                          BFT
04NBFLR                 MISC NATURAL BIRCH FLR              SFT
04NB44RS                4/4 RS NATURAL BIRCH                BFT
                                                                 More...
```

PHP Program using the same web service



```
<!DOCTYPE html>
<html>
<style>
table {
     border: 1px solid navy;
     color: navy;
}
td {
     border-bottom: 1px solid silver;
}
th {
     color: white;
     background-color: navy;
}
caption {
     border: 2px solid gray;
     color: blue;
}
</style>

<form method="post">
Please Enter Class: <input name='Class' />
<input type='submit' />
</form>

<BR><BR>

<?php
    // if user clicked submit button, submitting a class number
     if (isset($_POST['Class'])) {

           // check that entered class is 1 or 2 chars long.
           $classLen = strlen($_POST['Class']);
           // must be 1 or 2 chars long
           if ($classLen < 1 || $classLen > 2) {
               exit ("Class code must be 1 or 2 characters long. Please try again.");
           }
```

PHP Program using the same web service



Please Enter Class: 04

Item class 04 (PREMIUM CONSTRUCTION LUMBER): 16 items

Item Number	Description	Unit of Measure
04ACMLDGSTAR	MISC CHESTNUT MOULDING	LFT
04AC44RSC	4/4 RS CHESTNUT	BFT
04CB44RSWL	4/4 RS BRAZILIAN CHERRY RWRL	BFT
04CCFLRPN	MISC COUNTRY CHERRY FLOORING	SFT
04CC44RSWL	4/4 RS COUNTRY CHERRY RWRL	BFT
04CP44RSWL	4/4 RS PREMIUM CHERRY RWRL	BFT
04CYWP	CHERRY - MISC ITEM	EA
04C1X06SC	1X6 S4S CHERRY	LFT
04E44RS	4/4 RS ELM	BFT
04NBFLR	MISC NATURAL BIRCH FLR	SFT
04NB44RS	4/4 RS NATURAL BIRCH	BFT
041X03CCR	3/4X2-1/4 CHERRY REDUCER	LFT
041X03CGCR	3/4X2-1/4 CNTRY CHERRY REDUCER	LFT
041X03HR	3/4X2-1/4 HICKORY REDUCER	LFT
041X05CCLT	1X5 CHERRY LANDING TREAD	LFT
041X05HLT	1X5 HICKORY LANDING TREAD	LFT

What We'll Cover …

- Web Services 101
- Consuming a REST Web Service
- Consuming a SOAP Web Service
- Providing a SOAP Web Service
- Wrap-up

COMMON Europe

Enterprise Modernization Tour

Open your World Wide Windows and Supply Real-Time Information To Your Applications

COMMON EUROPE

Arcad SOFTWARE