Rest Web Service on IBM i

Mike Pavlak
Solutions Consultant
mike.p@zend.com

Agenda

- Zend Server Update
- Discuss web services in general
- DB2 access
- Product Availability RPG Program
- Q&A

What's New with Zend Server

IBM Relationship History

Zend is an IBM strategic business partner

IBM selected Zend to bring PHP to IBM i in 2005

Beta, then Core, then Platform now Server

Thousands of downloads from Zend.com

Zend Server part of IBM i6.1 & i7.1 (distribution media)

Products are always available at Zend.com

IBM Relationship FAQ's

What do we get?

- Zend Server Basic (Formerly Community Edition)
- Zend Studio (Licensed for IBM only!)
- 1 year web support and updates (email/web 72 hour response)
- Phone support & improved SLA available from Zend at a charge

What happens after 1 year?

- Products will still run as perpetual license
- No updates to studio, server at Zend discretion
- Forum support only
- Support available for purchase from Zend

What Is Zend Server?

Production quality PHP stack

PHP, ZF2, DB connectivity, debugging extensions, and much more

Three Editions with Zend Server 6

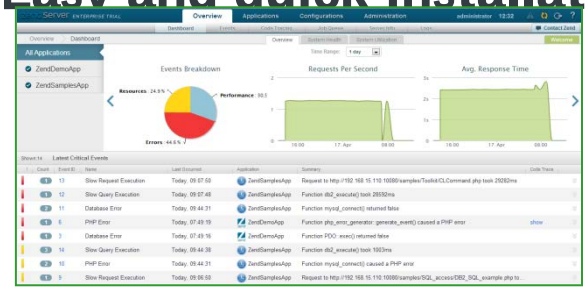
Basic Edition – Included with IBM SWMA,

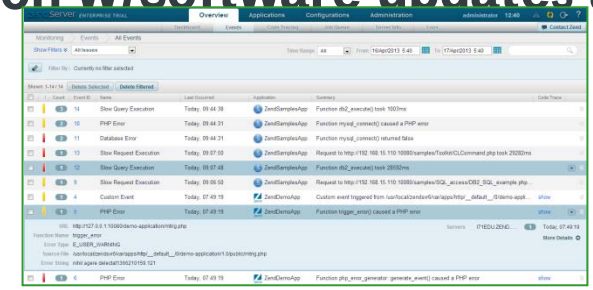
Enjoy Professional & Enterprise features for 7 days, upgrade to 28

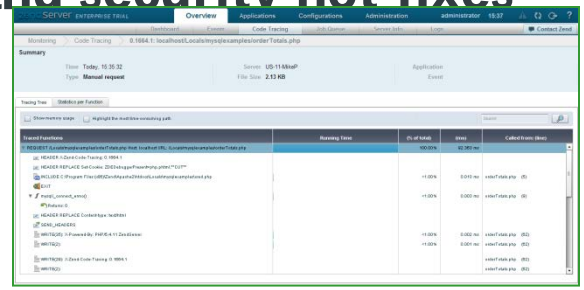
Application monitoring and diagnostics (integrated with Zend Studio)

Multi-level performance enhancement capabilities

Easy and quick installation w/Software updates and security hot fixes



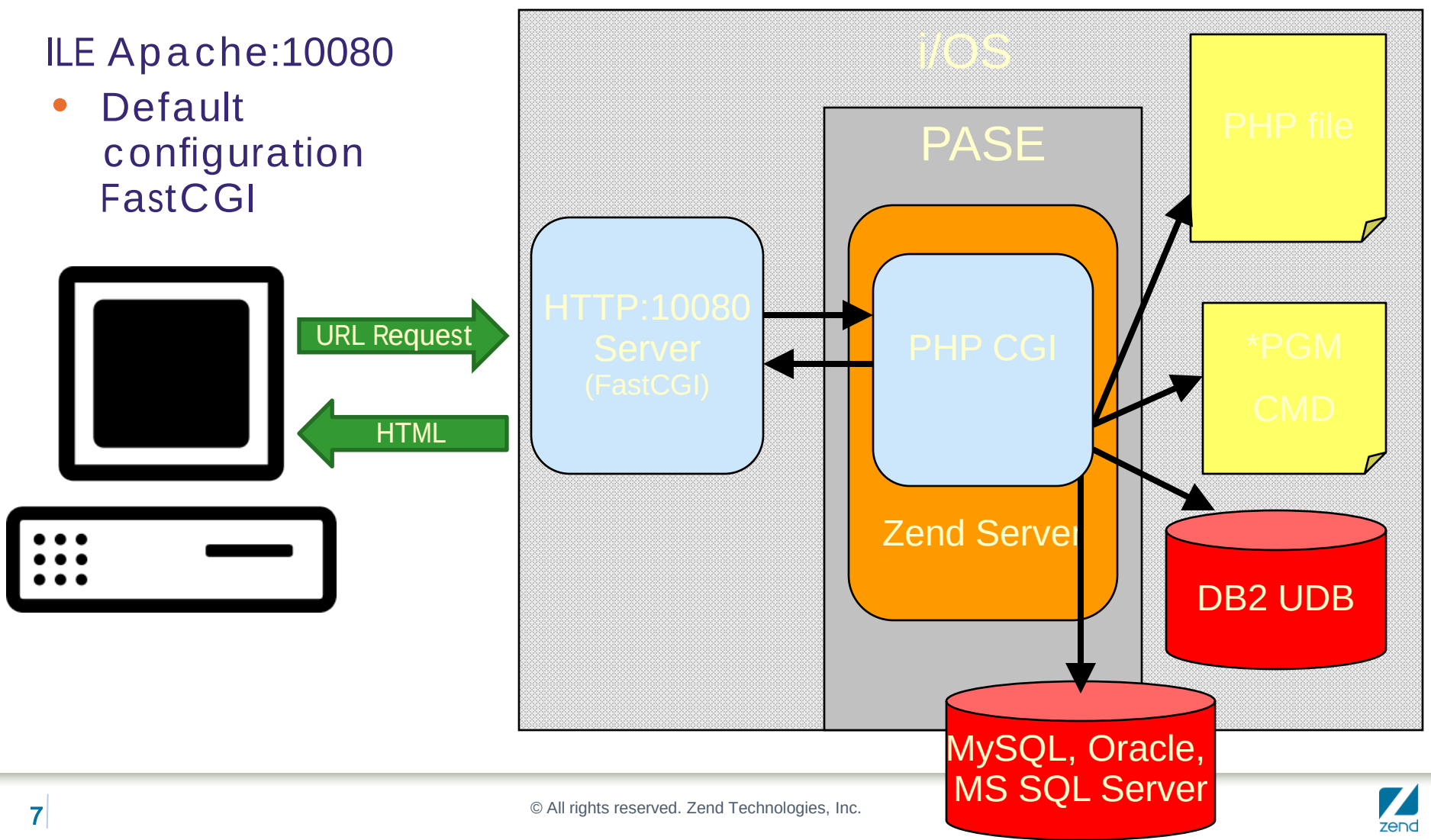




Zend Server Under the Covers

ILE Apache:10080

- Default configuration FastCGI



Zend Server 5.x Users

Must install Zend Server 6 side by side with 5.x

Uses new directories

- Copy your scripts to new directories, or
- Use paths in /usr/local and change include path

Uses different port 10080 rather than 10088

Once successfully on Zend Server 6

- Remove Zend Server 5.6 – DLTLICPGM (GO LICPGM, Option 12)

What's up with Web Services

zend

Introduction to Web Services

- **Tightly coupled**
  - Mine, mine, mine…I own it all! mwa-ha-ha-ha-ha
  - API program calls
  - Stored procedures and ODBC
- **Loosely coupled**
  - We don't own all of our business logic
  - Web Services
  - Mash-ups

Direct Program Call

Advantages
Compiled code
Stable, predictable & efficient
Changes required on both sides ensure security
Disadvantages
Changes require recompile
Test, test, test
Intersystem calls cumbersome

Usher in the new age of computing!

- **Books that are the basis for my position**
  - The Cathedral & the Bazaar by Eric Raymond
    - Open Source is OK!
    - Dynamic is in, Monolithic is out
    - But there is room for both…for a while
  - Small Pieces Loosely Joined by David Weinberger
    - Modular is better!
    - Many small pieces around the Internet
    - Scattered amongst the cloud!

What is a Web Service?

- **W3C says…**
  - A software system designed to support interoperable machine to machine interaction over a network.
- **Key words**
  - Software system
  - Interoperable
  - Machine to machine
  - Network
- **It's really just another way to call programs!**

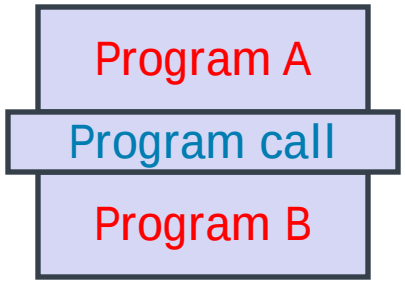
Who uses Web Services?

- **Web presence today**
  - Google ➔ Maps, Spell checkers, etc.
  - Amazon.com ➔ Product availability, Order processing
  - Salesforce.com ➔ Application Integration
  - TerraServer ➔ Cloud based applications
  - Many, many more
- **API layer replacement tomorrow**
  - Inter-program communication
  - Inter-system communication

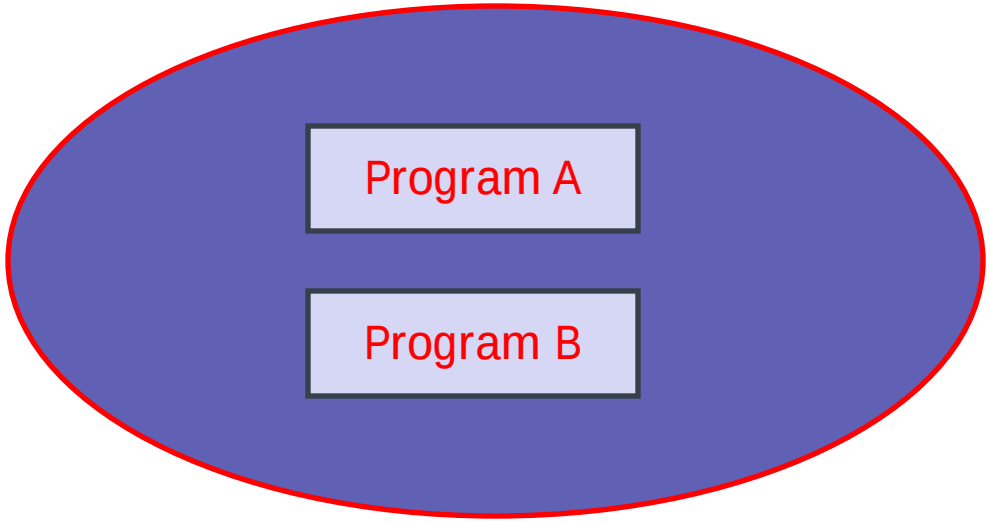
Why use Web Services?

- **SOA Layer**
  - Fancy term for modern program calls!
  - Parts are self defining and self documenting
  - Code is modular from the start
  - Facilitate the upgrade process

Yesterday's program call



Today's program call



What types

- SOAP
  - Simple Object Access Protocol
  - Self defining
  - Can be highly complex
- ReST
  - Representational State Transfer
  - Much like API call
  - Requires external documentation

Data Access using ReST

zend

One example for two methods

Part 1

- Form with drop down info populated by web service
- Select product and then move to part two
- No data on local IBM i

Part 2

- Take selected item and call web service
- Remote web service calls RPG program
- RPG program develops on hand quantity & returns
- Again, no data on local IBM i

Two system setup

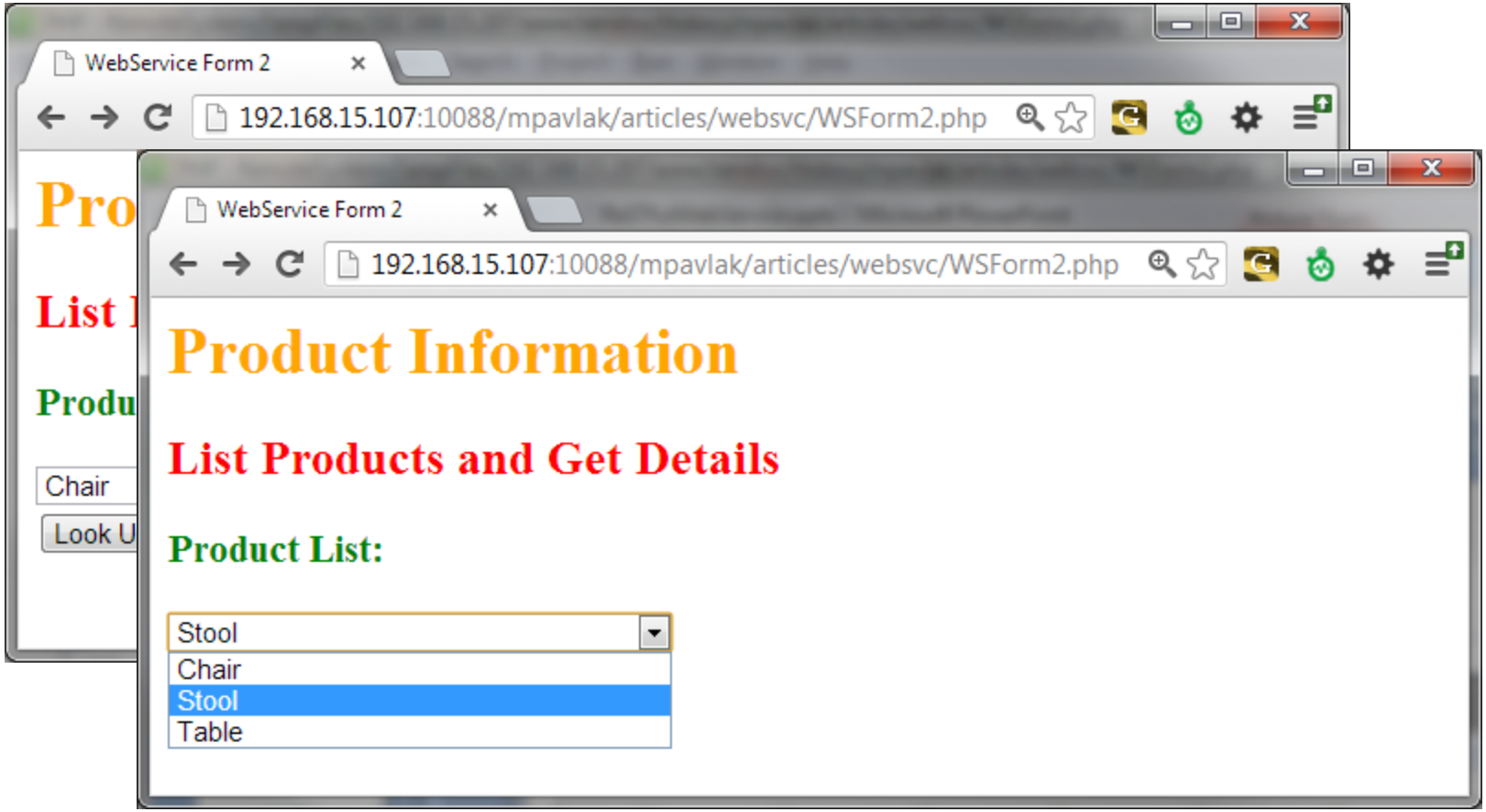
Local & remote both IBM i for this example

Local system can be ANY platform running PHP

Remote system can also be any platform

- In this example is IBM i
- But must have access to Inventory data
- Inventory availability is in RPG business logic
- Simple example, can be VERY complex!

HTML Form using PHP



Form source

```
1  <html><head><title>WebService Form 2</title></head>
2  <style>
3  h1       { color:orange; margin-bottom:3px; }
4  h2       { color:red;}      h3      { color:green;}
5  .ValText { width:80px; }    .OpText { width:250px; }  </style>
6  <body>
7  <h1>Product Information</h1><h2>List Products and Get Details</h2><pre>
8  <?php  require_once 'WSCommon.php';
9
10 // Call the web service to populate the drop down in the form.  Then call the client to show detail.
11
12 $SERVER_URL = $ProductInfo . '?Function=List';
13     //get the XML package
14 $response = file_get_contents($SERVER_URL); //Send the request
15     //turn it into an object
16 $productsResponse = simplexml_load_string($response);
17     //flip it into an array and process.
18 $response_array = json_decode(json_encode($productsResponse),true);
19
20 ?>
21 </pre>
22 <Form action="WSClient2.php" method="GET">
23     <h3>Product List:</h3>
24     <select class="OpText" name="ProductSelected">
25     <?php
26     foreach ($response_array as $productValue) {
27         print ('<Option value="'. $productValue['prodNum'] . '">'. $productValue['prodName'] . "</Option>\n\t");
28     }
29     ?></select>
30 <br /><Input type="submit" value="Look Up"></Form></body></html>
```

WSCommon.php

Common routines and standard locations

Here I store the location of my services

Easy to adjust and I can scan for common services

I'm using hardcoded IPs, but recommend DNS

```php
<?php

// Common routines among the scripts...

// Formulas array used in drop down and validation...
$Formulas = array(  'AreaRectangle'=>'Area of a Rectangle',
                    'Pythgorean'=>'Length of the hypotenuse of right triangle',
                    'VolumeCone'=>'Volume of a Cone');

$MathFormulaURLLocal = 'http://localhost:10088/mpavlak/articles/websvc/WSServer1.php';
$MathFormulaURLRemote = 'http://192.168.15.112:10088/mpavlak/articles/websvc/WSServer1.php';
$ProductInfo = 'http://192.168.15.112:10088/mpavlak/articles/websvc/WSServer2.php';
```

Target LPAR and host service, part 1

```php
<?php

require_once 'ServerUtils.php';

// Listing function and XML Builder

if ($_GET ['Function'] == 'List') {

    $products = listprod ();

    print '<?xml version="1.0"?>';
    print ("<products>\n") ;

    If (is_array ( $products )) {
        foreach ( $products as $prod_array ) {
            $productNumber = $prod_array ['PRODUCTNUMBER'];
            $productName = $prod_array ['PRODUCTNAME'];
            print ("<prod$productNumber>") ;
            print ("<prodNum>$productNumber</prodNum><prodName>$productName</prodName></prod$productNumber>\n") ;
        }
        print '</products>';
    } else {
        print ("<error>Product List not Available</error>") ;
    }
}
```

Server Utils Part 1

```
39 function listProd() {
40
41     $IBMiLink = Connect();
42     $sql = 'SELECT productnumber, productname FROM zenddata.product';
43     //$sql = 'SELECT CUSTOMER_NUMBER, CUSTOMER_NAME FROM zenddata.customer';
44     $stmt = getdata($IBMiLink,$sql);
45     while($row=db2_fetch_assoc($stmt)) {
46         //print_r($row); echo "<BR>";
47         $products[] = ($row);
48     }
49
50     if (is_array($products))
51         return $products;
52     else
53         return "Error retrieving products";
54 }
```

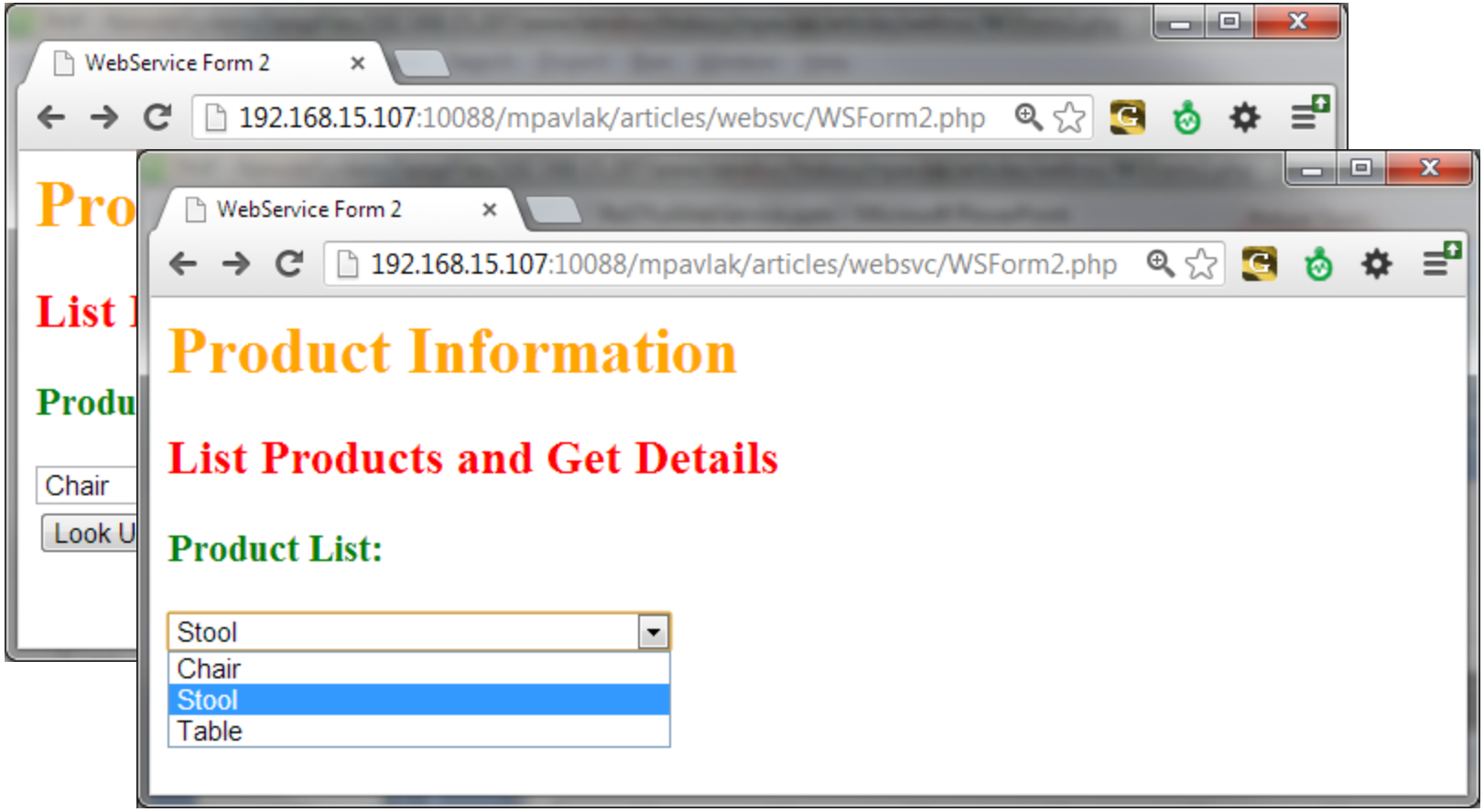
Call RPG using ReST

Remote service calls RPG

Review

- Created form to display products
- Products available listed on remote system
- Have product ID
- Now look up remote system for availability

Select product from form



Form source

```
1  <html><head><title>WebService Form 2</title></head>
2  <style>
3  h1       { color:orange; margin-bottom:3px; }
4  h2       { color:red;}      h3      { color:green;}
5  .ValText { width:80px; }    .OpText { width:250px; }  </style>
6  <body>
7  <h1>Product Information</h1><h2>List Products and Get Details</h2><pre>
8  <?php  require_once 'WSCommon.php';
9
10 // Call the web service to populate the drop down in the form.  Then call the client to show detail.
11
12 $SERVER_URL = $ProductInfo . '?Function=List';
13     //get the XML package
14 $response = file_get_contents($SERVER_URL); //Send the request
15     //turn it into an object
16 $productsResponse = simplexml_load_string($response);
17     //flip it into an array and process.
18 $response_array = json_decode(json_encode($productsResponse),true);
19
20 ?>
21 </pre>
22 <Form action="WSClient2.php" method="GET">
23     <h3>Product List:</h3>
24     <select class="OpText" name="ProductSelected">
25     <?php
26     foreach ($response_array as $productValue) {
27         print ('<Option value="'. $productValue['prodNum'] . '">'. $productValue['prodName'] . "</Option>\n\t");
28     }
29     ?></select>
30 <br /><Input type="submit" value="Look Up"></Form></body></html>
```

Now call WSClient2.php

```
1  <html><head><title>Formla web service</title>
2  <style>
3  h1      { color:orange; margin-bottom:3px; }
4  h2      { color:red;}
5  h3      { color:green;}
6  pre { background-color:lightgreen; padding:5px; border:10px solid red; width:250px }
7  </style></head><body>
8  <h1>On Hand Quantity</h1>
9  <?php
10 require_once("WSCommon.php");
11
12 //Build request
13 $prod = $_GET['ProductSelected'];
14
15 // Call the web service to determine on hand quantity or error.
16 $SERVER_URL = $ProductInfo . "?Function=onHand&Prod=$prod";
17
18 //get the XML package
19 $response = file_get_contents($SERVER_URL); //Send the request
20 //turn it into an object
21 $productsResponse = simplexml_load_string($response);
22 //flip it into an array and process.
23 $response_array = json_decode(json_encode($productsResponse),true);
24
25 print "here is the reponse: "; print_r($response_array). "<br /><br />";
26
27 $xml = simplexml_load_string($response); //Parse the response and display
28 echo '<br/><br />' . $xml->Parm1;
29 ?><b><a href="WSForm2.php">Start Over</a></b></body></html>
```

Target LPAR and host service, part 2

```php
function listOnHand($prod) {
    $sess = getSessions ();
    // Call the open source toolkit for product info...

    $db = 'V7R1GOLD';   $user = '';  $pass = '';  $extension = 'ibm_db2';  $ohQty = 0;  $error = '';

    $ToolkitServiceObj = ToolkitService::getInstance ( $db, $user, $pass, $extension );
    $ToolkitServiceObj->setToolkitServiceParams ( array (
            'InternalKey' => "/tmp/$sess"
    ) );

    $param [] = $ToolkitServiceObj->AddParameterPackDec ( 'both', 9, 0, 'PRODUCT', 'PRODUCT', $prod );
    $param [] = $ToolkitServiceObj->AddParameterPackDec ( 'both', 10, 0, 'ONHQTY', 'ONHQTY', $ohQty );
    $param [] = $ToolkitServiceObj->AddParameterChar ( 'both', 256, 'ERROR', 'ERROR', $error );
    $result = $ToolkitServiceObj->CLCommand ( "RMVLIBLE ZENDDATA" );
    $result = $ToolkitServiceObj->CLCommand ( "ADDLIBLE ZENDDATA" );
    $result = $ToolkitServiceObj->PgmCall ( "PRODAVAIL", "ZENDDATA", $param, null, null );

    if ($result) {
        return $result ['io_param'];
    } else
        return "Execution failed.";
    echo 'Hello Mike<br />';

    $ToolkitServiceObj->disconnect ();
}
```

RPG code

```
FFilename++IPEASF.....L.....A.Device+.Keywords++++++++++++++++++++++
FPRODINV   IF   E             K DISK


C      *ENTRY         PLIST
C                     PARM                    Prodin            9 0
C                     PARM                    QtyAvail         10 0
C                     PARM                    Error           256

 /free
     chain Prodin  PRODINVR;
     if %found;
        QtyAvail = ProdONH - ProdALC;
        Error = 'Order Inventory Successful';
     else;
        Error = 'Order Item not found';
     endif;
  *InLR = *On;
```

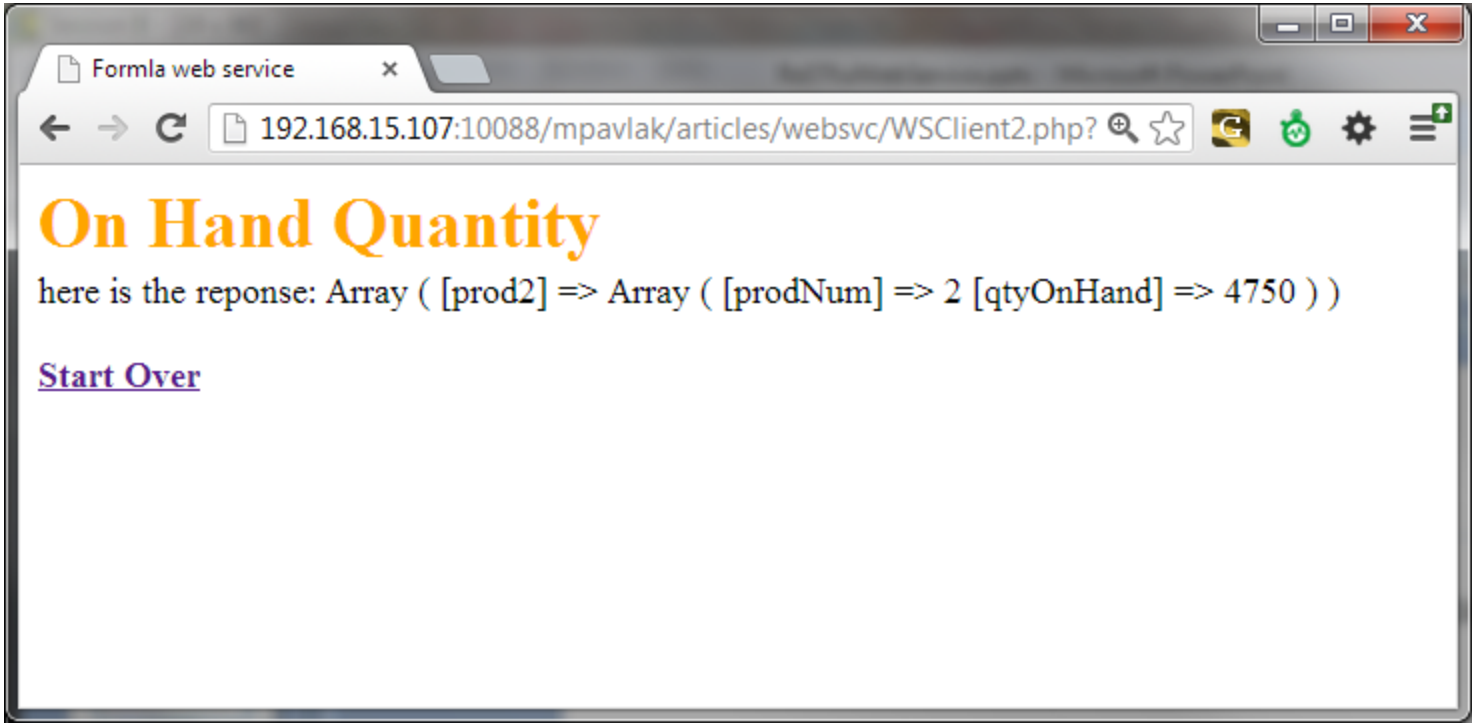
WSClient2.php receieves info and presents

```
<html><head><title>Formla web service</title>
<style>
h1      { color:orange; margin-bottom:3px; }
h2      { color:red;}
h3      { color:green;}
pre { background-color:lightgreen; padding:5px; border:10px solid red; width:250px }
</style></head><body>
<h1>On Hand Quantity</h1>
<?php
require_once("WSCommon.php");

//Build request
$prod = $_GET['ProductSelected'];

// Call the web service to determine on hand quantity or error.
$SERVER_URL = $ProductInfo . "?Function=onHand&Prod=$prod";

//get the XML package
$response = file_get_contents($SERVER_URL); //Send the request
//turn it into an object
$productsResponse = simplexml_load_string($response);
//flip it into an array and process.
$response_array = json_decode(json_encode($productsResponse),true);

print "here is the reponse: "; print_r($response_array). "<br /><br />";

$xml = simplexml_load_string($response); //Parse the response and display
echo '<br/><br />' . $xml->Parm1;
?><b><a href="WSForm2.php">Start Over</a></b></body></html>
```

Final output

Not pretty, but would be used in finished application

Shows on hand quantity for product selected



Feature article at iProDeveloper

With your subscription

Can read article and download code

Wrap it up!

zend

Q&A

www.zend.com

mike.p@zend.com