The PHP Company

PHP on IBM i
RPG Examples

Mike Pavlak

Solution Consultant

Mike.p@zend.com

Agenda

- Intro to PHP Open Source Toolkit
- RPG Program Calls
- Stored Procedures
- Q&A

Zend Server for IBM i

Key Features

- Z-Ray (Like Firebug for PHP)
- Server Dashboards
- Code libraries
- Application Deployment
- Much more!

http://192.168.15.107:10080/mpavlak/sessions/spreadsheet/spreadsheet01.php

f(x) Functions — Showing 1-16/16 — 20

Show (Manage Function Groups)

All	(16)	65.73 ms
(Custom)	(9)	49.22 ms
(PHP Native)	(7)	16.51 ms

Function	Count	Duration Inclusive (ms)	Duration Exclusive (ms)	Average (ms)	Defined at
{main}	1	66.50	8.63	66.50	
getData()	1	21.50	0.03	21.50	Connect.php:19
db2_exec()	1	21.24	21.24	21.24	
connect()	1	17.61	0.03	17.61	Connect.php:4
db2_connect()	1	17.54	17.54	17.54	

200 spreadsheet01.php | 69 ms / 168 KB | 0 | 0 | 1 | f(x) 16 | {} Variables | zend server

Zend Server for IBM i Editions

Basic

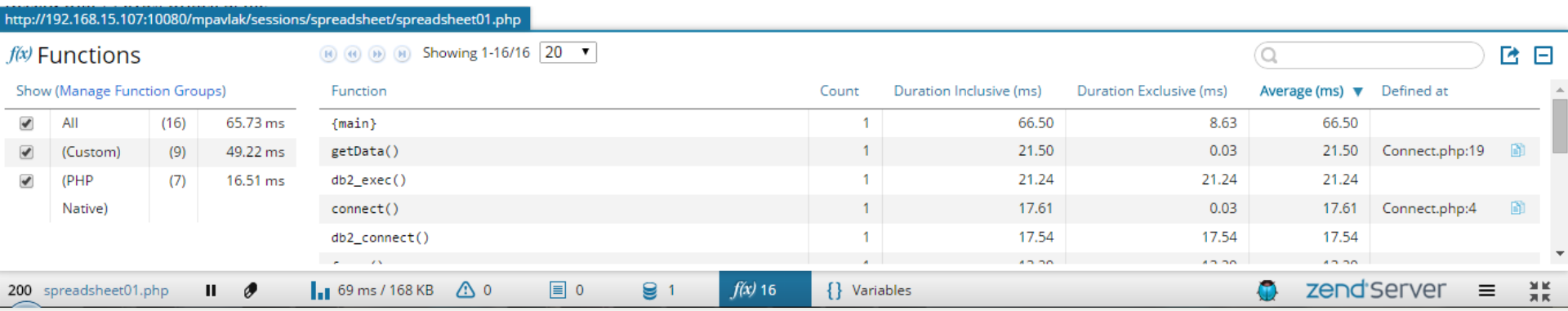
- Included with IBM i SWMA

Professional

- Business hour support and tools

Enterprise

- 24/7 support and more tools



PHP101

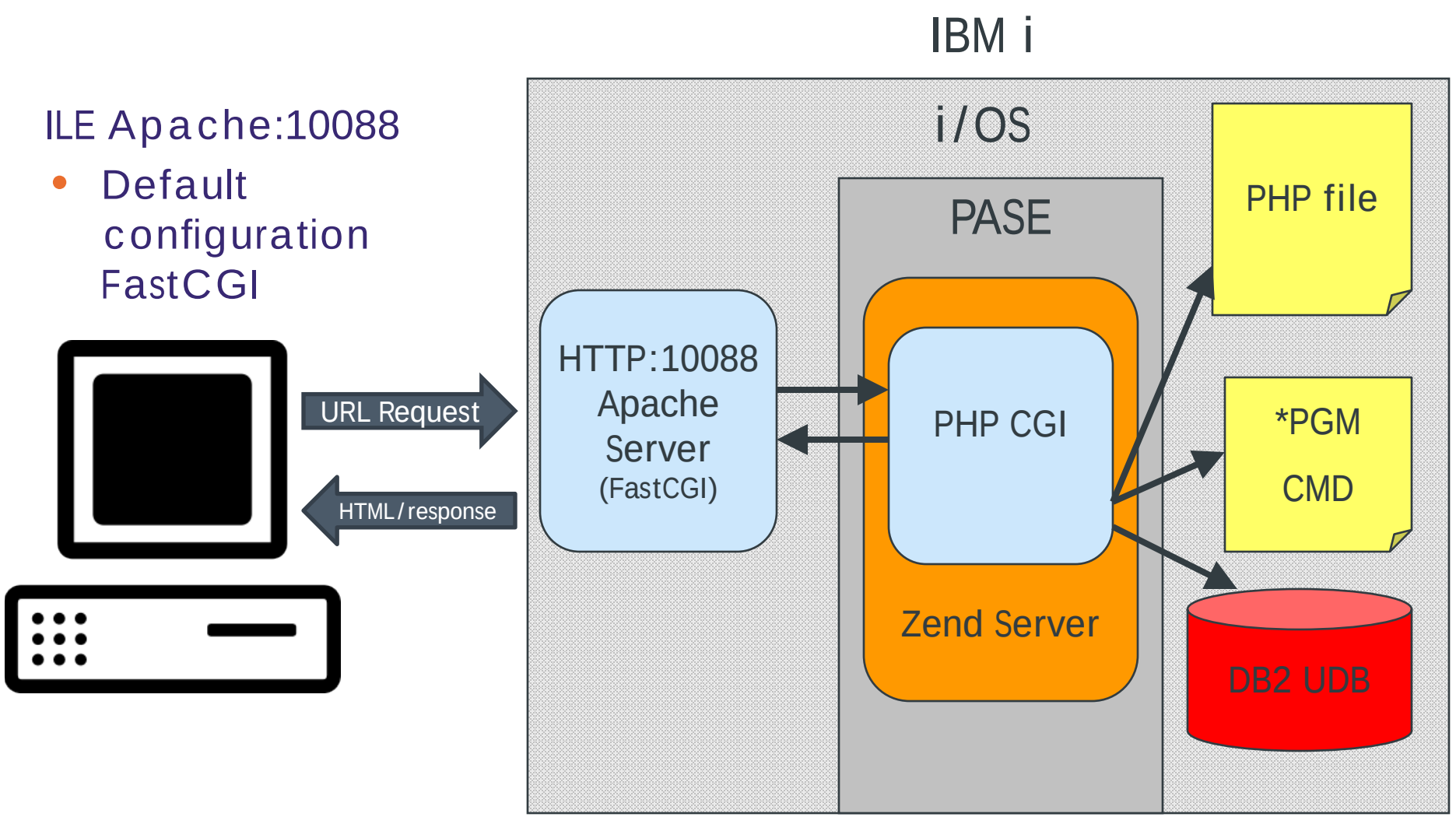
Introduction to PHP

zend

Zend Server under the covers

ILE Apache:10088

- Default configuration FastCGI



PHP Script Structure

- **<?php begins block of PHP code**
  - Either in HTML file or alone
  - Optionally terminate using ?>
  - Semicolon terminates simple statements
  - May embed multiple blocks of PHP code in single file
  - Can string multiple commands in single line

Rules for variables

- **Case sensitive**
- **Must begin with dollar sign “$”**
  - $thisIsMyVariable
  - $_AnotherVariable
  - $ this is not a variable
- **Implicit casting**
- **Can be re-typed (Dynamic typing)**
- **Constant – Value established in script that does not change**
  - Define(‘SPEAKER’, “Mike Pavlak”);

```
<?php

$field1 = 5;
$field2 = 10.3;
$field3 = $field1 + $field2;

?>
```

The 8 Data Types

- **Scalar**
  - Integer
    - -2,147,483,648 thru 2,147,483,647
    - Supports decimal, octal and hex representation
  - Floating-Point
    - 1.7E-308 thru 1.7E+308
    - 15 digits of decimal precision
  - Strings
    - Big. Really big. Too big to discuss!
  - Boolean
    - False is 0, 0.0, false keyword, empty string, object w/no values, null. All others are true
- **Object, array, null and resource**

Arrays: Value Assingment

- **Simple**
  - $Animals[0] = ‘Dog’
  - $Animals[1] = ‘Cat’
  - $Animals[2] = ‘Hamster’
- **Associative**
  - $Barnyard[‘Cow’] = ‘Calf’
  - $Barnyard[‘Chicken’] = ‘Chick’
  - $Barnyard[‘Horse’] = ‘Foal’
- **Multi-dimensional**
  - $farm[0] = $Animals
  - $farm[1] = $Barnyard

Strings

- **Most of PHP is character strings**
- **Single quotes**
  - Variables not expanded
  - Heavy use of concatenation “.”
  - $string_1 = ‘This is the value of variable x: ’ . $x
- **Double quotes**
  - Variables interpolation
  - $string_1 = “This is the value of variable x: $x”

Comments

- **A brief comment about comments**
- ***//* (C++ style)**
  - indicates single line comment
  - May use at end of line of live code
  - Easily comments a line of code
- ***/* … */* (C Style)**
  - Looks like CL? Pretty close
  - Comment block,
  - Can span multiple lines
- **Shell style #**

How many equal signs?

- **= Single equal sign is assignment**
  - $X=3
  - $Y=$X
  - Now $Y = 3
- **== Double equal sign is for conditions**
  - If ($x==$y) { do something} else {do something else}
  - If you see single equal in condition, assignment will occur
- **=== Triple equal sign is exact equal conditions**
  - $X= 3 (Integer) $Y=3.0 (Float)
  - If ($x==$y) will resolve to true
  - If ($x===$y) will resolve to false

Flow Control

- **If (condition is always in parenthesis)**
  - Work
- **Else**
  - Work

```
11  //One line
12  if ($x==5)
13      $x++;
14  else $x--;
```

```
16  //Multi line
17  if ($x==5) {
18      $x++;
19      $y++;
20  }
21  else {
22      $x--;
23      $y--;
24  }
```

Flow control...While's...

- **While (condition) {**
  - Do something
- **}**

```
72  while ($x < $y){
73      echo $x;
74      $x++;
75  }
```

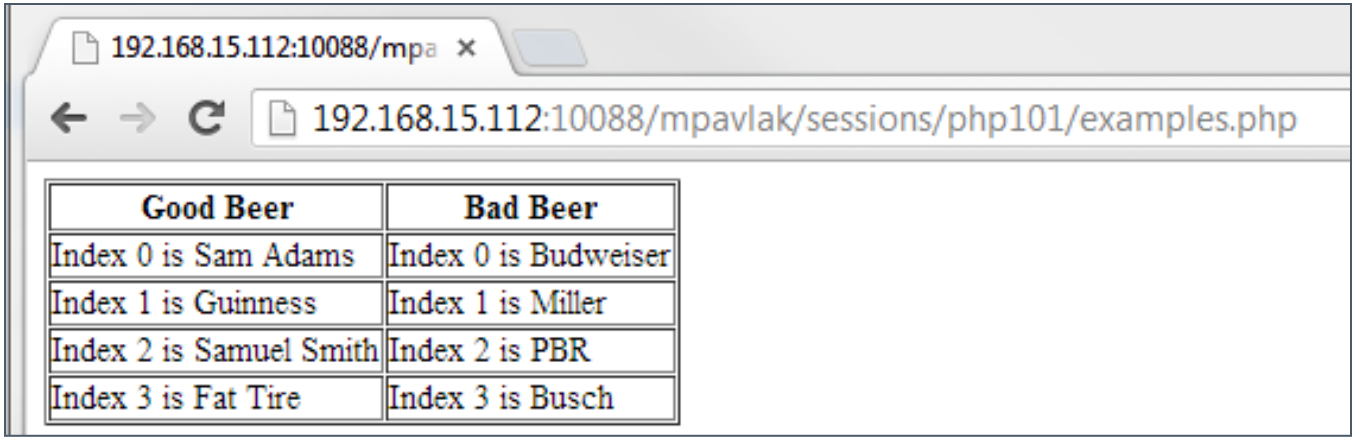
- **Do {**
  - Something
- **} while ($x<$y)**

```
78  do {
79      echo $x;
80      $x++;
81  } while ($x<$y)
```

Foreach

- **Jon's favorite function!**

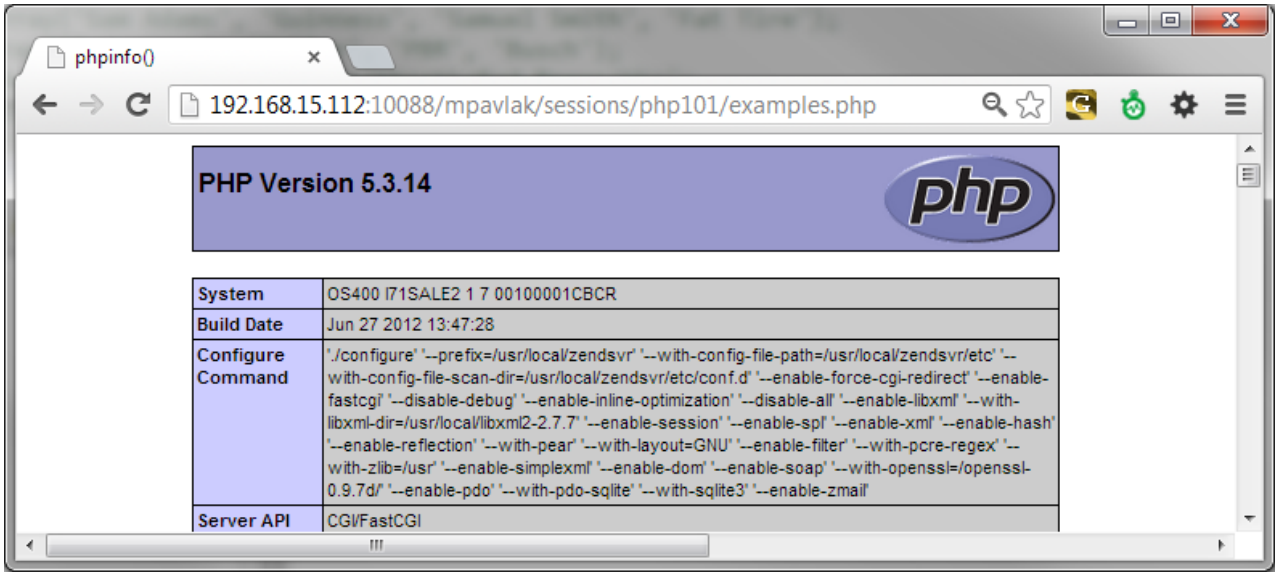
```
87 $goodBeer = array('Sam Adams', 'Guinness', 'Samuel Smith', 'Fat Tire');
88 $badBeer = array('Budweiser', 'Miller', 'PBR', 'Busch');
89 echo '<table border="2"><th>Good Beer</th><th>Bad Beer</th>';
90 foreach ($goodBeer as $beerkey=>$beerval) {
91     echo "<tr><td>Index $beerkey is $beerval</td>";
92     echo "<td>Index $beerkey is {$badBeer[$beerkey]}</td></tr>";
93 }
94 echo '</table>';
```



Most of PHP is functional

- **Like sub procedures…it you know them you got it!**
- **Loosely like subroutines, but not quite**
  - Pass parameters
  - Organize work
  - Get results

```
100 <?php
101
102 phpinfo();
103
104 ?>
```



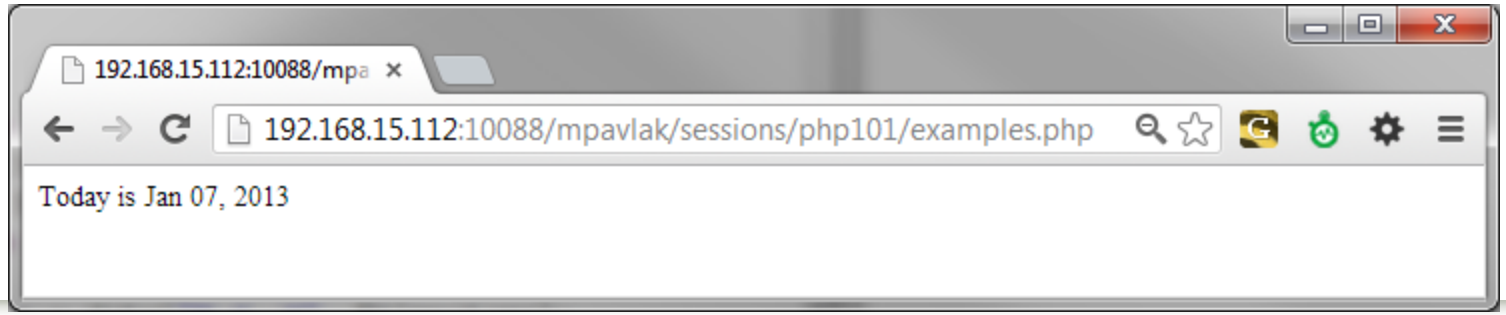
Functions come from three places

- **Built in**
  - Part of core PHP
  - Available everywhere
  - phpinfo(); strtoupper(); array();
- **Extensions**
  - Usually included with PHP
  - Requires local Compiler, tec.
  - DB2_connect; odbc_fetch_row();
- **User created**
  - Like sub procedures

Function is like sub-procedure

- Code it first, then use (not a hard rule)

```php
function getMikeyDate($date='today') {
    if ($date == 'today')
        $timestamp = time();
    else $timestamp=$date;

    $date = date("M d, Y", $timestamp);
    return $date;
}

$today = GetMikeyDate();
echo "Today is $today";
```



PHP101

Database Listing

zend

Employee list with payroll

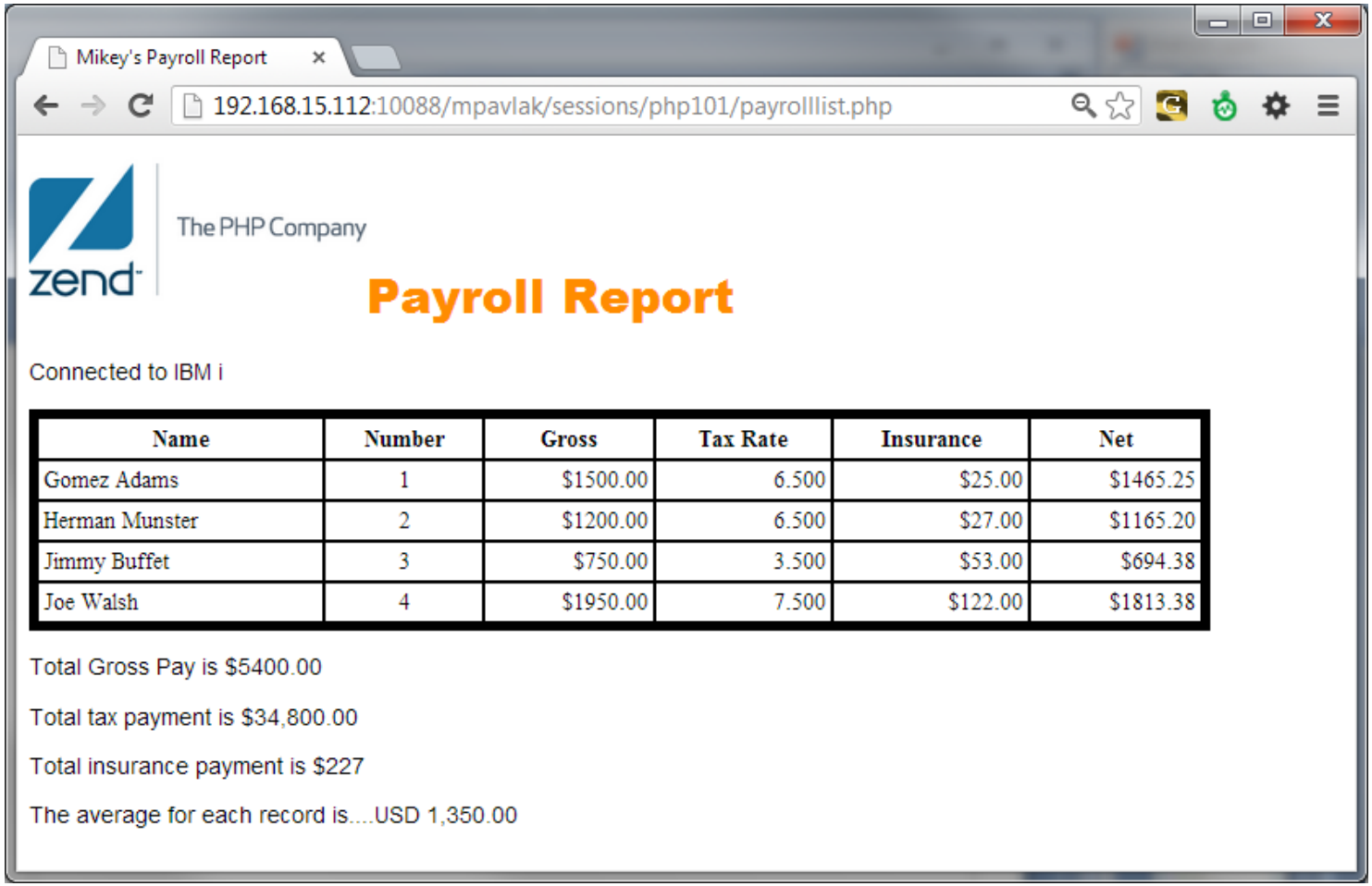
- Describe the file
- Explore database functions
- Perform some calculations
- Put the output to the web server

Data looks like….

- Raw data in STRSQL display…
- Using long field names as both long and short are available to PHP via SQL.

EMPLOYEE_NAME	EMPNUM	EMPLOYEE_WEEK_PAY	EMPLOYEE_TAX_RATE	EMPINS
Gomez Adams	1	1,500.00	6.500	25.00
Herman Munster	2	1,200.00	6.500	27.00
Jimmy Buffet	3	750.00	3.500	53.00
Joe Walsh	4	1,950.00	7.500	122.00
******** End of data ********				

Payroll Master output…



First page of code

```
<html><head><title>Mikey's Payroll Report</title>
<style type="text/css" media=screen>
   p {font-family: arial, serif}
   h1 {font-family: "arial black", serif;color: #FF8C00}</style></head>
<body bgcolor='#FFFFFF' text = '#00000'>
<h1><img src='zend_logo.gif'>Payroll Report</h1>

<?php

//define some variables for database connection
$conn = "V7R1GOLD"; //database name of IBM i

//connect to i5 DB2
$db2Connection = db2_connect($conn, "", "");  //Notice no user id or password
if ($db2Connection) echo "<p>Connected to IBM i</p>";
else {
    echo "<p>Connection failed: ".db2_stmt_error()." : ".db2_stmt_errormsg()."</p>";
    exit;
}
//prepare sql statement
$sql = "SELECT * from zenddata.payroll_master";

//execute sql statement to retrive the data
$resultSet = db2_exec($db2Connection,$sql)
 or die("<p>Failed query:".db2_stmt_error().":".db2_stmt_errormsg()."</p>");
```

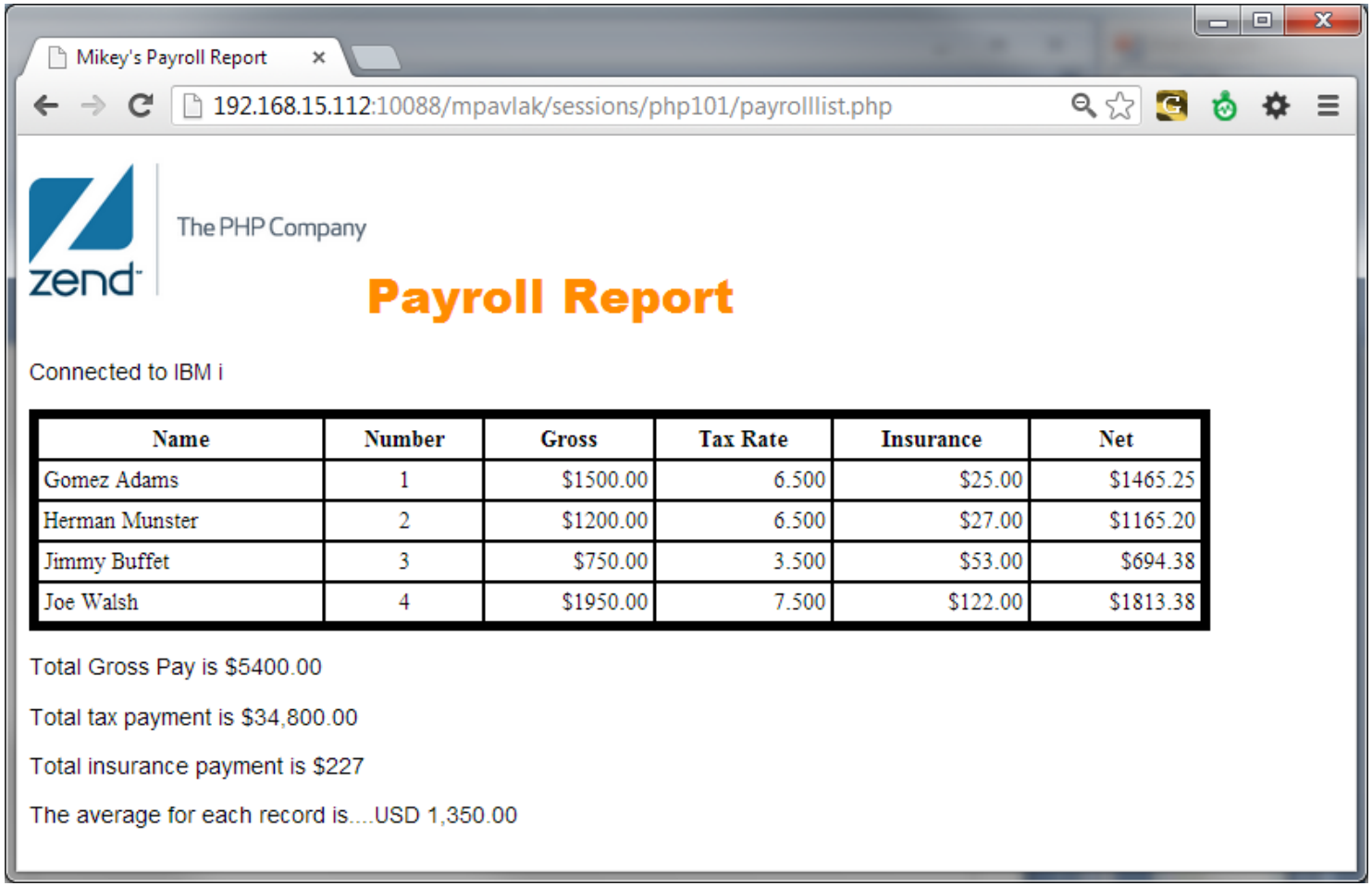
Second page of code...

```
 //begin output table content
echo "<table border='6' cellpadding=4 cellspacing=0 bordercolor='#000000' width='90%'>";
echo "<th>Name</th> <th>Number</th><th>Gross</th><th>Tax Rate</th><th>Insurance</th><th>Net</th>";

$Count = 0;
//fetch the data from each record and print them out
while($row=db2_fetch_array($resultSet)){
    //retrieve the fields from a row
    list( $EMP_NAME, $EMP_NUM, $WEEKLY_PAY, $TAX, $INSURANCE)= $row;
    $Net= $WEEKLY_PAY - ($WEEKLY_PAY * .001 * $TAX) - $INSURANCE;
    echo("<tr><td>$EMP_NAME</td> <td align='center'>$EMP_NUM</td>
    <td align='right'>$$WEEKLY_PAY</td>
    <td align='right'>$TAX</td><td align='right'>$$INSURANCE</td><td align='right'>$");

    printf("%.2f",$Net);            echo("</td></tr>");

    $Total_Gross = $Total_Gross + $WEEKLY_PAY;   //accumulate the total Gross
    $Total_Net = $Total_Net + $Net;   //accumulate the total Net
    $Total_Ins = $Total_Ins + $INSURANCE;  //accumulate the insurance
    $Total_Tax = $Total_Tax + $WEEKLY_PAY * $TAX;  //accumulate the tax
    ++$Count;  //count the total number of employees
}
echo "</table>"; //end of table output
```

Last page of code...

```
53 //print out totals
54 echo("<p>Total Gross Pay is $");
55      printf('%.2f',$Total_Gross);
56 echo ("</p>");
57
58 echo("<p>Total tax payment is $");
59 echo number_format($Total_Tax, 2, '.', ',');
60 echo ("</p>");
61
62 echo("<p>Total insurance payment is $$Total_Ins </p>");
63
64 $Average= round($Total_Gross / $Count, 2); // Average Pay Rate
65
66 setlocale(LC_MONETARY, 'en_US');
67 echo("<p>The average for each record is....");
68 echo    money_format('%i',$Average);
69 echo ("</p>");
70
71 db2_close($db2Connection);   //close the database connection
72
73 ?>
74 </body></html>
```

Payroll Master output…



RPG Examples

www.zend.com

Open Source Toolkit

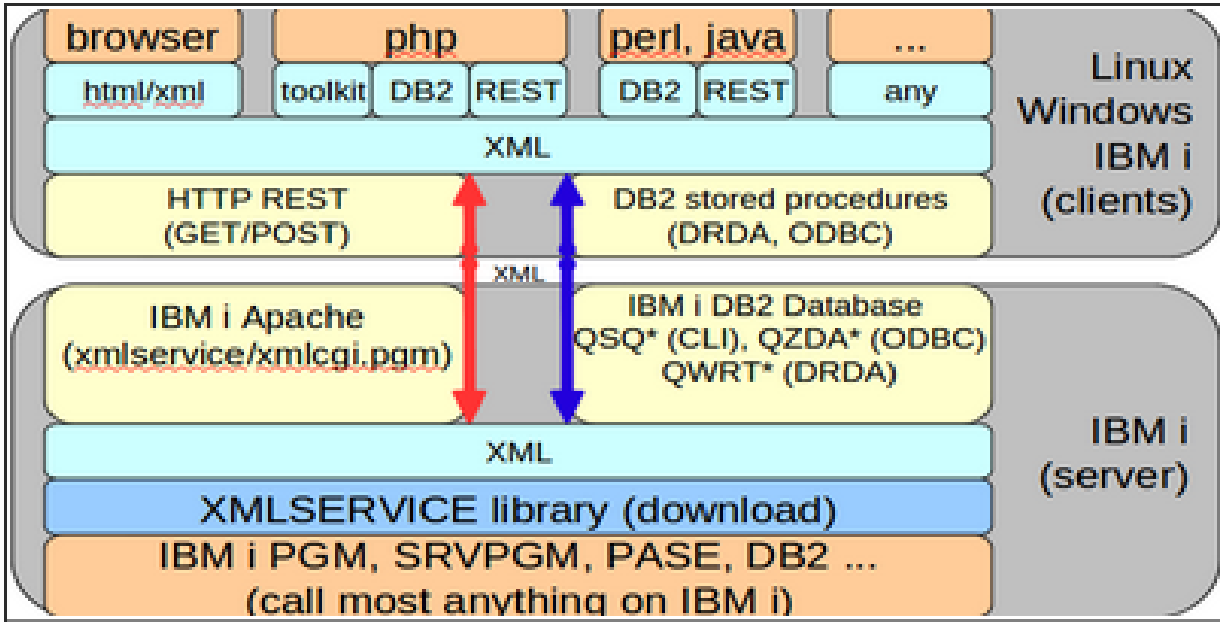
zend

What is the toolkit?

- **Set of classes that access IBM i native artifacts**
- **All Program Objects**
  - RPG, COBOL, CL, etc.
- **Others**
  - Command processor
  - Data Queue's
  - Spooled File
  - More…
- **Access naturally from PHP code. No SP's required**
- **Easy to use**

XML Service

- Cross platform
- Language agnostic
- Basic plumbing for all open source languages



PHP Classes

- **Class where all wrapper functions start**
  - Yes, this is OO but not too bad…
  - Look at it and poke around, It's OK!
  - This is why we need the path set correctly

```
include_once 'ToolkitService.php';
```

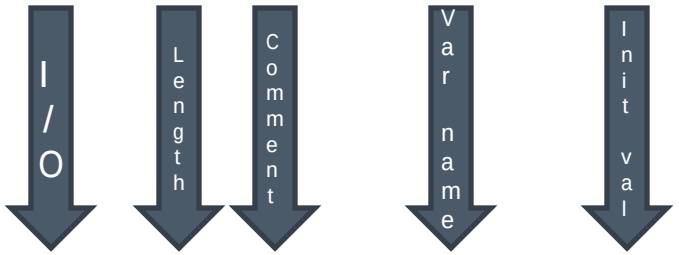
Housekeeping…

- **Need to set the extension and instantiate the object**
  - Singleton pattern
  - Try & Catch
  - Can put this in your own include

```
21 $extension='ibm_db2';
22 try {
23     $ToolkitServiceObj = ToolkitService::getInstance($db, $user, $pass, $extension);
24     }
25
26     catch (Exception $e)
27     {
28         echo  $e->getMessage(), "\n";
29         exit();
30     }
31
32
33 $ToolkitServiceObj->setToolkitServiceParams(array('InternalKey'=>"/tmp/$user"));
34 $code =  $_POST ['code'];
35 $desc = ' ';
```

Now the program call

- **Set parameters based on function**
  - Call program
  - Output results




```
37 $param[] = $ToolkitServiceObj->AddParameterChar('both', 10,'CODE', 'CODE', $code);
38 $param[] = $ToolkitServiceObj->AddParameterChar('both', 10,'DESC', 'DESC', $desc);
39
40 $result = $ToolkitServiceObj->PgmCall("COMMONPGM", "ZENDSVR", $param, null, null);
41
42 if($result){
43   showTable( $result['io_param']);
44 }
45 else
46     echo "Execution failed.";
```

Additional Parameter Types

- AddParameterInt32($io, $comment, $varName, $value, $dimension)
- AddParameterInt64($io, $comment, $varName, $value, $dimension)
- AddParameterUInt32($io, $comment, $varName, $value, $dimension)
- AddParameterUInt64($io, $comment, $varName, $value, $dimension)
- AddParameterFloat ($io, $comment, $varName, $value, $dimension)
- AddParameterReal($io, $comment, $varName, $value, $dimension)
- AddParameterPackdec($io, $length, $scale, $comment, $varName, $value, $dimension)
- AddParameterZoned($io, $length, $scale, $comment, $varName, $value, $dimension)
- AddParameterBin($io, $size, $comment, $varName, $value, $dimension)
- AddDataStruct($array, $parameters, $name='struct_name', $dim, $by, $isArray, $labelLen)

RPG Examples

www.zend.com

Stored Procedure – Record Set

zend

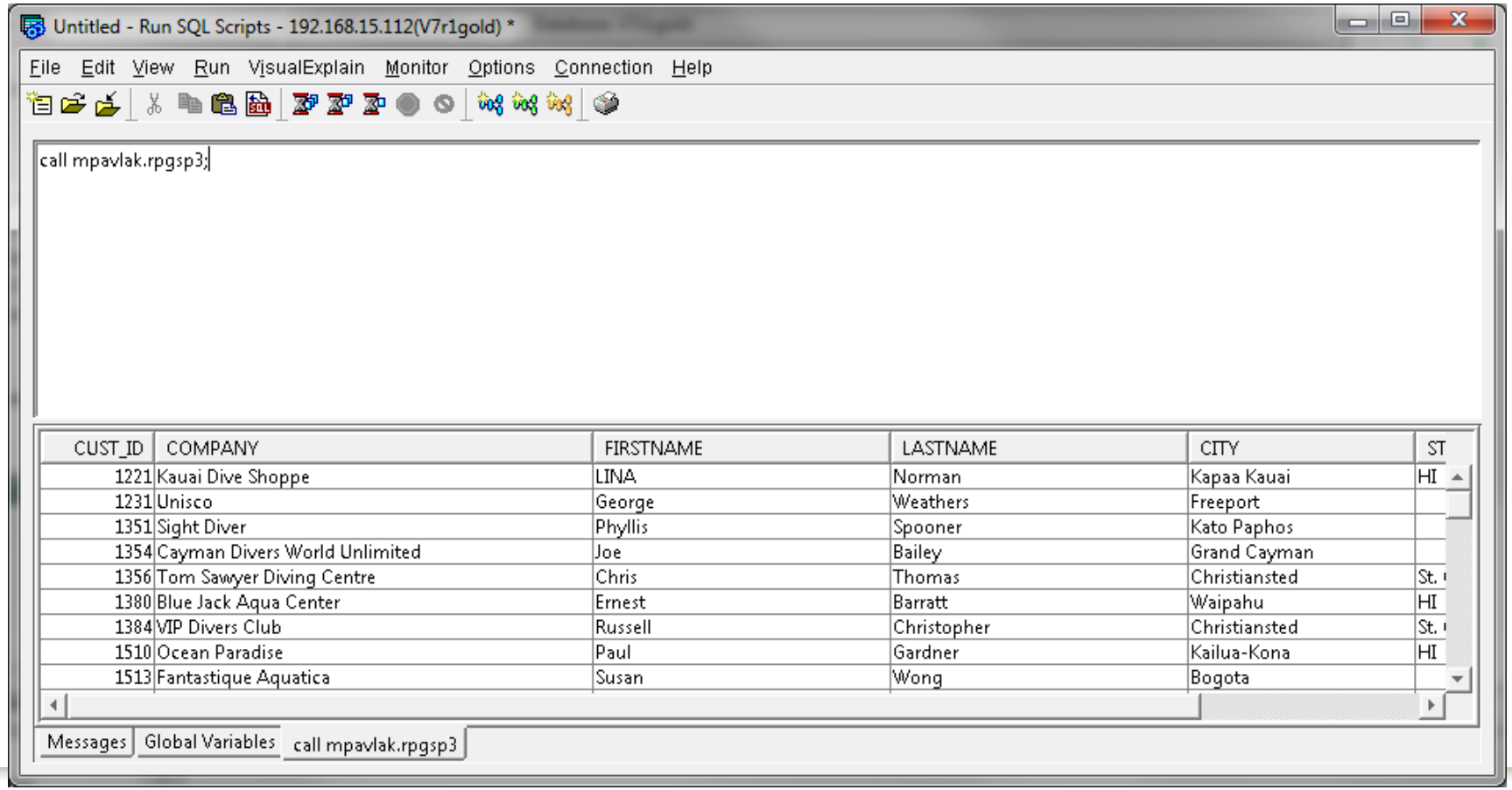
Have RPG build the result set and use SQL…

```
003.00 DRESULT           DS                  OCCURS(20)
004.00 C/EXEC SQL DECLARE C2 CURSOR WITH RETURN TO CLIENT
005.00 C+ FOR SELECT CUST_ID, COMPANY, FIRSTNAME, LASTNAME, CITY, STATE
006.00 C+  FROM zendsvr/sp_cust FOR FETCH ONLY
007.00 C/END-EXEC
008.00 C/EXEC SQL
009.00 C+  OPEN C2
010.00 C/END-EXEC
011.00 C                   SETON                                        LR
012.00 C                   RETURN
```

SQL to create the SP & test in iNav

CREATE PROCEDURE mpavlak/rpgsp3() LANGUAGE RPGLE

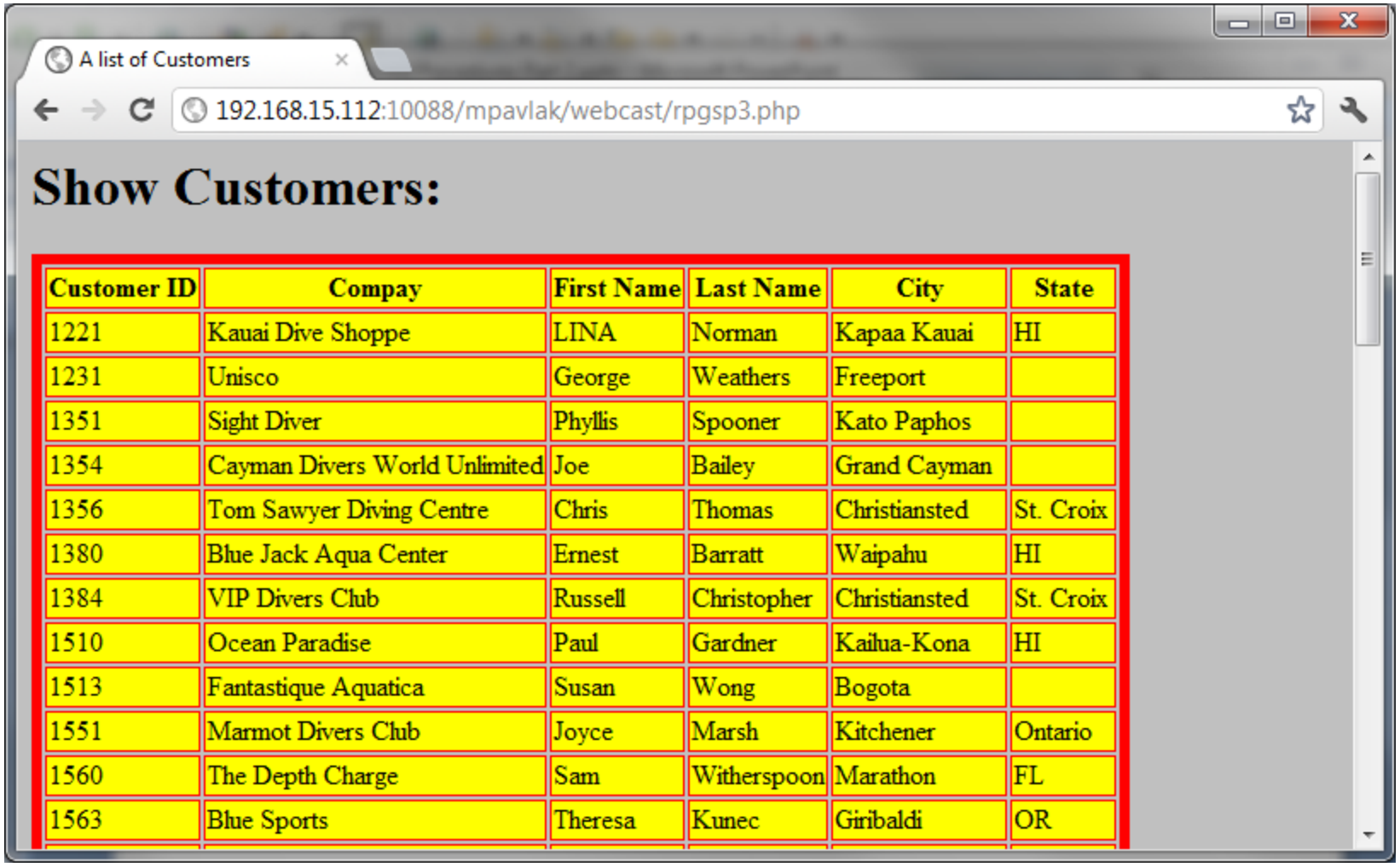
EXTERNAL NAME mpavlak/rpgsp3 GENERAL



Now open results in PHP…

```
20 function GetData($i5link) {
21
22     $sql = "call mpavlak.rpgsp3";
23     $stmt = db2_exec($i5link,$sql)
24         or die("Failed query:".db2_stmt_error().":".db2_stmt_errormsg());
25
26     return $stmt;
27 }
28
29 function listData($stmt) {
30     // loop di loop through customer recs...
31     while($row=db2_fetch_array($stmt)) {
32
33     list ($CustID, $Company, $firstName, $lastName, $city, $state ) = $row;
34     echo("<tr><td> $CustID </td><td> $Company </td>
35             <td> $firstName </td><td> $lastName </td>
36             <td> $city </td><td> $state </td></tr>");
37     }
38 }
```

Results in browser…



Customer ID	Compay	First Name	Last Name	City	State
1221	Kauai Dive Shoppe	LINA	Norman	Kapaa Kauai	HI
1231	Unisco	George	Weathers	Freeport	
1351	Sight Diver	Phyllis	Spooner	Kato Paphos	
1354	Cayman Divers World Unlimited	Joe	Bailey	Grand Cayman	
1356	Tom Sawyer Diving Centre	Chris	Thomas	Christiansted	St. Croix
1380	Blue Jack Aqua Center	Ernest	Barratt	Waipahu	HI
1384	VIP Divers Club	Russell	Christopher	Christiansted	St. Croix
1510	Ocean Paradise	Paul	Gardner	Kailua-Kona	HI
1513	Fantastique Aquatica	Susan	Wong	Bogota	
1551	Marmot Divers Club	Joyce	Marsh	Kitchener	Ontario
1560	The Depth Charge	Sam	Witherspoon	Marathon	FL
1563	Blue Sports	Theresa	Kunec	Giribaldi	OR

Zend Server Update

www.zend.com

Pricing Example

Standard line discount

- RPG is black box, callable from nearly anywhere
- PHP has function to contain RPG call
- Routine to spin through line call for each quantity

Mainline of code

```
73 $quantityArray = [ 27, 35, 56, 77 ];
74
75 $price = 50;
76
77 /* Foreach loop to spin through quantity array... */
78 foreach ( $quantityArray as $quantity ) {
79
80     // calculate discountPrice, here:
81     $discountPrice = calcDiscount ( $quantity, $price );
82
83     // Print Values...
84     print "<tr><td>" . number_format ( $price ) . "</td>";
85     print "<td>" . number_format ( $quantity ) . "</td>";
86     print "<td>$" . number_format ( $discountPrice ['0'] ) . "</td>";
87     print "<td>" . date ( 'M d, Y', strtotime ( $discountPrice ['1'] ) ) . "</td></tr>";
88 }
```

CalcDiscount() Part 1

- Setup session for user-RPG program dedication

```
 8  function calcDiscount($discountQuantity, $Price) {
 9
10      // Setup includes for toolkit classes...
11      require_once 'ToolkitService.php';
12
13      static $PHPToolkitObj;
14
15      if(!isset($_SESSION))
16      {
17          $_SESSION['sessid'] = session_id();
18      }
19      $session = $_SESSION['sessid'];
```

CalcDiscount() Part 2

- **Create object, etc.**

```
21      if (! is_object ( $PHPToolkitObj )) {
22          // Set the authorization values... IBM i credentials...
23          $db = 'SALES1';
24          $user = '';     $pass = '';  $extension = 'ibm_db2';
25          static $PHPToolkitObj;
26
27          // Using try & catch, instantiate the toolkit object...
28          if (! is_object ( $PHPToolkitObj )) {
29              try {
30                  $PHPToolkitObj = ToolkitService::getInstance (
31                          $db, $user, $pass, $extension );
32              } catch ( Exception $e ) {
33                  echo $e->getMessage (), "\n";
34                  exit ();
35              }
36              $PHPToolkitObj->setToolkitServiceParams ( array (
37              'InternalKey' => "/tmp/$session" ) );
38          }
39      }
```

CalcDiscount() Part 3

- Call program

```
45      $qty = $discountQuantity;   $pricein = $Price;  $priceout = 0.0;    $date = '';
46
47      $param [] = $PHPToolkitObj->AddParameterPackDec ( 'both', 9, 0, 'QUANTITY', 'QUANTITY', $qty );
48      $param [] = $PHPToolkitObj->AddParameterPackDec ( 'both', 13, 2, 'PRICEIN', 'PRICEIN', $pricein );
49      $param [] = $PHPToolkitObj->AddParameterPackDec ( 'both', 13, 2, 'PRICEOUT', 'PRICEOUT', $priceout );
50      $param [] = $PHPToolkitObj->AddParameterChar ( 'both', 10, 'DATE', 'DATE', $date );
51
52      // Call the RPG program...
53      $result = $PHPToolkitObj->CLCommand ( "RMVLIBLE ZENDDATA" );
54      $result = $PHPToolkitObj->CLCommand ( "ADDLIBLE ZENDDATA" );
55      $result = $PHPToolkitObj->PgmCall ( "RPGPRICE", "*LIBL", $param, null, null );
56
57      // Print the results...
58      if (isset ( $result ['io_param'] )) {
59          $output = $result ['io_param'];
60          $priceout = $output ['PRICEOUT'];
61          $date = $output ['DATE'];
62      }
63      // echo '</pre>'; print_r($result); echo '</pre>'; exit();
64      return array ( $priceout, $date );
65  }
```

Zend Server Update

www.zend.com

Miscellaneous Toolkit Points

What was passed?

- **Toolkit.ini setting**
  - Debug = TRUE
- **Turns on DebugOutput file**
  - debugLogFile = "/usr/local/zendsvr/share/ToolkitApi/debug.log“
- **Many other directive settings here**

Contents of output

```
Creating new conn with database: '*LOCAL', user or i5 naming flag: '', ext
Going to create a new db connection.
Exec start: 2012-03-15 23:11:13
IPC: '/tmp/ipc_cw_QTMHHTTP_791133185307303002030O'. Control key: *cdata *sb
Stmt: call ZENDSVR.iPLUG4K(?,?,?,?)
Input XML: <?xml version="1.0" encoding="ISO-8859-1" ?><script>
    <pgm name='RPGPRICE' lib='PHPLAB'>
    <parm io='both' comment='QUANTITY'><data var='QUANTITY' type='9p0' vary

    </pgm>
    </script>
Output XML: <?xml version="1.0" encoding="ISO-8859-1" ?><script>
    <pgm name='RPGPRICE' lib='PHPLAB'>
<parm io='both' comment='QUANTITY'>
<data var='QUANTITY' type='9p0' varying='off'><![CDATA[30]]></data>
</parm>
<parm io='both' comment='PRICEIN'>
<data var='PRICEIN' type='13p2' varying='off'><![CDATA[136.00]]></data>
</parm>
<parm io='both' comment='PRICEOUT'>
<data var='PRICEOUT' type='13p2' varying='off'><![CDATA[122.40]]></data>
</parm>
```

Stateful processing

- IPC
- Use session ID, for example
- $PHPToolkitObj->setToolkitServiceParams (array ('InternalKey' => "/tmp/$user"));

Zend Server Update

www.zend.com

Wrap it up

zend

zendcon 2015

October 19-22, 2015 • Las Vegas, NV

Early Bird Available!
Register by
June 15, 2015

Join us at ZendCon

The premier PHP conference!

October 19-22, 2015 – Las Vegas, NV

Conference Themes

PHP in 2015 - *The latest PHP technologies and tools*
Learn how to leverage the latest mobile, HTML 5, testing and PHP best practices, PHP 7 and more

Zend Framework 2 - *Hit the ground running*
Learn how to build faster, more modular and more expandable applications

Development & The Cloud - *A love story*
Learn how the latest developments in cloud-based services, infrastructure and best practices can benefit you

Conference Highlights

- Sessions focused on how to best develop and deploy PHP
- Sessions designed for all knowledge levels
- Intensive tutorials for accelerated learning
- PHP Certification crash courses and testing
- Exhibit hall showcasing the latest products
- Special networking opportunities during meals and events

www.zendcon.com

Q&A

www.zend.com

mike.p@zend.com

Please fill out your Session Evaluation!