FRESCHE SOLUTIONS

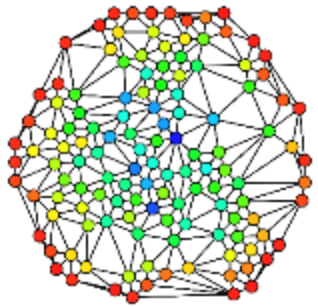
30+ PHP Tips in 60 Minutes

Tip 1 – RTFM – php.net

- Don’t waste money on books that summarize functions
  - Becomes out of date QUICKLY
  - Online manual is free!
  - Books worth buying:
    - Cookbooks
    - Design patterns
    - Advanced PHP, etc
- Open Source community “assumes” you have done this
  - Good questions get answered quickly
  - Stupid questions tend to get ignored or ridiculed
- Install a browser you rarely use (like IE) and set initial page to php.net
- DB2 Extension is documented quite extensively here

Tip #2 – Keep Current

- PHP is NOT RPG.
  - Only RPG supports 40 years of backward compatibility.
- PHP needs updating
  - Security patches (www)
  - New features
  - Deprecated functions



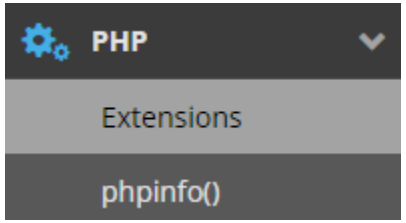
- Long Term Support comes to IBM i community FOR FREE!!!
- PHP is an “edge” technology and these require frequent updating

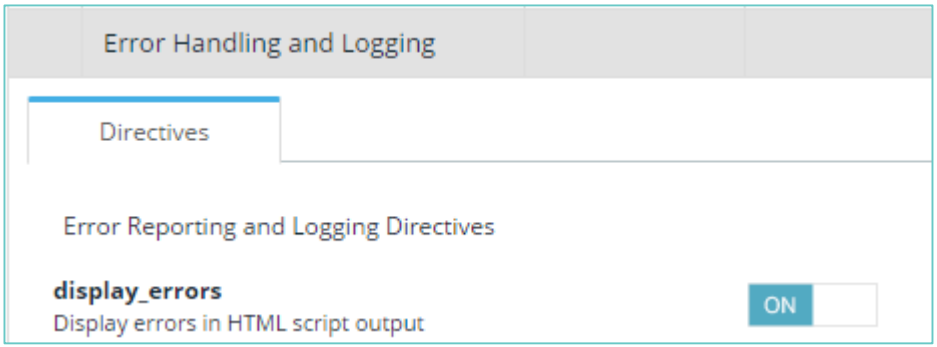
Tip #3 – Master Web Development

- HTML 5
  - The glue or DDS of the web
  - Easy to start
- Cascading Style Sheets
  - Bring organization to site
  - Powerful feature set (csszengarden.com)
- JavaScript
  - Power browser functions
  - Many good code libraries
    - jQuery
    - ExtJS-Sencha
    - Angular

Tip #4 – Develop with display_errors on

- PHP is best suited for iterative development model
  - Code → Save → Refresh → Repeat
- ALSO: Find errors as you go at the script level
  - error_reporting(-1); // report all error levels
  - ini_set('display_errors', '1'); // *show me* the errors
- Installation of Zend Server for Production turns this off
  - COMMON for this to be off in single LPAR environments
  - Get a developer LPAR (later)
- Admin GUI
  - PHP→Extensions →





Tip #5 – Get a Development LPAR

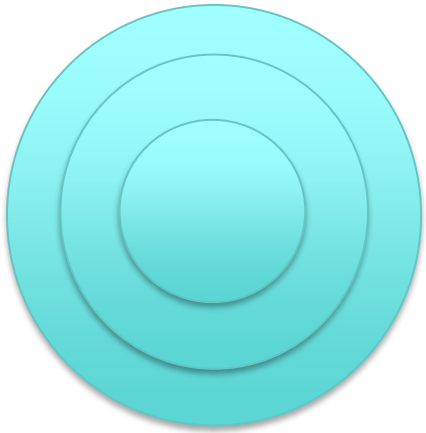
- No more excuses!
  - IBM has made this MUCH easier (i on i)
  - IBM has made this much less expensive
    - Couple thousand, Disk, RAM
- Only one instance of Zend Server supported per LPAR
  - Settings are global (php.ini)
  - Library list can be manipulated, but isolation requires CHROOT, etc.
- New PHP developers don't get having dev & prod on same LPAR
  - "off the street" struggle with one server, let alone libl, etc…
  - Can do dangerous things like issue ALTER TABLE to CUSTMAST
- You are doing more with your IBM i now than ever before
  - "I never had a dev LPAR before" is not a valid excuse
  - Give your devs the freedom to experiment

Tip #6 – Don’t develop PHP like RPG

- RPG applications often are built with main file then chains
  - SETLL then READE ORDDETAIL
    - CHAIN PRODUCT CHAIN INVENT CHAIN ONPLAN
- Consider a single SQL statement
  - Join files are more efficient that chains, in most cases
  - SQL optimizers are genius!
  - Learn good indexing strategy
  - Go to SQL sessions at conferences
  - Ask Db2 guys, they’ll tell you!
- Great Redbooks:
  - Database modernization
  - RPG and Application Modernization

Tip #7 – Bit Twiddling 101

- Consider this VERY carefully, what is the most expensive resource?
- Single quotes vs. double quotes
- Echo vs. Print
- Sprint() instead of “”
- Unset variables for memory management
- Close database connection if you are done unless pconnect
- $row[‘id’] is faster than $row[id];
- Static pages (html) are OK!!!
- Not everything has to be a Framework



Fastest to slowest:

Construct
Function
Static method
Object
Framework

Tip #8 – Understand modern development

- Tools
  - Git
  - SSH
  - HTTP / HTTPS
  - Terminal Usage
- Your role
  - Front / Back / Full stack developer
  - Dev Ops
  - Test / Scrum master

The 2018 Web Developer Roadmap

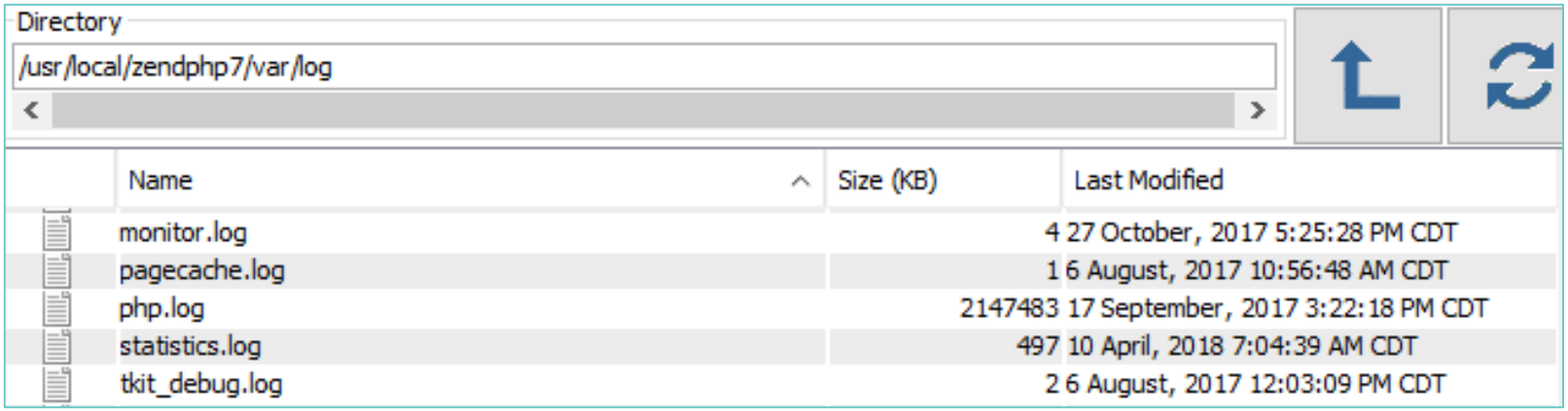
An illustrated guide to becoming a Frontend or Backend Developer with links to courses

https://codeburst.io/the-2018-web-developer-roadmap-826b1b806e8d

Tip #9 - Performance of logs

- PHP logfile captures EVERY PHP error (can be adjusted)
- “clip” the logfile periodically
  - Rename or delete
  - Add this to weekly/monthly CL process
  - PHP just creates a new one when it needs it!
- Don’t forget to read them, too!
  - Notice errors & deprecated functions

Rod Says 1
Rod Says 2

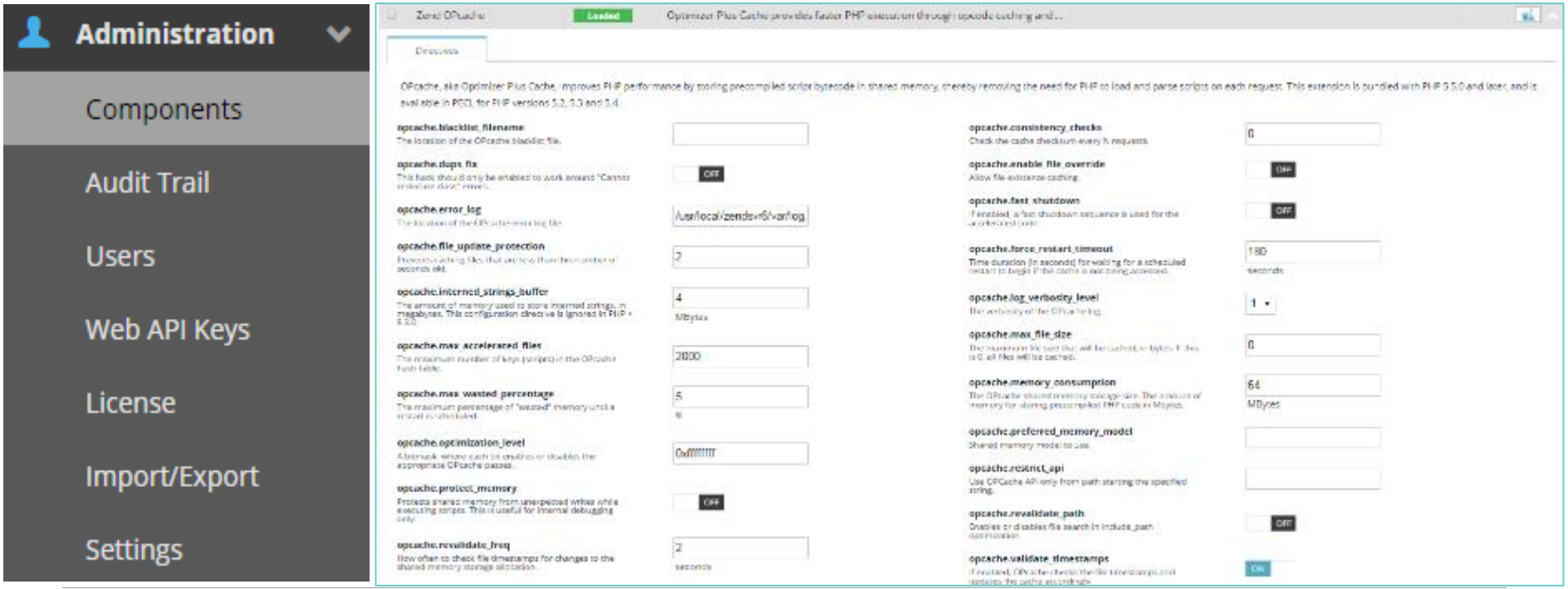


Tip #10 – Get vs. Post

- GET is great for embedded URL
  - Take you directly to a detailed page
  - Routing through application
  - Benign data
- POST
  - Better for more sensitive content
  - Always encrypt
  - Always filter and escape

Tip #11 – Use opcode cache

- PHP is interpreted
- Each script is read from disk, then parsed to opcode, then run
- Opcode cache can keep parsed copy in memory
- All or nothing purge



Tip #12 – Data Cache

- Green screen lookup window vs. drop down
- Data Cache (Zend or APC) saves data in memory
- Performance
- White paper
- Better performance for frequently accessed lists
- Less stress on DB2
- Aggregate values (BI)
- Look at examples…

Standard DB2 Call

```php
function getOrders()
{
    $DB2handle = Connect();

    $sql = 'select * from classicmodels.orders';
    $stmt = db2_exec($DB2handle, $sql);

    if (! $stmt) {
        echo 'Connection failed: ' . db2_stmt_error() . ' : ' . db2_stmt_errormsg();
        return FALSE;
    }
    var_dump($stmt);
    while (($row = db2_fetch_assoc($stmt)) !== FALSE) {
        $orders[] = $row;
    }

    return $orders;
}
```

Function	Count	Duration Inclusive (ms)
{mai		
Conn		
db2_		
getO		
db2_		
db2_		

Function	Count	Duration Inclusive (ms)
2 rows are not displayed		
{main}	1	572.896
getOrders()	1	76.508
db2_exec()	1	27.182
Connect()	1	17.922
db2_connect()	1	17.855
db2_fetch_assoc()	327	16.845

DB2 Call w/Data Cache

```php
function getOrders2()
{
    $orders = zend_shm_cache_fetch('my-orders');

    if ($orders === FALSE) {

        $DB2handle = Connect();

        $sql = 'select * from classicmodels.orders';
        $stmt = db2_exec($DB2handle, $sql);

        if (! $stmt) {
            echo 'Connection failed: ' . db2_stmt_error() . ' : ' . db2_stmt_errormsg();
            return FALSE;
        }
        while (($row = db2_fetch_assoc($stmt)) !== FALSE) {
            $orders[] = $row;
        }

        // Store results...326 orders...
        zend_shm_cache_store('my-orders', $orders, 24 * 3600);
    }
```

Function	Count	Duration Inclusive (ms)
3 rows are not displayed		
{main}	1	533.768
getOrders2()	1	75.485
db2_exec()	1	27.003

Function	Count	Duration Inclusive (ms)
2 rows are not displayed		
{main}	1	427.947
getOrders2()	1	2.088

Tip #13 – That one connection

- PHP can support multiple connections to DB2
  - Also MySQL, MSSQL, Oracle, etc.
- Only one connection required per DB, more costs performance
  - Multiple SQL statements and result sets can share a connection
  - Connection consumes memory, don't waste
  - Connection startup & shutdown
- Sometimes you need another connection
  - If you need different *LIBL
  - Connecting to second partition

Tip #14 – Prepared SQL

- Test in ACS Database View…
- DB2_Exec
- DB2_Execute

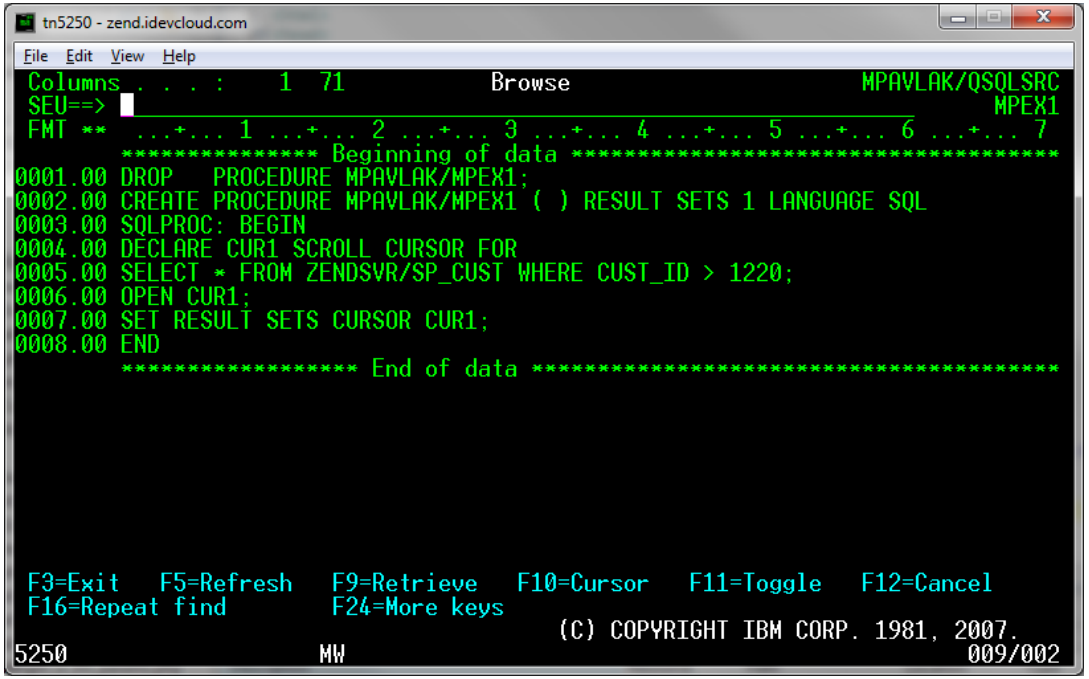
```
23 /* Construct the SQL statement */
24 $sql = "SELECT * FROM ZENDSVR.SP_CUST WHERE CUST_ID > ? FOR FETCH ONLY";
25
26 /* Prepare, bind and execute the DB2 SQL statement */
27 $stmt= db2_prepare($conn_resource, $sql);
28 $lower_limit = 1220; //from the CUST_ID value
```

```
45 //Execute statement , uses a binding of parameters
46 db2_bind_param($stmt, 1, "lower_limit", DB2_PARAM_IN);
47 $result = db2_execute($stmt);
48
49       if (!$result) {
50           echo 'The db2 execute failed. ';
51           echo 'SQLSTATE value: ' . db2_stmt_error();
52           echo ' Message: ' .   db2_stmt_errormsg();
53        }
54        else
55        {
56          while ($row = db2_fetch_array($stmt))
```

Tip #15 – Use Stored Procedures

- The fastest method for accessing data and business logic
- Faster than the toolkit
- Ubiquitous
- If you got ‘em, use ‘em.

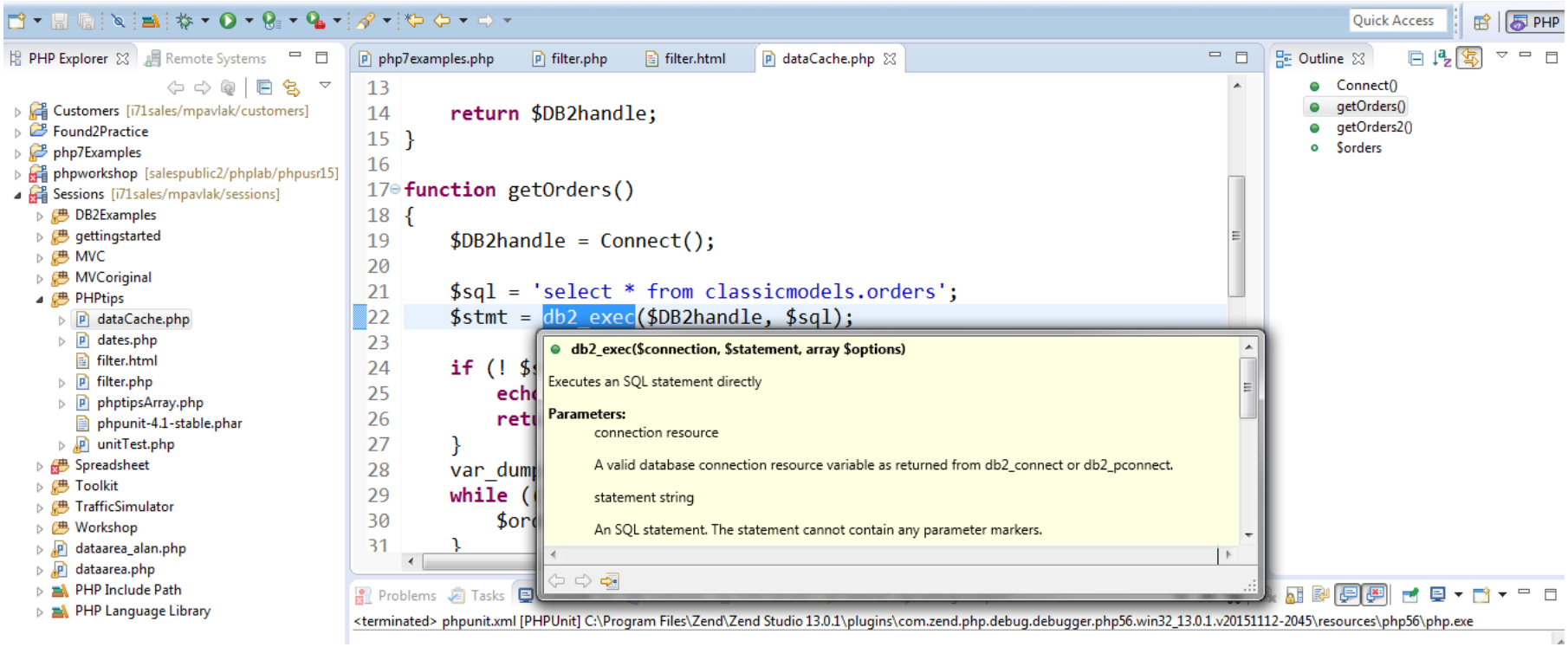
```
CREATE PROCEDURE sales_price(
                        IN Customer# CHAR(8),
                        IN Product# CHAR(8),
                        OUT price DECIMAL(12,2))
LANGUAGE SQL
BEGIN
            SELECT CustPrice            FROM priceTable
                        WHERE CUSTNO=Customer# AND
                         PRODNO=Product# AND
                        active = 'Y'
END;
```



Tip #16 – Use a good editor

- Zend Studio is an IDE
  - Syntax check, Variable prompting (case sensitive)
  - HTML, JavaScript, CSS support
  - Free for first year
- Notepad is painful
- Other editors
  - Notepad++
  - Netbeans
  - PHP Storm

Good editor


PHP Explorer
Remote Systems
Customers [i71sales/mpavlak/customers]
Found2Practice
php7Examples
phpworkshop [salespublic2/phplab/phpusr15]
Sessions [i71sales/mpavlak/sessions]
DB2Examples
gettingstarted
MVC
MVCoriginal
PHPtips
dataCache.php
dates.php
filter.html
filter.php
phptipsArray.php
phpunit-4.1-stable.phar
unitTest.php
Spreadsheet
Toolkit
TrafficSimulator
Workshop
dataarea_alan.php
dataarea.php
PHP Include Path
PHP Language Library
php7examples.php
filter.php
filter.html
dataCache.php
13
14 return $DB2handle;
15 }
16
17 function getOrders()
18 {
19 $DB2handle = Connect();
20
21 $sql = 'select * from classicmodels.orders';
22 $stmt = db2_exec($DB2handle, $sql);
23
24 if (! $
25 echo
26 retu
27 }
28 var_dump
29 while (
30 $ord
31 }
db2_exec($connection, $statement, array $options)
Executes an SQL statement directly
Parameters:
connection resource
A valid database connection resource variable as returned from db2_connect or db2_pconnect.
statement string
An SQL statement. The statement cannot contain any parameter markers.
Outline
Connect()
getOrders()
getOrders2()
$orders
Problems
Tasks
<terminated> phpunit.xml [PHPUnit] C:\Program Files\Zend\Zend Studio 13.0.1\plugins\com.zend.php.debug.debugger.php56.win32_13.0.1.v20151112-2045\resources\php56\php.exe

Tip #17 – System Naming

- Gives scripts a library list like logon screen
- 3 ways to do library list (see other presentation DB2 Examples)

```
function Connect()
{
    // Standard DB connection to DB2...
    $conn = "*LOCAL";
    $name = "";
    $pwd = "";
    $DB2handle = d
    if (! $DB2hand
        echo 'Conn
                                db2_conn_errormsg();

    return $DB2handle
}
```

```
$options = array("i5_lib"=>"ZENDSVR6");

$con1 = db2_connect($db, $user, $pwd, $options);
```

```
$options = array(    'i5_naming' => DB2_I5_NAMING_ON,
                     'i5_libl'=>"MPAVLAK ZENDSVR6 AMAZON");

$con1 = db2_connect($db, $user, $pwd, $options);
if (!$con1) {
    echo 'Error connecting to ' . $db . ' - ' .
                                  db2_conn_error() . ' - ' .
                                  db2_conn_errormsg();

    exit();
}
```

Tip #18 – Global Variables

- Don't use them
  - Globals are the antithesis of functional development
  - They are the GOTO of the variable world
  - Remember RPG III? Yep, you get it…
- Register Globals
  - Security hole
  - Deprecated in 5.4
  - Removed in 5.6
  - Replace with extract()
- Avoid the Global Namespace
  - Prevent collisions

```
3  function formatDate() {
4
5      global $date;
6
7      $formattedDate = date("Y/m/d");
8      return $formattedDate;
9
10 }
11
12 $date = 'Herman';
13
14 $displayDate = formatdate();
```

Tip #19 – Static Typing

- PHP 7 introduces many new features and scalar type hinting is one…
  - http://php.net/manual/en/migration70.php

```php
function add(float $a, float $b) {
    return $a + $b;
}

$returnValue = add(1.5, 2.5); // int(4)
// Works

$returnValue = add("1 foo", "2");
// PHP 5.6 and below gives a Notice
// PHP7 TypeError

$returnValue = add(1, 2); // int(3)
// Widening
```

Static Typing (cont.)

- So is return typing… (Thanks Cal!)

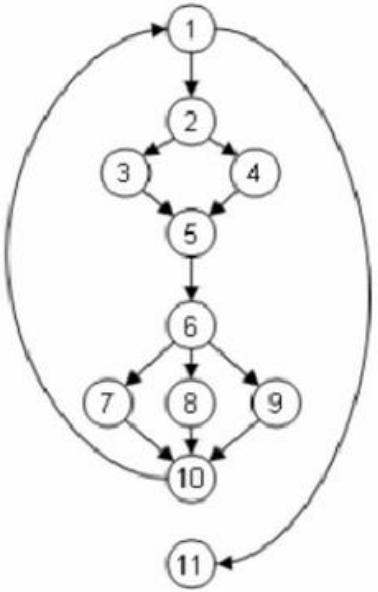
```
function foo(): array {
    return [];
}
```

```
Class MyOtherClass extends MyClass {

    function make(): MyOtherClass
    {
        return new MyOtherClass();
    }
}
```

Tip #20 - Manage Cyclomatic Complexity

- How deeply nested is the logic?
- Impacts all languages
- Gets worse over time (Maintenance)
- Use phploc



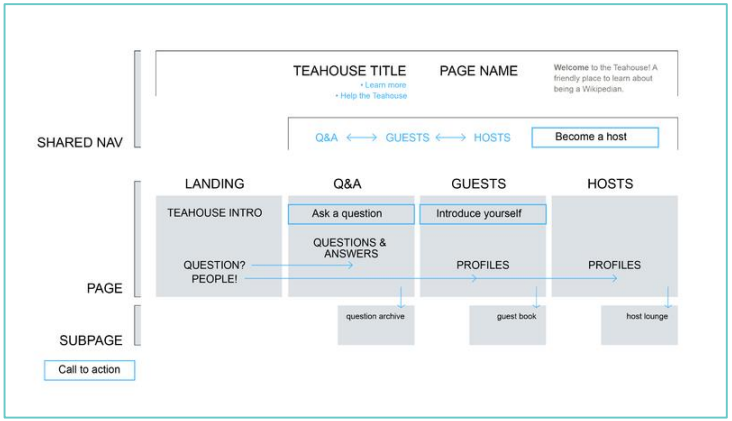
Laravel	1.62
Symfony	1.88
Slim	2.40
Zend	2.76
CakePHP	3.30

*Taylor Ottwell

< 10 Easy to maintain
11-20 Harder to maintain
21+ Candidates for refactoring/redesign

Tip #21 – Agile & Wireframe

- Agile
  - Deliver a little bit, more often
  - Don’t need full blown scrum for agile techniques
- Wireframe
  - Design, mock up app before coding
  - Modern design document

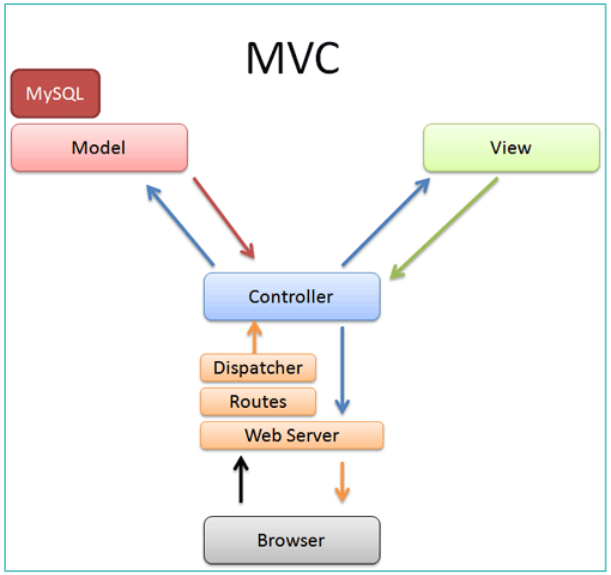


Tips for successful wireframming

- Consistency
- Use actual content, fill dummy boxes with data
  - Tell a story, consider your audience
- Wire in teams (User & Developer)
  - If not, do your homework, spend tie in the user area
- Pick your tools (white board, paper, 3rd party, etc…)
- Practice non-attachment
- Leverage net for examples
- Think flow, not style

Tip #22 – Study Design Patterns

- Foundation of most frameworks
- Developed initially for traffic management
- Supports many methodologies



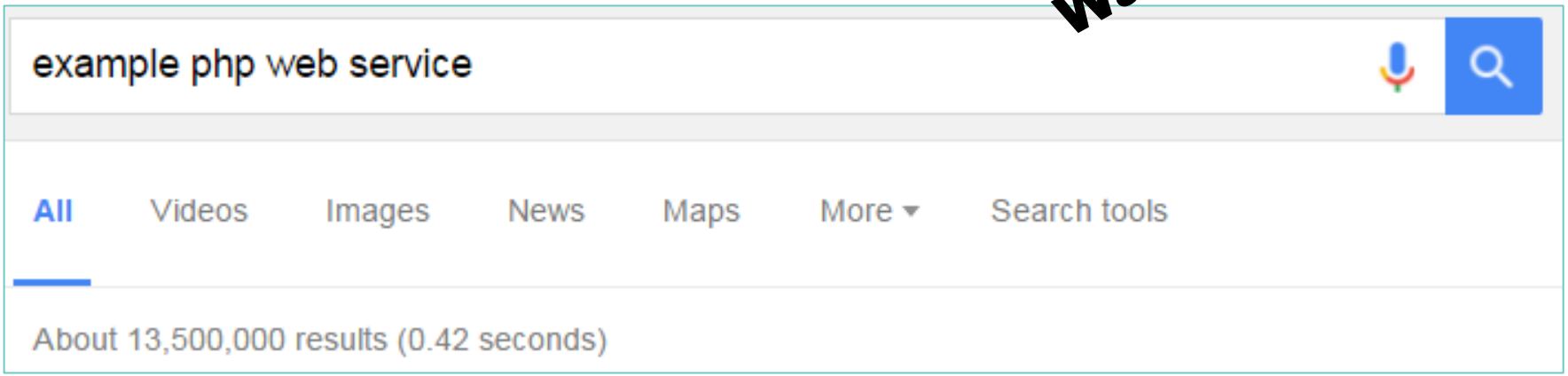
- ZF1 – Singleton
- ZF2 – Factory
- Symfony

- No, MVC is a design architecture…not a design pattern!

Tip #23 – Google and PHP.net

- Your two best friends in PHP development
- When learning a new skill, oftentimes don't know question to ask
- Google heuristics can estimate what you are looking for
  - Often can lead you back to php.net
  - May lead to examples

Watch the Dates!



Tip #24 – Just do it!

- Don’t wait around for your boss to tell you
- Find an excuse to play
- Build a website (Go Daddy is $10 a month)
- Volunteer
  - Church
  - VFW
  - Local rock band
- Internal department project
  - IT help desk
  - Intranet
  - Media wiki for documentation

Tip #25 - OPC (or as Jon likes to say OPO)

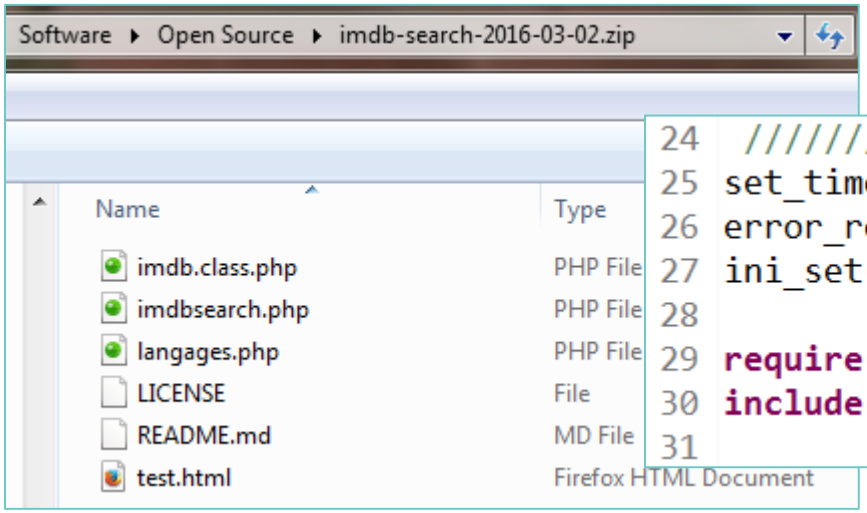
- Other Peoples Code
  - It's how you learned RPG/COBOL
  - Community loves to share, but be careful, not all good code
- Google can help find resources
  - Hotscripts.com, Phpclasses.org, Sourceforge
- Sort by date, descending…for most current stuff!

launchpad

Bitbucket

GitHub

OPC (cont.)



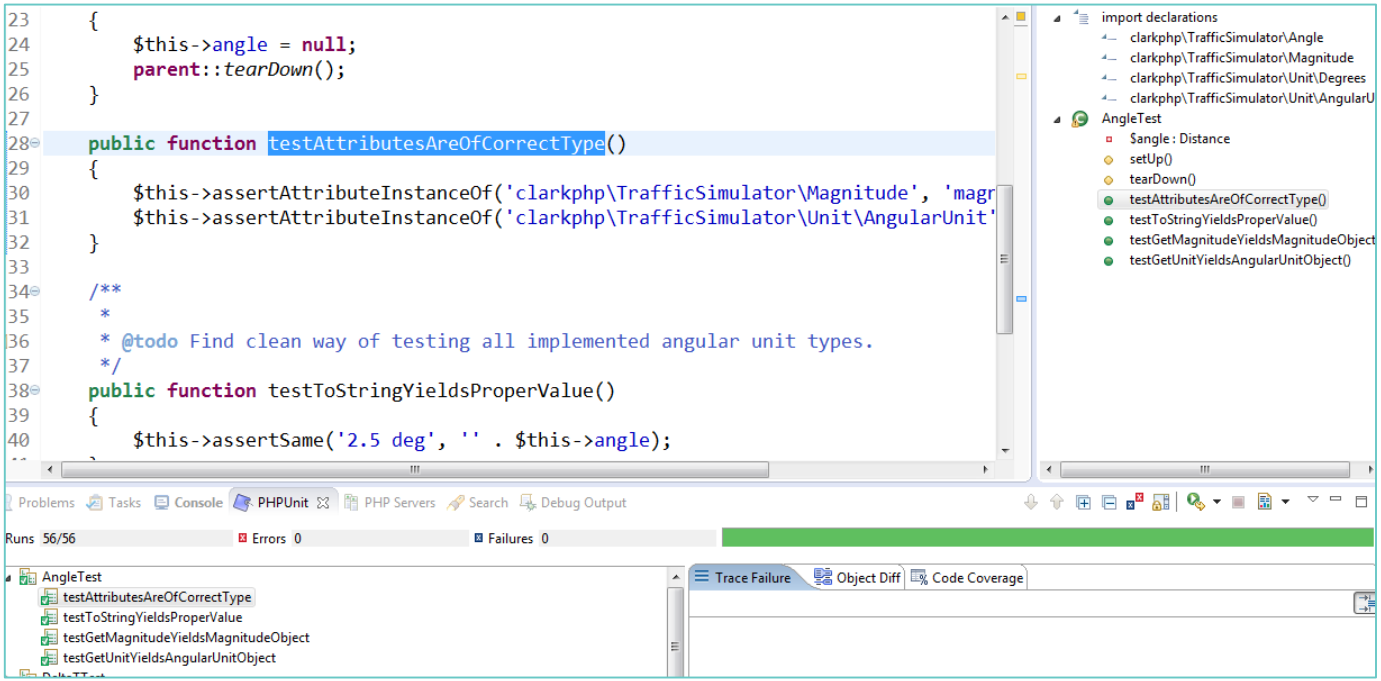

```php
////////////////////////////////////////////////////
set_time_limit(0);
error_reporting(E_ERROR | E_WARNING | E_PARSE);
ini_set("memory_limit" , -1);

require('langages.php');
include("imdb.class.php");
```

```php
class Imdb
{
    // Scrape movie information from IMDb page and return results in an array.
    // this function becomes public by ArouG and has been modified !
    // I prefer use the IMDb search engine and not ASK, GOOGLE and the others ! Much better ^^
    public function scrapeMovieInfo($imdbUrl, $getExtraInfo = true, $myip, $Country = '')
    {
        require('langages.php');
        $arr = array();
        $html = $this->geturl("${imdbUrl}combined",  $myip);
        $title_id = $this->match('/<link rel="canonical" href="http:\/\/www.imdb.com\/title\/(tt\d+)\/combined" \/>/ms',
        if(empty($title_id) || !preg_match("/tt\d+/i", $title_id)) {
            $arr['error'] = "No Title found on IMDb!";
            return $arr;
        }
        $arr['imdb_id'] = $title_id;
        $tmp= str_replace('"', '', trim($this->match('/<title>(IMDb \- )*(.*?) \(.*?<\/title>/ms', $html, 2)));
        $arr['title'] = html_entity_decode($tmp, ENT_COMPAT, 'UTF-8');
        $arr['year'] = trim($this->match('/<title>.*?\(.*?(\d{4}).*?\).*?<\/title>/ms', $html, 1));
```

Tip #26 – Consider test driven development

- Continuous Integration / Delivery is the goal
- Requires company to automate EVERY step in the software life cycle
- Automate Testing, start slow
  - Create test scripts for functions
- Phpunit

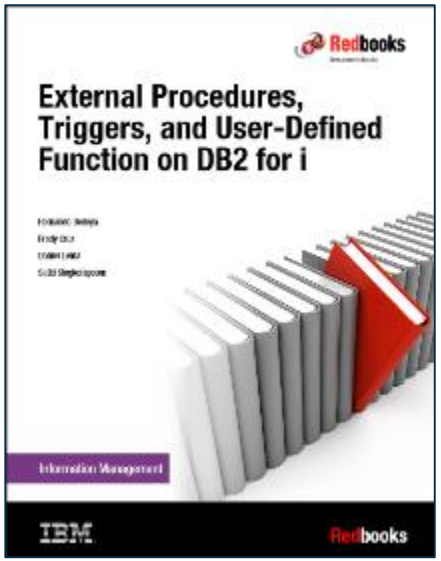


Tip #27 – Source Control

- Existing commercial products handle PHP
  - Rocket, Arcad, Remain, etc.
- Consider a code respository
  - SVN, GiT (on IBM i)
- You are doing more with this source code than ever before
  - Maybe it's time for a change
  - Consider all the elements that make up a website
    - CSS, HTML, JavaScript
- New resources require new rules…

Tip #28 – Master SQL

- IBM DB2 Extension uses same DB2 CLI as the native environment
- Any changes to DB2 are immediately available to PHP
- Every Technology Refresh contains changes to DB2 and often SQL
- SQL is the primary language for data access in PHP
  - Query optimizer
  - iNav interface
  - Index advisor



- New http://publib-b.boulder.ibm.com/redpieces/abstracts/sg246503.html

Tip #29 – Master Security

- You are embarking on a new world (www)
- Consider access to your data
- Consider access to your IBM i
  - Have you looked at your datasets?
- Take a networking class (the network/internet is the computer)

After

Before

User	Group	Object Authority
*PUBLIC		*ALL
MPAVLAK		*ALL

		Object	Data				
User	Group	Authority	Read	Add	Update	Delete	Execute
*PUBLIC		*EXCLUDE	_	_	_	_	_
MPAVLAK		*ALL	X	X	X	X	X
PHPUSER		*USE	X	_	_	_	X

Tip #30 – Date Handling

- Formatting dates in PHP is all about time stamps
- DB2 installations usually store date as integers or date fields
- To leverage PHP, must master conversions

```
3  $today = time();
4
5  echo $today;
6
7  $dateIn = '02/05/2016';
8
9  $dateWrk = strtotime($dateIn);
10
11 echo '<br /><br />' . $dateWrk;
12
13 $formattedDate = date('l F d, Y', $dateWrk);
14
15 echo '<br /><br />' . $formattedDate;
```

1457571317

1454659200

Friday February 05, 2016

Tip #31 – Error suppression

- Embrace your mistakes
- Use of the @ to suppress errors is still supported in PHP 7
  - THIS IS BAD!!!!! Equivalent of MONMSG CPF0000
- Learn to handle errors
  - Try & Catch (example in samples!)
  - Exception processing

```
try {
    $ToolkitServiceObj = ToolkitService::getInstance($db, $user, $pass, $extension);
    }

    catch (Exception $e)
    {
        echo  $e->getMessage(), "\n";
        exit();
    }
```

Practical guide to PHP error from RPG call

http://club.alanseiden.com/blog/2018/04/05/monitoring-for-errors-in-php-script-called-from-clle/

Tip #32 – Never trust external data

- Developer is agent of security in web app design
- Filter inputs (Frameworks do this but so does PHP!)
- JavaScript may do this, but that is not enough…

- Validate filters
- Sanitize filters
- Other filters
- Filter flags

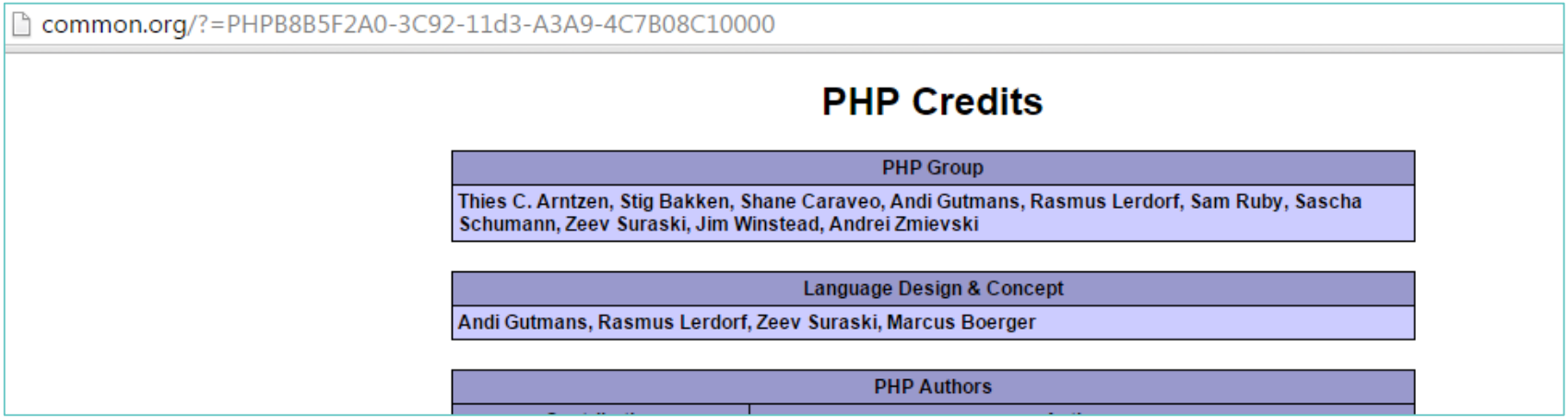
```php
$redisplayForm = FALSE;
$errorCount = 0;

$quantityOK = filter_input(INPUT_POST, 'quantity', FILTER_VALIDATE_INT);

if ($quantityOK == TRUE) {
    $quantityClean = $_POST['quantityIn'];
    echo "Good Quantity!!!";
} else {
    $errorCount++;
    $errorMessage[] = 'Bad Quantity Entered, please try again...';
    $redisplayForm = TRUE;
    echo "Bad Quantity!!!";
}
```

Tip #33 – Who are you?

- Netcraft knows all
- Hide phpinfo
- Never show a path, Clean URL's
- Display errors in development ONLY
- Expose_php in PHP.ini file to reduce visibility
- Gone in PHP7 (Yay!)



THANK YOU

Mike Pavlak

Mike.Pavlak@freschesolutions.com

FRESCHE SOLUTIONS