

The background is a dark navy blue. It features several large, overlapping, semi-transparent geometric shapes in various colors: bright green, cyan, magenta, orange, and red. These shapes are arranged in a way that creates a sense of depth and movement, with some appearing to be layered on top of others. The overall aesthetic is modern and tech-oriented.

MODERNIZING LEGACY APPLICATIONS

Eugene Web Devs, February 22, 2018

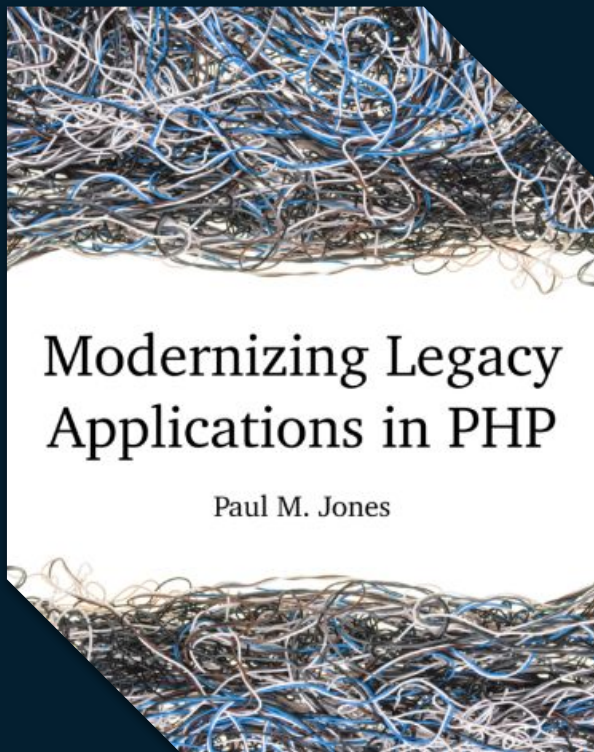


HELLO!

I am Jason Welch

I am the lead application engineer at IDX.

You can find me at @jdubyawelch



Modernizing Legacy Applications in PHP

Get your code under control in a series of small, specific steps.

by Paul M. Jones

<https://leanpub.com/mlaphp>

Rewrite or Refactor?

Rewrite

The act or result of re-implementing a large portion of existing functionality without re-use of its source code or writing inscription.

Refactor

The process of restructuring existing code without changing its external behavior.

The background is a dark navy blue. In the top-left and bottom-left corners, there are overlapping, semi-transparent geometric shapes in shades of green, blue, orange, and pink. In the top-right and bottom-right corners, there are similar overlapping shapes in shades of green, blue, purple, and orange. The shapes are angular and layered, creating a sense of depth and movement.

REWRITE

Things to consider before rewriting.

Context-switching Problem

Split time between old and new projects

- › Less productive for the developer having to switch

The Knowledge Problem

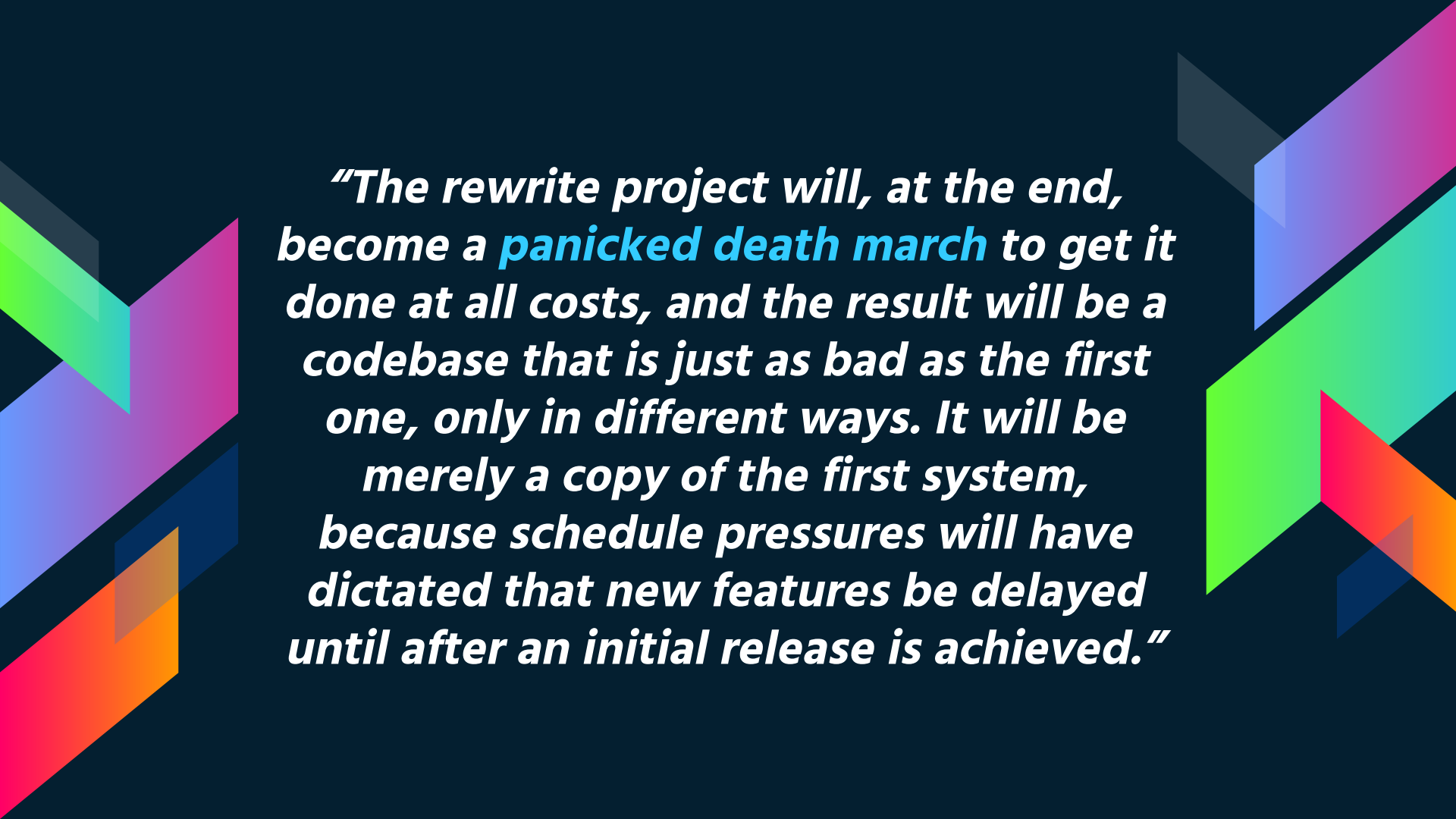
Hire more developers and assign to dedicated systems (old/new)

- › New devs won't know enough about the existing system, problems, goals, best practices
- › A 100% increase in the number of programmers will result in a less than 50% increase in output

The Schedule Problem

Dedicate all developers to rewrite, delay maintenance and upgrades on existing system until the rewrite is done.


- › Tempting = overestimated ability to perform and underestimate time needed
- › Overconfidence and optimism morph into pain and frustration
- › Disappoints existing customers and fails to attract new ones



"The rewrite project will, at the end, become a **panicked death march** to get it done at all costs, and the result will be a codebase that is just as bad as the first one, only in different ways. It will be merely a copy of the first system, because schedule pressures will have dictated that new features be delayed until after an initial release is achieved."

IDX Chose Rewrite

- › Took two years to rewrite entire app
- › 4 ½ years start to finish
- › Dedicated all devs to V2
- › It was a tough process, but we got through it



Netscape made the “single worst strategic mistake that any software company can make” by deciding to rewrite their code from scratch.

ITERATIVE REFACTORING

A decorative graphic on a dark blue background. It features several overlapping, semi-transparent geometric shapes in vibrant colors: purple, blue, green, yellow, and orange. These shapes are arranged in a way that suggests movement or a path. Additionally, there are two small, bright yellow stars positioned near the top right of the main title.

The program is improved in small steps,
without changing the functionality of the
program.

Iterative Refactoring

- › A single small change is added.
- › Test system to ensure it works
- › Deploy to production
- › Then, another small change.

Over a period of time, the system becomes easier to maintain and improve.

Refactoring Drawbacks

Less appealing than a rewrite

Have to continue working with the system as is, warts and all.

Don't get to switch over to the latest hottest framework

Don't get to "do things right the first time"

Developers usually prefer to start projects rather than maintain older projects developed by others.

Refactoring Benefits

Individual refactorings themselves are small compared to any similar portion of a rewrite project.

Can be applied in much shorter periods of time

Leave existing codebase in a working state at the end of each iteration

Can be integrated into a larger process with scheduling that allows for cycles of bug fixes, feature additions, and refactoring to improve the next cycle.

Small steps toward easily-visualized goals that can be accomplished in short time frames.

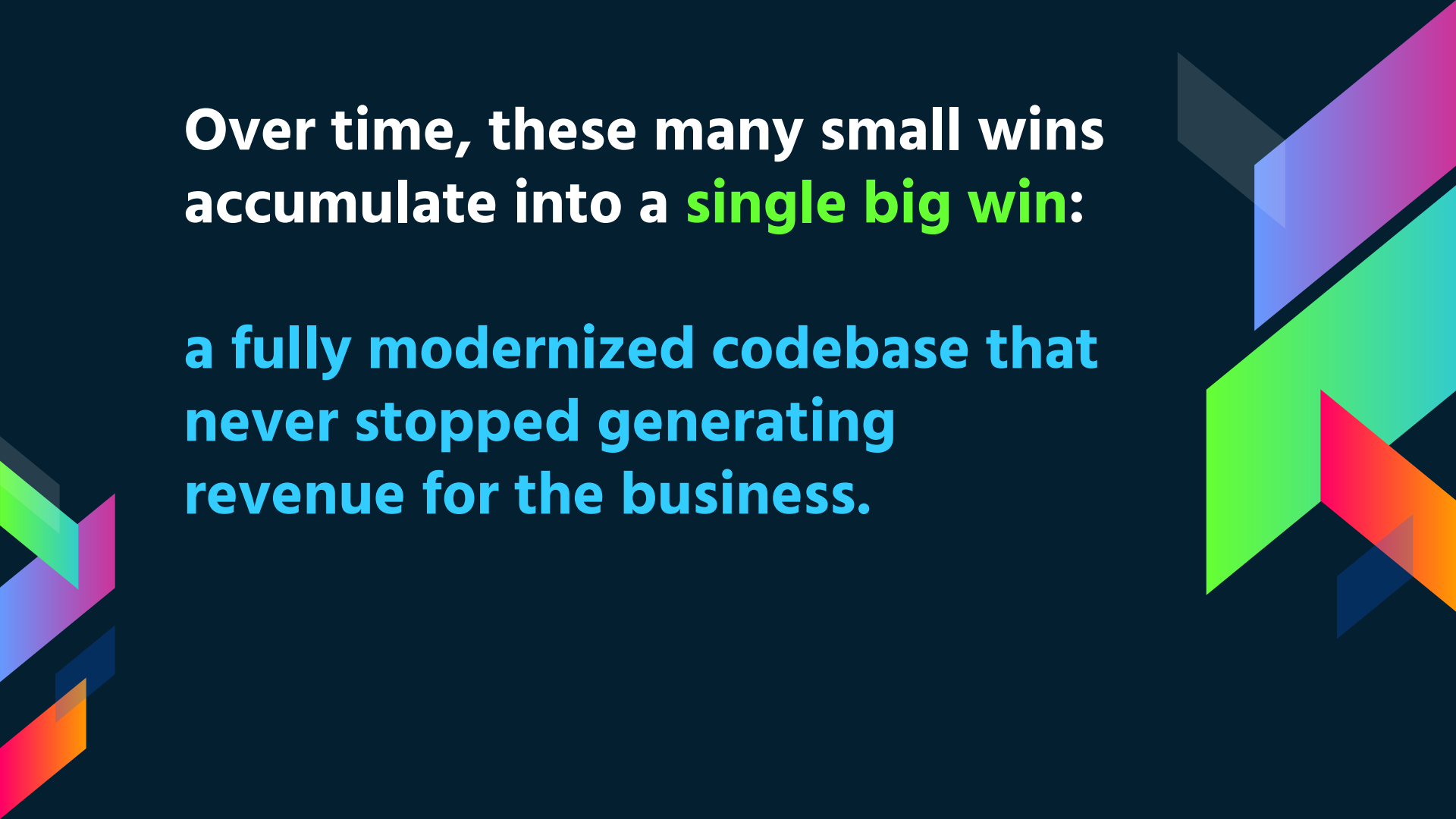
Each small refactoring “win” will both improve morale and drive enthusiasm for the next refactoring step.

The goal of any single
refactoring is not
“**perfection**”.

The goal in each step is
merely “**improvement**”.

Over time, these many small wins
accumulate into a **single big win:**

a fully modernized codebase that
never stopped generating
revenue for the business.



LET'S GET CODING



Refactoring my first ever PHP application



Ocean Crest

[Home](#)

[Login](#)

[Register](#)

OCEAN CREST

[Home](#)[Calendar](#)[Logout](#)[Change Password](#)

Jason

Welch
Schumacher

[Prev](#)

February 2018

[Next](#)

SUN	MON	TUE	WED	THUR	FRI	SAT
				1	2 Jason	3 Jason
4 Jason	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24

OCEAN CREST

[Home](#)[Calendar](#)[Logout](#)[Change Password](#)

Jason

Add a Calendar Event

Date of Arrival:	2/9/2018
No. of days staying**	<input type="text" value="1"/>
<input type="button" value="Add Event"/>	

**note: the number of days does not carry over to the next month

OCEAN CREST

[Home](#)[Calendar](#)[Logout](#)[Change Password](#)

Jason

Jason have reserved the cabin on 2/2/2018

[Return to Calendar](#)

OCEAN CREST

[Home](#)[Login](#)[Register](#)

Register

Name: (for example: Jason & Deena)

Side:

Email Address:

Password: Use only letters and numbers. Must be between 4 and 20 characters long.

Confirm Password:

Unable to load the *Are You a Human PlayThru™*. Please contact the site owner to report the problem.

Register

OCEAN CREST

[Home](#)[Login](#)[Register](#)

Login-Update

Your browser must allow cookies in order to log in.

Email Address:

Password:


[forget password?](#)


Login

OCEAN CREST

[Home](#)[Calendar](#)[Logout](#)[Change Password](#)

Change Your Password

New Password:  Use only letters and numbers. Must be between 4 and 20 characters long.

Confirm New Password: 

[Change My Password](#)

- |— activate.php
- |— add_event.php
- |— calendar.php
- |— change_password.php
- |— details.php
- |— forgot_password.php
- |— index.php
- |— login.php
- |— logout.php
- |— register.php

- |— cgi-bin
 - |— oc
 - |— dbConnection.php
- |— css
 - |— first.css
- |— images
 - |— OceantCrest.jpg
 - |— header.jpg
- |— includes
 - |— ayah.php
 - |— footer.php
 - |— header.php

```
1 <?php
2 include ('../includes/header.php');
3 require_once("../cgi-bin/oc/dbConnection.php");    Include header & database
4
5 if (strlen($_GET['day']) < 1){
6     header("Location: index.php");                Redirect if invalid date
7     exit();
8 }
9
10 if (isset($_SESSION['name'])) {                   Show user's name if logged in
11     echo "<p>$_SESSION[name]</p>";
12 } else {
13     $url = 'http://' . $_SERVER['HTTP_HOST'] . dirname($_SERVER['PHP_SELF']);
14     if ((substr($url, -1) == '/') OR (substr($url, -1) == '\\')) {
15         $url = substr ($url, 0, -1); // Chop off the slash.
16     }
17     $url .= '/index.php';
18
19     ob_end_clean(); // Delete the buffer.          Redirect if not logged in
20     header("Location: $url");
21     exit(); // Quit the script.
22 }
```

```
24 $date = $_GET['day'];           Extract the day for database query
25 $dateArray = explode("/", $date);
26 $startTimestamp = mktime(0,0,0,$dateArray[1],$dateArray[0],$dateArray[2]);
27 $endTimestamp = mktime(23,59,00,$dateArray[1],$dateArray[0],$dateArray[2]);
```

```
28
29 $sql = "SELECT id,family, UNIX_TIMESTAMP(dateField) AS timestampField ";
30 $sql .= "FROM ocCalendar ";
31 $sql .= "WHERE UNIX_TIMESTAMP(dateField) ≥ " . $startTimestamp . " AND UNIX_TIMESTAMP(dateField) ≤ " . $endTimestamp . " ";
32 $sql .= "ORDER BY UNIX_TIMESTAMP(dateField) ASC";
```

```
33
34 $dbResult = mysql_query($sql) or die ("MySQL Error: " . mysql_error() );
35 $numRecords = mysql_num_rows($dbResult);
36 for ($i=0;$i < $numRecords;$i++){
37     $temp = mysql_fetch_assoc($dbResult);
38     if (!get_magic_quotes_gpc()) {
39         $temp['family'] = stripslashes($temp['family']);
40     }
41     $records[] = $temp;
42 }
```

Build query & and get data

```
43
44 $month = $dateArray[1];
45 $day = $dateArray[0];
46 $year = $dateArray[2];
```

Handle cancellation

```
48 if (isset($_POST['delete'])) {
49
50     if ($name == $records[0]['family']) {
51         $id = $_POST['id'];
52
53         $query = "DELETE FROM `ocCalendar` WHERE `id` = '$id' LIMIT 1";
54         $result = mysql_query($query);
55
56         if ($result) {
57             header("Location: calendar.php?month=$month&year=$year");
58             exit();
59         }
60     } else {
61         echo "it's not.";
62     }
63 }
64 ?>
65
```

```

66 <h3><?php echo $records[0]['family'] . " have reserved the cabin on $month/$day/$year"; ?></h3>
67 <p>&nbsp;</p>
68 <?php
69 if (strlen($records[0]['family']) > 0) {
70     $id = $records[0]['id'];
71
72     if ($name == $records[0]['family']) {
73         echo "<form action=\"\" method=\"post\">
74             <input name=\"id\" type=\"hidden\" value=\"\$id\">
75             <input name=\"delete\" type=\"submit\" value=\" click if not staying anymore \">
76         </form>";
77     }
78
79 } else { ?>
80     <p align='center'><h3>No Current Records</h3></p>
81 <?php } ?>
82 <p>&nbsp;</p>
83 <p><a href="calendar.php">Return to Calendar</a> </p>
84 <?
85 include ('./includes/footer.php');
86 ?>

```

HTML of who has the reservation

html/details.php

If user owns registration, show cancellation form

Lastly, close the HTML out by including the footer


```
1  <?php                                                    includes/header.php
2  ob_start();
3  session_start();          PHP session start
4
5  if (!isset($page_title)) {
6      $page_title = 'O C E A N   C R E S T';
7  }
8  ?>
9  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
10 |      <a href="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
11 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
12 <head>          Outputs the HTML head
13 |     <meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
14 |     <title><?php echo $page_title; ?></title>
15 <style type="text/css" media="screen">@import ".css/first.css";</style>
16 </head>
```

```
18 <body>
19 <div id="container">
20     <div id="header">
21         <h1>Ocean Crest </h1>
22         <ul>
23             <li><a href="index.php">home</a></li>
24             <?php                      Echo navigation for logged in
25             if (isset($_SESSION['name']) AND (substr($_SERVER['PHP_SELF'], -10) != 'logout.php'))
26                 echo "<li><a href=\"calendar.php\">calendar</a></li>\n";
27                 echo "<li><a href=\"logout.php\">logout</a></li>\n";
28                 echo "<li><a href=\"change_password.php\">change password</a></li>\n";
29             } else { // they're not logged in
30                 Echo navigation for guest
31                 echo "<li><a href=\"login.php\">login</a></li>\n";
32                 echo "<li><a href=\"register.php\">register</a></li>\n";
33             }
34         </ul>
35     </div>
36 </div>
```



```
1     </div>
2     <div id="footer">
3         <p>&copy;2007 bookoceancrest.com | contact
4             <a href="mailto:jason@jwelchdesign.com">webmaster</a>
5         </p>
6     </div>
7     </div>      Outputs the closing HTML
8 </body>
9 </html>
```

GIT

```
> git init  
> git add all  
> git commit -m "Initial commit"
```

Step 1: Implement An Autoloader

```
1  {  
2      "autoload": {  
3          "psr-4": {  
4              "OceanCrest\\": "app/"  
5          }  
6      }  
7  }
```


composer.json

```
1  <?php
2
3  require __DIR__ . '/../vendor/autoload.php';
4
5      Include the new autoloader
6  ob_start();
7  session_start();
8
9  if (!isset($page_title)) {
10     $page_title = 'O C E A N   C R E S T';
11 }
12 ?>
13 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
14     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
15 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
16 <head>
17     <meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
18     <title><?php echo $page_title; ?></title>
19     <style type="text/css" media="screen">@import "../css/first.css";</style>
20 </head>
```

Step 2: Consolidate **Classes** and **Functions**

Remove all the `include` calls that load up class and function definitions.

Consolidate Classes

1. Find an **include** statement that pulls in a class definition file.
2. Move that class definition file to our central class directory location, making sure that it is placed in a sub-path matching the PSR rules.
3. In the original file and in all other files in the codebase where an include pulls in that class definition, **remove that include statement**. 
4. Commit, push, and notify QA.
5. Repeat until there are no more include calls that pull in class definitions.

*Not applicable for my app, no classes

Consolidate Functions Into Class Files

1. Find a function definition file being included.
2. Convert the functions into a class file of static methods.
 - a. Make class unique and update methods to be more suitable.
3. Throughout the app, change calls to those functions into static method calls. ✓
4. Move class into app/ and spot check again. ✓
5. Remove any relevant include calls. ✓
6. Commit, push, notify QA
7. Repeat until no more include calls pull in function definitions.

```
+use OceanCrest\DB;    Import new class
```

```
+
```

```
// Set the page title and include the HTML header.
```

```
$page_title = 'O C E A N  C R E S T >> Change Your Password';
```

```
include ('./includes/header.php');
```

```
@@ -30,7 +34,7 @@
```

```
    // Check for a new password and match against the confirmed password.
```

```
    if (eregi ('^[[[:alnum:]]{4,20}$', stripslashes(trim($_POST['password1'])))) {
```

```
        if ($_POST['password1'] == $_POST['password2']) {
```

```
-            $p = escape_data($_POST['password1']);
```

```
+    Update to call static method    $p = DB::escape_data($_POST['password1']);
```

```
        } else {
```

```
            $p = FALSE;
```

```
            echo '<p><font color="red" size="+1">Your password did not ma
```



```
1 <?php namespace OceanCrest;
```

```
2
3 // app/DB.php
```

Function is the same, just with
public static in front and is in a
class.

```
4 class DB {
```

```
5
6     public static function escape_data ($data) {
```

```
7
8         if (ini_get('magic_quotes_gpc')) {
```

```
9             $data = stripslashes($data);
```

```
10         }
```

```
11
12         if (function_exists('mysql_real_escape_string')) {
```

```
13             global $dbc;
```

```
14             $data = mysql_real_escape_string (trim($data), $dbc);
```

```
15         } else {
```

```
16             $data = mysql_escape_string (trim($data));
```

```
17         }
```

```
18
19         return $data;
```

```
20     }
```

```
21 }
```

Step 3: Replace `global` With Dependency Injection

Remove all uses of the `global keyword` from our classes and `inject` the necessary dependencies instead.

```
5 + private $connection;
6 +
7 + public function __construct($connection)
8 + {
9 +     $this->connection = $connection;
10 + }
11 +
```

```
5 12 // Create a function for escaping the data.
6 - public static function escape_data ($data) {
13 + public function escape_data ($data) {
7 14
8 15 // Address Magic Quotes.
9 16 if (ini_get('magic_quotes_gpc')) {
❖ @@ -12,8 +19,7 @@ public static function escape_data ($data) {
12 19
13 20 // Check for mysql_real_escape_string() support.
14 21 if (function_exists('mysql_real_escape_string')) {
15 - global $dbc; // Need the connection.
16 - $data = mysql_real_escape_string (trim($data), $dbc);
22 + $data = mysql_real_escape_string (trim($data), $this->connection);
17 23 } else {
18 24 $data = mysql_escape_string (trim($data));
```

```
@@ -12,18 +12,20 @@

2   12     if (isset($_POST['submitted'])) { // Check if the form has been submitted.
3   13
4   14         require_once("../cgi-bin/oc/dbConnection.php"); // Connect to the database.
5   15     +
6   16     +     $db = new DB($dbc);
7   17
8   18         // Validate the email address.
9   19         if (!empty($_POST['email'])) {
10  20     -         $e = DB::escape_data($_POST['email']);
11  20     +         $e = $db->escape_data($_POST['email']);
12  21     } else {
13  22         echo '<p><font color="red" size="+1">You forgot to enter your email a
14  23         $e = FALSE;
15  24     }
16  25
```

- › Look for superglobals as well
 - › `$_SERVER['REQUEST_METHOD']`
- › Can use a Request object instead.
- › Treat `$GLOBALS` just like global.

```
17
18  // Build the Request Object
19  $request = new \Mlaphp\Request($GLOBALS);
20
21  // Server is not set in GLOBALS for whatever reason
22  // Let's manually add it here ourselves
23  $request->server = $_SERVER;
24
```

```
28 28 // define their name
29 -$name = $_SESSION['name'];
30 -$uid = $_SESSION['user_id'];
31 -$side = $_SESSION['side'];
29 +$name = $request->session['name'];
30 +$uid = $request->session['user_id'];
31 +$side = $request->session['side'];
32 32
33 33
34 -if (isset($_POST['submit'])) {
34 +if (isset($request->post['submit'])) {
35 35
36 - $submit = $_POST['submit'];
37 - $month_now = $_POST['month'];
38 - $year_now = $_POST['year'];
36 + $submit = $request->post['submit'];
37 + $month_now = $request->post['month'];
38 + $year_now = $request->post['year'];
39 39
40 40 // Subtract one from month for prev and add 1 for next
41 41 if ($submit == "Prev") {
```

Step 4: Replace **new** With Dependency Injection

There still could be hidden dependencies in classes where **new objects are being created** in inappropriate locations, tightly coupling the classes together.

*Might not be applicable in old apps

Step 5: Write Tests

- › Install PHPUnit
- › Create a `tests/` directory
- › Pick a class
- › Write a test case for every public method in your classes

Don't skip this step!

Start small and get some early wins.

Step 6: Extract **SQL** statements to Gateways

Extract more logic from our page scripts and place them into classes.

1. Keeps various concerns of the application **separated**
2. The extracted logic can be **tested** so that any breaks will be easy to notice before we deploy to production.

```
// Create Date Time Field
$time = "00:00:00";
$dateArray = explode("/", $startDate);
$d = $dateArray[0];
$yr = $dateArray[2];
$mo = $dateArray[1];

if ($duration > 1) {
    for ($i = 0; $i < $duration; $i++) {
        $reserveDates = $d + $i;
        $datetime = "$yr-$mo-$reserveDates $time";
        $sql = "INSERT INTO ocCalendar (dateField,family,event)
                VALUES ('" . $datetime . "','" . $family . "','" . $event . "')";
        mysql_query($sql);
    }
} else {
    $datetime = "$yr-$mo-$d $time";
    $sql = "INSERT INTO ocCalendar (dateField,family,event)
            VALUES ('" . $datetime . "','" . $family . "','" . $event . "')";
    mysql_query($sql);
}
```

SQL insert for multiple days

SQL insert for single day

```
$eventGateway = new EventGateway($dbc);  
$eventGateway→reserve($family, $date, $staying);
```

Code is cleaner and more readable now

```
/** @test */
```

```
function it_stores_the_date_for_stay()
```

```
{  
    $user = $this->makeUser();  
  
    $event_id = $this->gateway->reserve(  
        $user->side,  
        '02/01/2016',  
        1  
    );  
  
    $this->assertGreaterThan(0, $event_id);  
    $events = $this->gateway->monthlyEvents('02', '2016');  
    $this->assertCount(1, $events);  
}
```

Step 7: Extract Domain Logic to Transactions

Extract the domain logic to one or more classes so that we can test them independently of any particular page script. We can then use the classes throughout, removing possible duplication.

- › Most demanding step.
- › No modifications or improvements to the domain logic itself.
- › Any changes are just enough to get data into our new Transaction classes
- › Any existing problems are now testable problems.

html/activate.php
Without transactions

```
if ($x > 0) {  
  
    $db = new DB(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);  
    $userGateway = new UserGateway($db);  
    $activated = $userGateway->activate($x);    Use the gateway to activate a  
                                              user  
  
    if ($activated) {  
        echo "<h3>Your account is now active. You may now log in.</h3>";  
    } else {                                Output success/failure  
        echo '<p><font color="red" size="+1">Your account could not be activated. Please re-check the link or contact  
    }  
  
    $email = $userGateway->email($x);    Get user's email and send them a  
                                       notification  
  
    $body = "You're bookoceancrest account has now been activated.\n\n  
Please login at http://www.bookoceancrest.com/login.php using your email and the password you set up.\n\n  
This was an automatically generated email, please do not reply.\n\n  
With questions about the website contact Jason Welch at jason@jwelchdesign.com\n\n";  
  
    mail($email, "Ocean Crest Account", $body, 'From: info@bookoceancrest.com');
```


html/activate.php
With transactions

Notice the instantiations here...

Do all the things to activate an
account using a transaction

Output success/failure

```
3 use OceanCrest\DB;
4 use OceanCrest\UserGateway;
5 use OceanCrest\UserTransactions;
6
7 // ...
8
9 if ($x > 0) {
10     $db = new DB(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);
11     $userGateway = new UserGateway($db);
12     $userTransactions = new UserTransactions($userGateway);
13
14     $activated = $userTransactions->activate($x);
15
16     if ($activated) {
17         echo "<h3>Your account is now active. You may now log in.</h3>";
18     } else {
19         echo '<p><font color="red" size="+1">'.implode('<br />', $userTransactions->getErrors()).'</font></p>';
20     }
21 } else { // Redirect.
```

```
public function activate($user_id)
{
    $this->errors = [];

    $activated = $this->userGateway->activate($user_id);

    if (! $activated) {
        $this->errors[] = 'Your account could not be activated. Please re-check the link or contact the system admin';
        return false;
    }

    if (! $this->sendActivationNotification($user_id)) {
        $this->errors[] = 'Could not send activation notification';
        return false;
    }

    return true;
}
```

Step 8: Extract **Presentation Logic** to **View Files**

Now we need to **separate our presentation logic** to its own layer so we can test it separately from our business logic.

Put presentation code in **/views**

```
1  <?php
2  ▢ if (!isset($page_title)) {
3      $page_title = 'OCEAN CREST';
4  }
5  ?>
6  ▢ <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
7      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
8  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
9  ▢ <head>
10     <meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
11     <title><?php echo \$page\_title; ?></title>
12     <style type="text/css" media="screen">@import "../css/first.css";</style>
13 </head>
14     Extract presentation only header
15     code to this new partial view
16 ▢ <body>
17     <div id="container">
```

```
1  <?php
2  ini_set('display_errors', 'On');
3  error_reporting(E_ERROR / E_WARNING / E_PARSE);
4
5  require __DIR__ . '/../vendor/autoload.php';
6  require __DIR__ . '/../cgi-bin/oc/dbConnection.php';
7
8  ob_start();          Extract non-presentation header
9  session_start();     code to this new bootstrap file
10
11 $request = new \Mlaphp\Request($GLOBALS);
12 $request->server = $_SERVER;
13
14
```

```
3      3      use OceanCrest\DB;
4      4      use OceanCrest\EventGateway;
5
6      -include ('./includes/header.php');
7      -require_once("../cgi-bin/oc/dbConnection.php"); ?>
8      6      +require_once("../bootstrap/start.php");
9      7
10     9      -<?php
11     8      // Check that we have a date parameter in the URL, if none redirect back to calendar
12     9      if(strlen($request->get['day']) < 1){
13     10         header("Location: index.php");
14
15     @@ -15,7 +13,6 @@
16     13
17     14      // MAKE SURE THE
18     15      if (isset($request->session['name'])) {
19     -      echo "<p>$request->session[name]</p>";
20     16      $name = $request->session['name'];
21     17      } else {
22     18      // Start defining the URL.
```

Instead of echoing, we store as variables to pass to the view.

```
58
59 $response = new \Mlapph\Response(__DIR__.'/../views');
60 $response->setView('details.php');
61 $response->setVars([
62     'request' => $request,
63     'name' => $request->session['name'],
64     'event' => [
65         'family' => $records[0]['family'],
66         'id' => $records[0]['id'],
67         'date' => "$month/$day/$year",
68         'owned' => ($request->session['name'] == $records[0]['family'])
69     ]
70 ]);
71 $response->send();
72
```

html/details.php

No echoing, instead pass all data into the response object.


```
1 <?php
2 include (__DIR__.'../views/partials/header.php');
3 echo "<p>{$name}</p>";
4 ?>
5 <h3><?php echo "{$event['family']}" have reserved the cabin on "{$event['date']}"; ?></h3>
6 <?php
7 if(strlen($event['family']) > 0) {
8     if ($event['owned']) {
9         echo '<form action="" method="post">
10             <input name="id" type="hidden" value="'. $this->esc($event['id']).' ">
11             <input name="delete" type="submit" value=" click if not staying anymore ">
12             </form>';
13     }
14 } else { ?>
15     <p align='center'><h3>No Current Records</h3></p>
16 <?php
```

Views are only concerned with
presentation logic.


```
public function setUp()
{
    $this->request = new \Mlaphp\Request($GLOBALS);
    $this->response = new \Mlaphp\Response(__DIR__.'/../views');
    $this->response->setView('details.php');
    $this->response->setVars([
        'request' => $this->request,
        'name' => 'Jason & Deena',
        'event' => [
            'family' => 'Jason & Deena',
            'id' => 1,
            'date' => "3/30/2016",
            'owned' => true
        ]
    ]);
    $this->output = $this->response->requireView();
}
```

Given this data

/** @test */

```
function it_displays_the_date_and_who_is_staying()
{
    We expect this output
    $expect = '<h3>Jason & Deena have reserved the cabin on 3/30/2016</h3>';
    $this->assertOutputHas($expect);
}
```

Step 9: Extract **Action Logic** to **Controllers**

We must extract the action code to a class of its own to separate the various remaining concerns of our page script.

*Tip: Rearrange script code into blocks:
Dependency, Controller, Finished*

Dependency

html/details.php

validate day

verify user logged in

Dependency

get event info from db

handle the reservation cancelation

response

Finished

Controller

Controller code is copied to a controller.

```
1 <?php
2
3 use Mlaphp\Response;    Dependency
4 use OceanCrest\DB;
5 use OceanCrest\EventGateway;
```

```
6 use OceanCrest\Controllers\EventController;
```

```
7
8 require_once("../bootstrap/start.php");
9
```

```
10 $db = new DB(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);
```

```
11 $eventGateway = new EventGateway($db);
```

```
12 $response = new Response(DIR '../views');
```

```
13 $controller = new EventController($request, $eventGateway, $response);
```

Controller

```
15 $response = $controller->__invoke();
```

```
16 $response->send();    Finished
17
```

Step 10: Replace **Includes** In Classes

We need to **replace include calls with method calls** throughout our classes.

Find **include** calls in classes

If included only once:

Paste the contents of the included file inline.

Refactor the copied code so that it follows all our existing rules: no globals, no new, inject dependencies, return instead of output, and no include calls.

Included multiple times

Copy contents, as is, to a new class method.

Replace include with new inline instantiation.

Create a test for new class, refactor using existing rules.

Step 11: Separate **Public** and **Non-Public** Resources

Extract all public resources to a new document root to keep non-public resources from being delivered by accident.





└─ tests

└─ TestCase.php

└─ app

└─ AuthGatewayTest.php

└─ AuthTransactionsTest.php

└─ CalendarPresenterTest.php

└─ Controllers

└─ CalendarControllerTest.php

└─ EventControllerTest.php

└─ EventGatewayTest.php

└─ EventTransactionsTest.php

└─ GatewayTester.php

└─ UserGatewayTest.php

└─ UserTransactionsTest.php

└─ WeekOwnershipSwapTest.php

└─ WeekOwnershipTest.php

└─ bootstrap.php

└─ sbx

└─ calendar-list.txt

└─ calendar_math.php

└─ dateWithEvenWeekEachYear.php

└─ dateWithOddWeekEachYear.php

└─ weekOwnershipList.php

└─ views

└─ AddEventHtmlTest.php

└─ CalendarHtmlTest.php

└─ DetailsHtmlTest.php

PHPUnit 4.8.29 by Sebastian Bergmann and contributors.

.....

Time: 1.94 seconds, Memory: 11.00MB

OK (50 tests, 65 assertions)

Step 12: Decouple URL Paths From File Paths

Decouple the paths so that we can route URLs independently to any target we want.



```
1 <?php
2
3 require dirname(__DIR__).'./bootstrap/start.php';
4
5 // Set up the router.
6 $pages_dir = dirname(__DIR__).'./pages'; New up the Router pointed to /pages
7 $router = new Mlaphp\Router($pages_dir); Move from /html to /pages
8
9 // Match against the url path.
10 $path = parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH);
11 $route = $router->match($path);
12
13 // Require the page script. Require the file /details.php or /login.php
14 require $route;
15
16 if (isset($controller)) {
17     $response = $controller->__invoke();
18     $response->send();
19 }
```

Step 13: Remove Repeated Logic in Page Scripts

Scripts in `/pages` are pretty similar:

- › Loads a setup script
- › Instantiates dependencies for controller
- › Invokes the controller
- › Sends response



@@ -3,9 +3,7 @@

<pages/details.php>

3 3 use Mlaphp\Response;

4 4 use OceanCrest\DB;

5 5 use OceanCrest\EventGateway;

6 -use OceanCrest\Controllers\EventController;

7 - No need for start.php (front.php is doing that for us)

8 -require_once("../bootstrap/start.php");

6 +use OceanCrest\Controllers\ReservedEventController;

9 7

10 8 // MAKE SURE THEY ARE LOGGED IN

11 9 if (! isset(\$request->session['name'])) {



@@ -27,7 +25,6 @@

27 25 \$db = new DB(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);

28 26 \$eventGateway = new EventGateway(\$db);

29 27 \$response = new Response(__DIR__.'../views');

30 - \$controller = new EventController(\$request, \$eventGateway, \$response);

28 + \$controller = new ReservedEventController(\$request, \$eventGateway, \$response);

No need, front.php is calling

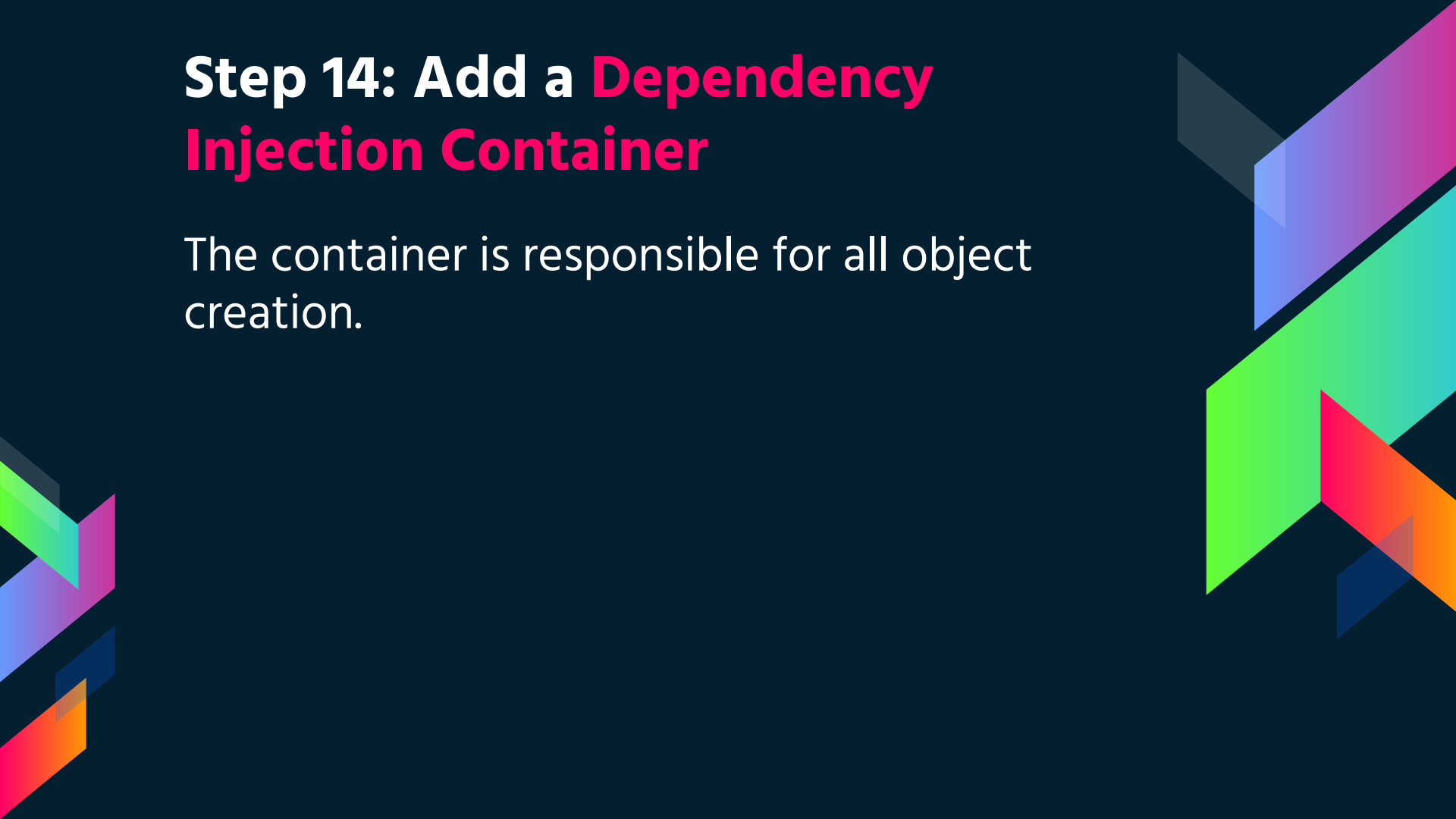
\$controller->__invoke()

32 - \$response = \$controller->show();

33 - \$response->send();

Step 14: Add a **Dependency Injection Container**

The container is responsible for all object creation.



```
1 <?php
2 require dirname(__DIR__).'./bootstrap/start.php';
3 require dirname(__DIR__).'./bootstrap/services.php';
4
5 $router = $di->get('router'); Replaced router instantiation with $di->get()
6
7 // Match against the url path.
8 $path = parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH);
9 $route = $router->match($path);
10
11 // Container service, or page script?
12 if ($di->has($route)) {
13     // Create a new controller instance.
14     $controller = $di->newInstance($route);
15     $response = $controller->__invoke();
16     $response->send();
17 } else {
18     // Require the page script.
19     require $route;
20 }
```

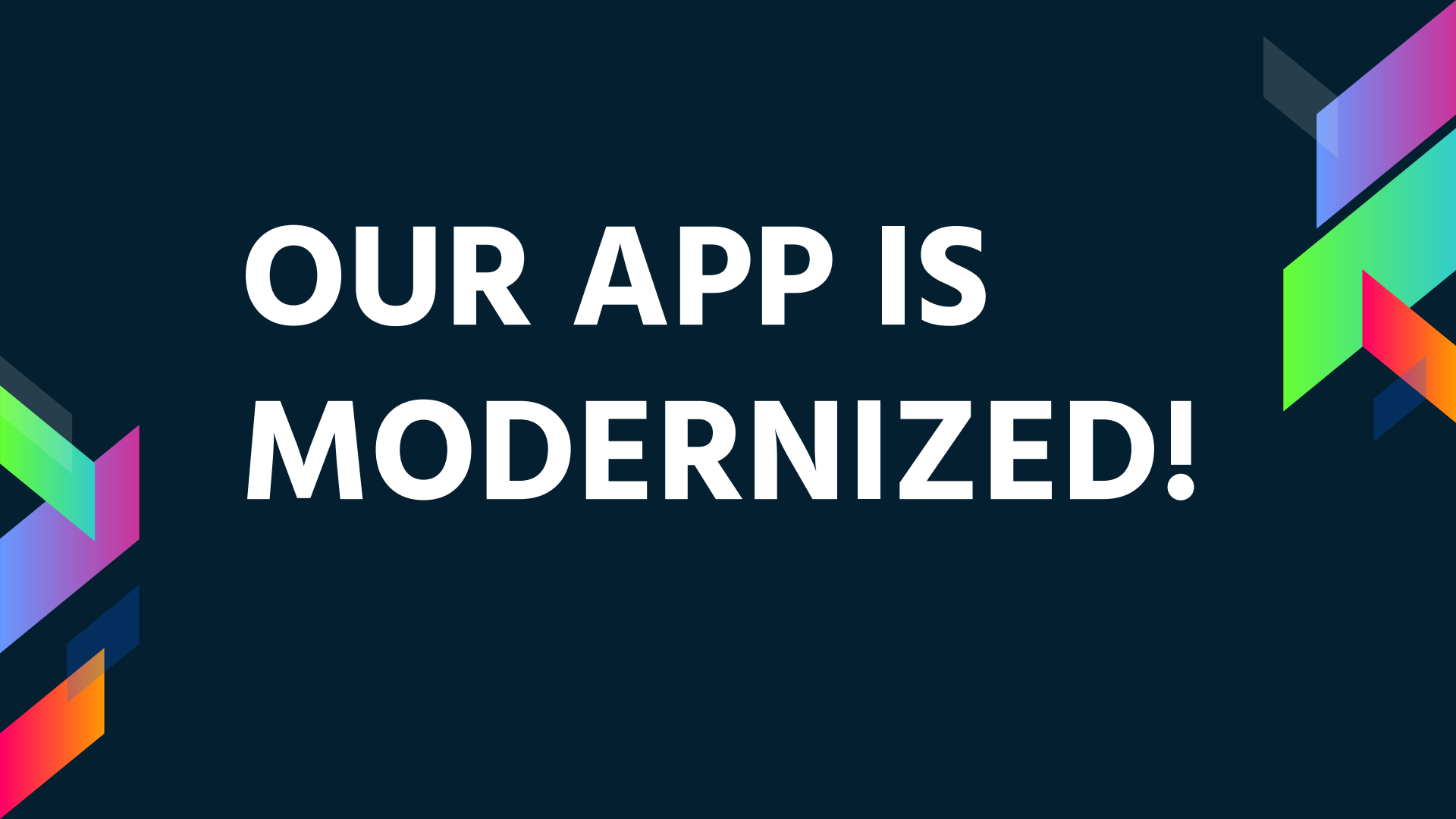
```
1  <?php
2  $di = new Mlapp\Di($GLOBALS);
3
4  $di->set('database', function() {
5      return new OceanCrest\DB(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);
6  });
7
8  $di->set('response', function() {
9      return new \Mlapp\Response(__DIR__.'/../views');
10 });
11
12 $di->set('router', function() {
13     $router = new Mlapp\Router(dirname(__DIR__).'./pages');
14     $router->setRoutes([
15         '/calendar.php' => 'OceanCrest\Controllers\CalendarController',
16         '/calendar'     => 'OceanCrest\Controllers\CalendarController',
17         '/add_event.php' => 'OceanCrest\Controllers\CreateEventController',
18         '/details.php'  => 'OceanCrest\Controllers\ReservedEventController',
19     ]);
20     return $router;
```



```
41 $di->set('OceanCrest\Controllers\ReservedEventController', function() use ($di) {
42     $di->get('middleware');
43     return new OceanCrest\Controllers\ReservedEventController(
44         $di->request,
45         $di->newInstance('OceanCrest\EventGateway'),
46         $di->newInstance('response')
47     );
48 });
49
50 $di->set('OceanCrest\EventGateway', function() use ($di) {
51     return new OceanCrest\EventGateway($di->get('database'));
52 });
```

```
71 $di->set('middleware', function() use ($di) {
72     // MAKE SURE THEY ARE LOGGED IN
73     if (! isset($di->request->session['name'])) {
74         // Start defining the URL.
75         $url = 'http://' . $di->request->server['HTTP_HOST'] . dirname($di->request->server['PHP_SELF']);
76         // Check for a trailing slash.
77         if ((substr($url, -1) == '/') OR (substr($url, -1) == '\\')) {
78             $url = substr ($url, 0, -1); // Chop off the slash.
79         }
80         // Add the page.
81         $url .= '/index.php';
82
83         ob_end_clean(); // Delete the buffer.
84         header("Location: $url");
85         exit(); // Quit the script.
86     }
87 });
```

**OUR APP IS
MODERNIZED!**



OCEAN CREST

[Home](#)[Calendar](#)[Logout](#)[Change Password](#)

Jason

Welch
Schumacher

[Prev](#)

February 2018

[Next](#)

SUN	MON	TUE	WED	THUR	FRI	SAT
				1	2 Jason	3 Jason
4 Jason	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24

Welch
Schumacher

FEBRUARY 2018

[Last Month](#)[Next Month](#)

SUN	MON	TUE	WED	THUR	FRI	SAT
				1	2 Kathy & Tom	3 Kathy & Tom
4 Kathy & Tom	5 Kathy & Tom	6	7	8	9 Bob & Sally	10 Bob & Sally
11 Bob & Sally	12	13	14	15	16 Rob & Sherry	17 Rob & Sherry
18 Rob & Sherry	19	20	21	22	23 Bob & Sally	24 Bob & Sally
25 Bob & Sally	26	27	28			

February ▾

2018 ▾

[View Month](#)



THANKS!

Any questions?

You can find me at:

@jdubyawelch · jason@idxbroker.com

Credits

Special thanks to all the people who made and released these awesome resources for free:

- › Presentation template by [SlidesCarnival](#)
- › Photographs by [Startupstockphotos](#)