



How to eat an elePHPant

# Modernize Legacy PHP Applications



# Modernize Legacy PHP Applications



Ayesh Karunaratne | <https://aye.sh/talk/dpc2023-modernize-php>



# Ayesh Karunaratne

---

**Freelance Software Developer, Security Researcher**  
**Full-time traveler, Author of PHP.Watch**

 **Hanoi, Vietnam**

 **<https://aye.sh> | <https://php.watch>**

 **@Ayesh**

 **ayesh@aye.sh**

# Modernize Legacy PHP Applications



Ayesh Karunaratne | <https://aye.sh/talk/dpc2023-modernize-php>



**There is only one way to  
eat an elephant**



**There is only one way to  
eat an elephant**

**One bite at a time**

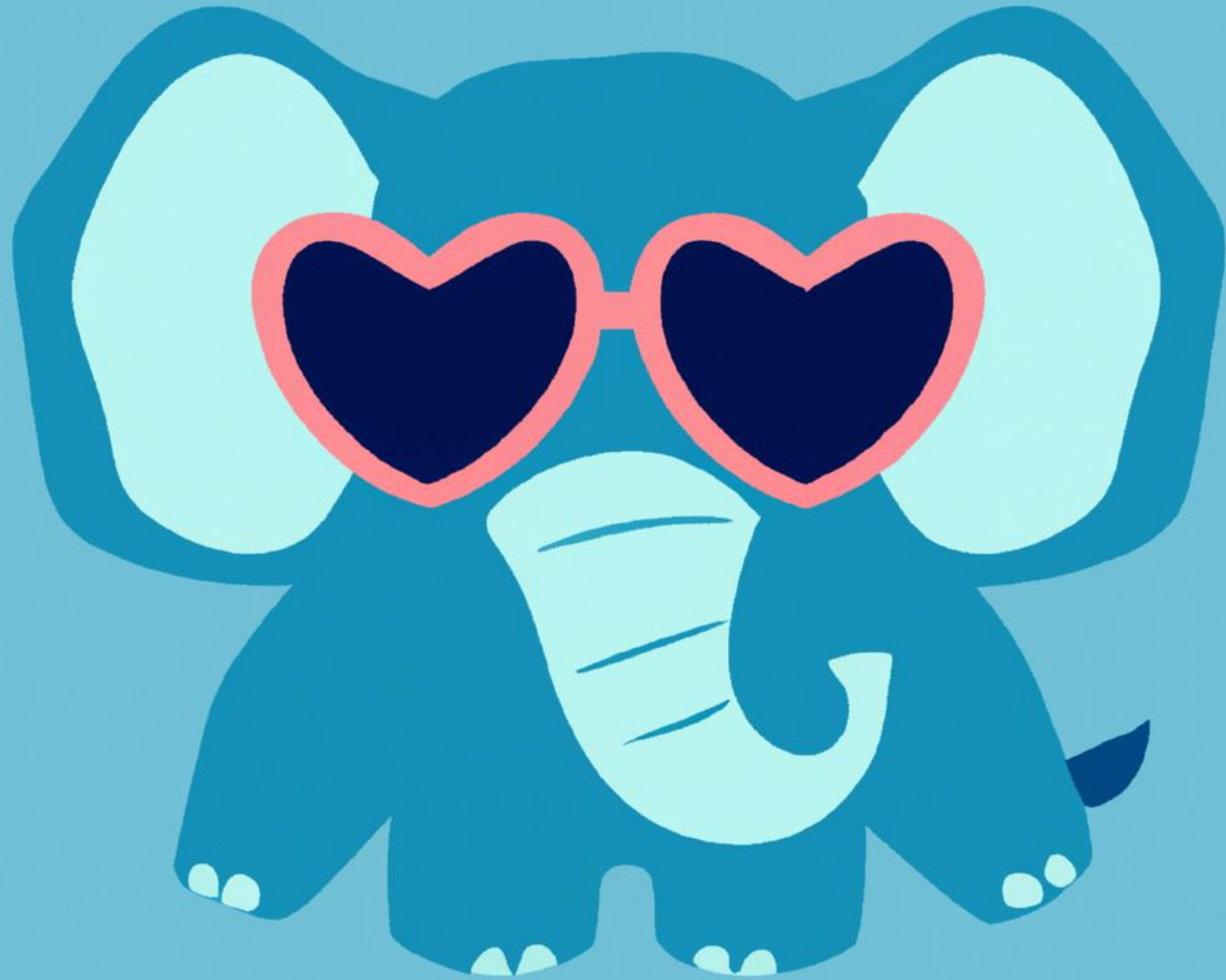
– Desmond Tutu



**There is only one way to  
eat an elePHPant**

**One bite at a time**

– Desmond Tutu



**No Animals Were Harmed**  
in the making of this presentation



# **Modernize Legacy PHP Applications**

# Modernize Legacy PHP Applications

```
css/  
images/  
js/  
includes/  
  - db.inc  
templates/  
  - footer.php  
  - header.php  
  - nav.php  
libraries/  
  - phpmailer/  
    - phpmailer.php  
index.php  
login.php  
user.php  
register.php
```

```
css/  
images/  
js/  
includes/  
  - db.inc  
templates/  
  - footer.php  
  - header.php  
  - nav.php  
libraries/  
  - phpmailer/  
    - phpmailer.php  
index.php  
login.php  
user.php  
register.php
```



css/

images/

js/

includes/

- db.inc

templates/

- footer.php

- header.php

- nav.php

libraries/

- phpmailer/

  - phpmailer.php

index.php

login.php

user.php

register.php

css/

images/

js/

includes/

- db.inc

templates/

- footer.php

- header.php

- nav.php

libraries/

- phpmailer/

- phpmailer.php

index.php

login.php

user.php

register.php

```
$db = new PDO('mysql:host=localhost;dbname=mydb', 'myuser', 'mypass');
```

css/

images/

js/

includes/

- db.inc

templates/

- footer.php

- header.php

- nav.php

libraries/

- phpmailer/

- phpmailer.php

index.php

login.php

user.php

register.php

```
$db = new PDO('mysql:host=localhost;dbname=mydb', 'myuser', 'mypass');
```

- css/
- images/
- js/
- includes/
  - db.inc
- templates/
  - footer.php
  - header.php
  - nav.php
- libraries/
  - phpmailer/
    - phpmailer.php
- index.php
- login.php
- user.php
- register.php

```
$db = new PDO('mysql:host=localhost;dbname=mydb', 'myuser', 'mypass');
```



- css/
- images/
- js/
- includes/
  - db.inc
- templates/
  - footer.php
  - header.php
  - nav.php
- libraries/
  - phpmailer/
    - phpmailer.php
- index.php
- login.php
- user.php
- register.php

```
<footer>  
  © <?= $sitename ?> | <a href="https://example.com">Home</a>  
</footer>
```

- css/
- images/
- js/
- includes/
  - db.inc
- templates/
  - footer.php
  - header.php
  - nav.php
- libraries/
  - phpmailer/
    - phpmailer.php
- index.php
- login.php
- user.php
- register.php

```
<footer>  
  © <?= $sitename ?> | <a href="https://example.com">Home</a>  
</footer>
```

- css/
- images/
- js/
- includes/
  - db.inc
- templates/
  - footer.php
  - header.php
  - nav.php
- libraries/
  - phpmailer/
    - phpmailer.php
- index.php
- login.php
- user.php
- register.php

```
<footer>  
  © <?= $sitename ?> | <a href="https://example.com">Home</a>  
</footer>
```

```
css/  
images/  
js/  
includes/  
  - db.inc  
templates/  
  - footer.php  
  - header.php  
  - nav.php  
libraries/  
  - phpmailer/  
    - phpmailer.php  
index.php  
login.php  
user.php  
register.php
```

- Updated manually
- No integrity checks
- Included on-demand



- css/
- images/
- js/
- includes/
  - db.inc
- templates/
  - footer.php
  - header.php
  - nav.php
- libraries/
  - phpmailer/
    - phpmailer.php
- index.php
- login.php
- user.php
- register.php

```
<?php
require 'includes/db.inc';
?>
<html>
<head>
    <?php include 'templates/header.php' ?>
</head>
<body>
<?php if (user_logged_in()): ?>
    <?= 'Hello ' . $user ?>
<?php else: ?>
    <a href="https://example.com/login.php">Login</a>
<?php endif ?>
</body>
</html>
```

- css/
- images/
- js/
- includes/
  - db.inc
- templates/
  - footer.php
  - header.php
  - nav.php
- libraries/
  - phpmailer/
    - phpmailer.php
- index.php
- login.php
- user.php
- register.php

```
<?php
require 'includes/db.inc';
?>
<html>
<head>
    <?php include 'templates/header.php' ?>
</head>
<body>
<?php if (user_logged_in()): ?>
    <?= 'Hello ' . $user ?>
<?php else: ?>
    <a href="https://example.com/login.php">Login</a>
<?php endif ?>
</body>
</html>
```

- css/
- images/
- js/
- includes/
  - db.inc
- templates/
  - footer.php
  - header.php
  - nav.php
- libraries/
  - phpmailer/
    - phpmailer.php
- index.php
- login.php
- user.php
- register.php

```
<?php
require 'includes/db.inc';
?>
<html>
<head>
    <?php include 'templates/header.php' ?>
</head>
<body>
<?php if (user_logged_in()): ?>
    <?= 'Hello ' . $user ?>
<?php else: ?>
    <a href="https://example.com/login.php">Login</a>
<?php endif ?>
</body>
</html>
```

```
css/  
images/  
js/  
includes/  
  - db.inc  
templates/  
  - footer.php  
  - header.php  
  - nav.php  
libraries/  
  - phpmailer/  
    - phpmailer.php  
index.php  
login.php  
user.php  
register.php
```

```
<?php
```

```
global $db;  
global $user;  
// ...
```

```
css/  
images/  
js/  
includes/  
  - db.inc  
templates/  
  - footer.php  
  - header.php  
  - nav.php  
libraries/  
  - phpmailer/  
    - phpmailer.php  
index.php  
login.php  
user.php  
register.php
```

```
<?php
```

```
global $db;  
global $user;  
// ...
```





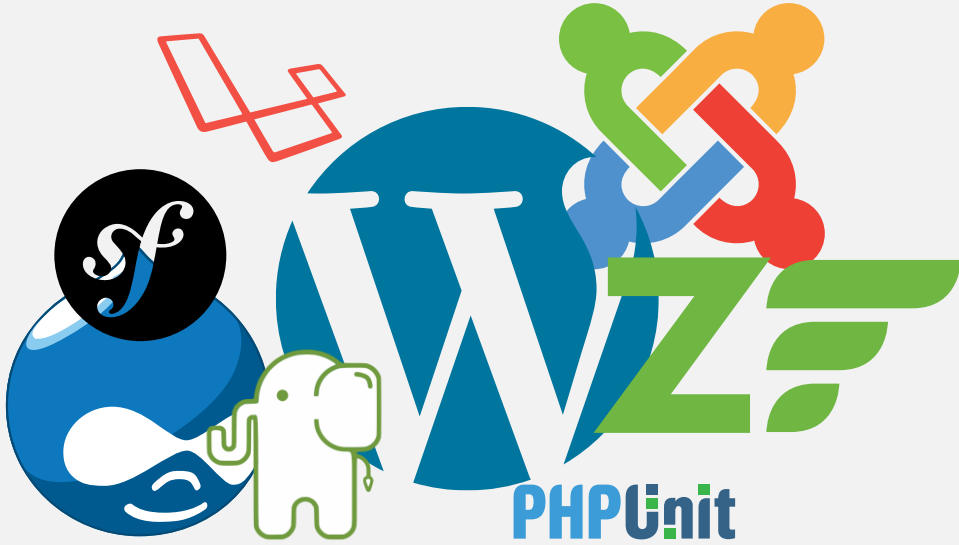


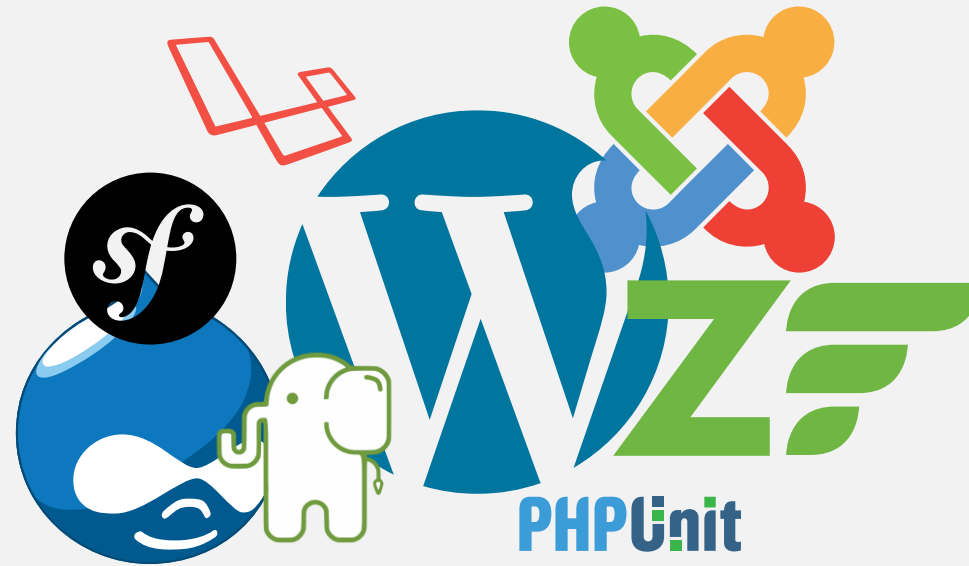
PHP 5.0 – PHP 5.6



PHP 5.0 – PHP 5.6

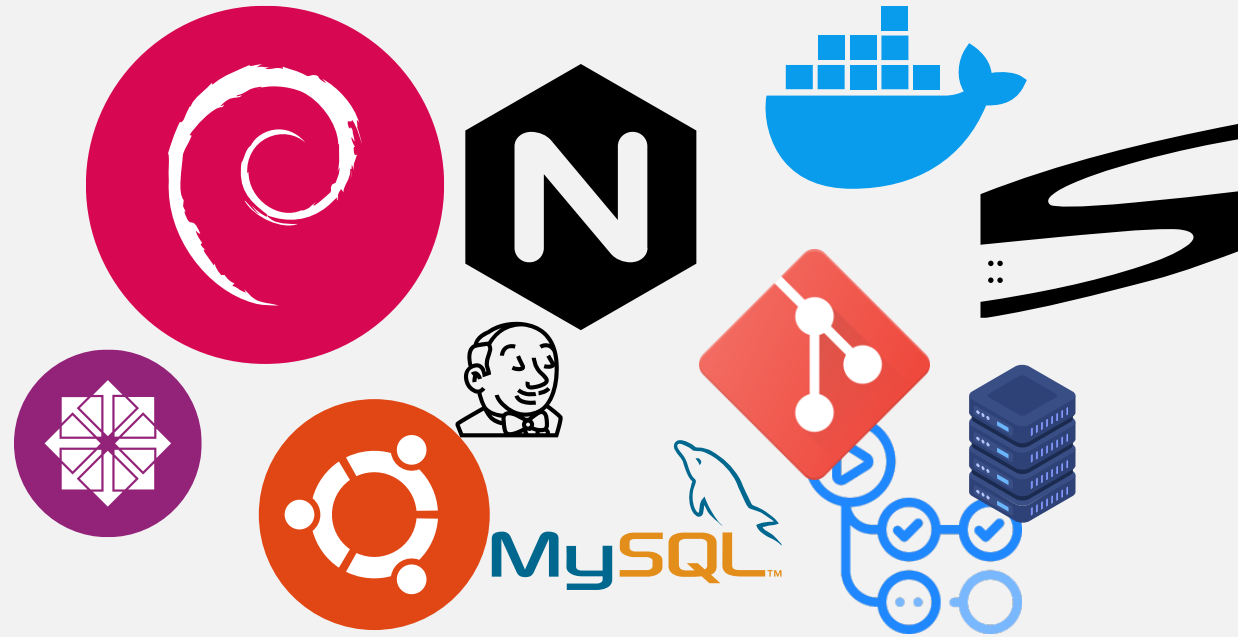
- Lack of namespaces support (< 5.3)
- Lack of class autoload support (< 5.3)
- No built-in CSPRNG
- Missing Opcache improvements, no AST, no JIT
- Register globals, magic quotes and safe mode (5.4)
- No Password Hashing built-in (< 5.5)
- Proper exception support
- Lack of scalar types
- Extensions removed in newer PHP versions (mysql, mhash, etc)

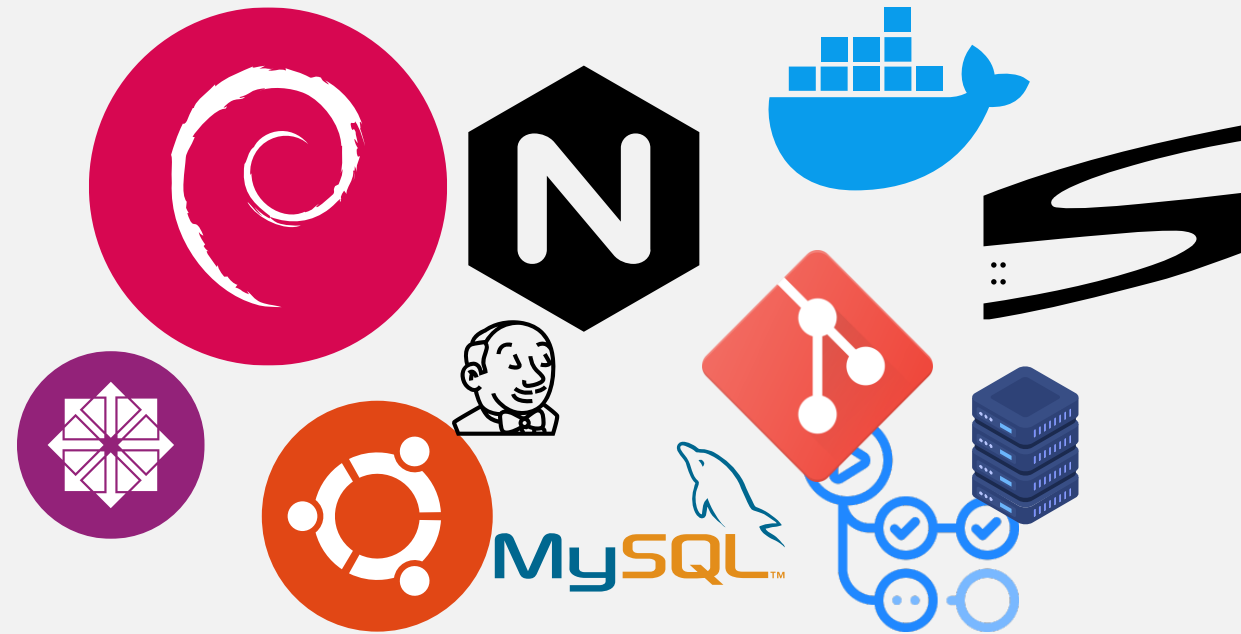




## Libraries and Frameworks

- End-of-life libraries and frameworks
- Major paradigm shifts
- Plugins/Modules incompatibilities
- Unfeasible configuration management
- Tight coupling with external software and libraries





## Server and Dev Dependencies



## Server and Dev Dependencies

- End-of-life Operating systems and server software
- Lack of proper Unicode support
- Inability to use modern protocols (HTTP/2, HTTP/3, etc.)
- Difficult/impossible in development and staging environments
- Tightly coupled software and configuration
- Incompatible features in abandoned systems

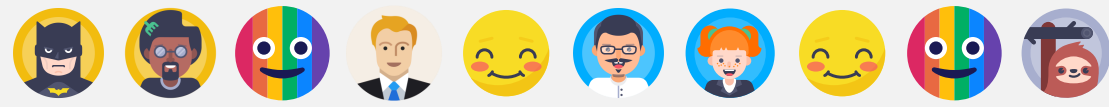
*Litmus test*







## People and Organizational Structures



## People and Organizational Structures

- Various levels of expertise
- Various areas of expertise
- Resistance to change
- Lack of funds
- Lack of motivation

# **Modernize Legacy PHP Applications**

# **Why Modernize**

**Why Modernize**  
**Why Not Rewrite**

It already serves the  
**Business Purpose**

# **Familiarity**



# **Practicality**

# **Modernize Legacy PHP Applications**

# How to Modernize

*Juicy stuff!*

# How to Modernize



Directory Structure



Consolidate Classes/Functions



Consolidate Configuration



Dependency Injection Container



Composer & Autoloading



Separate Business Logic and View Logic



Error Handling

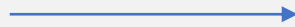
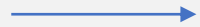


Front Controller

# Directory Structure

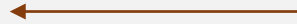
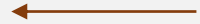
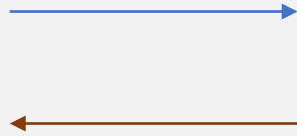
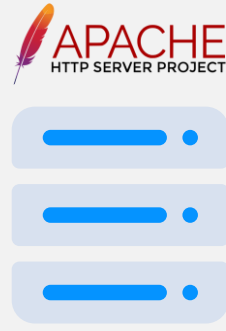
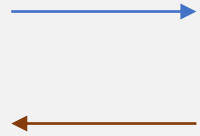
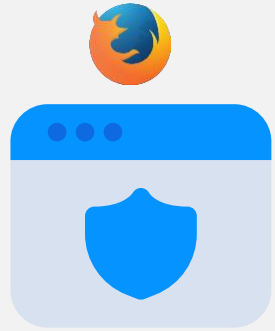


```
css/  
images/  
js/  
includes/  
  - db.inc  
templates/  
  - footer.php  
  - header.php  
  - nav.php  
libraries/  
  - phpmailer/  
    - phpmailer.php  
index.php  
login.php  
user.php  
register.php
```



```
css/  
images/  
js/  
includes/  
  - db.inc  
templates/  
  - footer.php  
  - header.php  
  - nav.php  
libraries/  
  - phpmailer/  
    - phpmailer.php  
index.php  
login.php  
user.php  
register.php
```



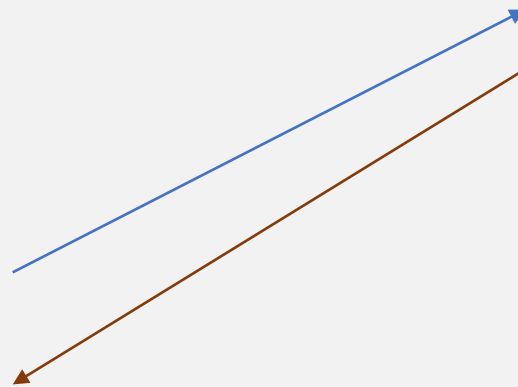
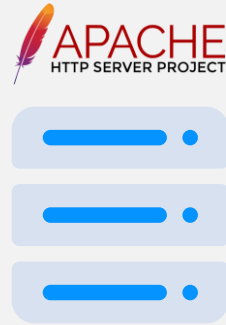
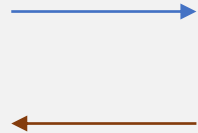
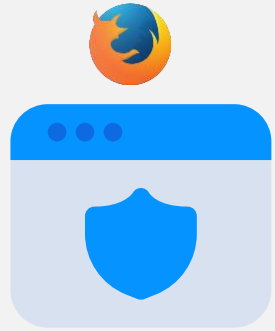


GET `css/styles.css`

```
css/  
images/  
js/  
includes/  
  - db.inc  
templates/  
  - footer.php  
  - header.php  
  - nav.php  
libraries/  
  - phpmailer/  
    - phpmailer.php  
index.php  
login.php  
user.php  
register.php
```







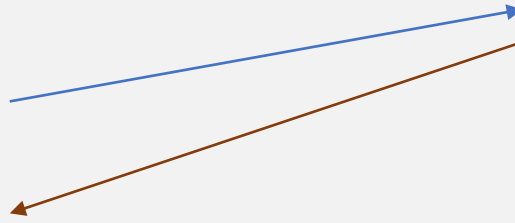
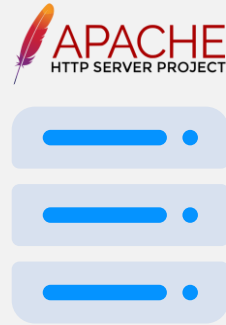
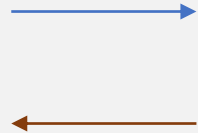
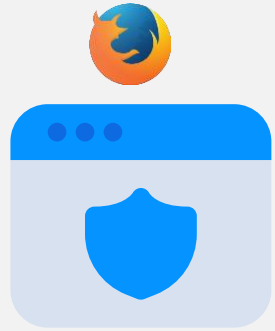
```
css/  
images/  
js/  
includes/  
  - db.inc  
templates/  
  - footer.php  
  - header.php  
  - nav.php  
libraries/  
  - phpmailer/  
    - phpmailer.php  
index.php  
login.php  
user.php  
register.php
```

```
GET css/styles.css
```

```
HTTP/2 200
```

```
html {  
  font-size: 16px;  
}
```





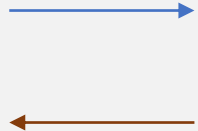
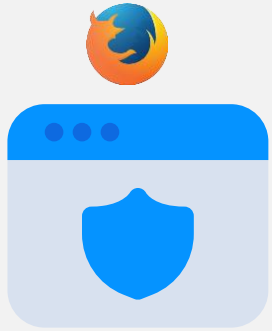
```
css/  
images/  
js/  
includes/  
  - db.inc  
templates/  
  - footer.php  
  - header.php  
  - nav.php  
libraries/  
  - phpmailer/  
    - phpmailer.php  
index.php  
login.php  
user.php  
register.php
```

GET js/script.js

HTTP/2 200

```
document.getElementById();
```



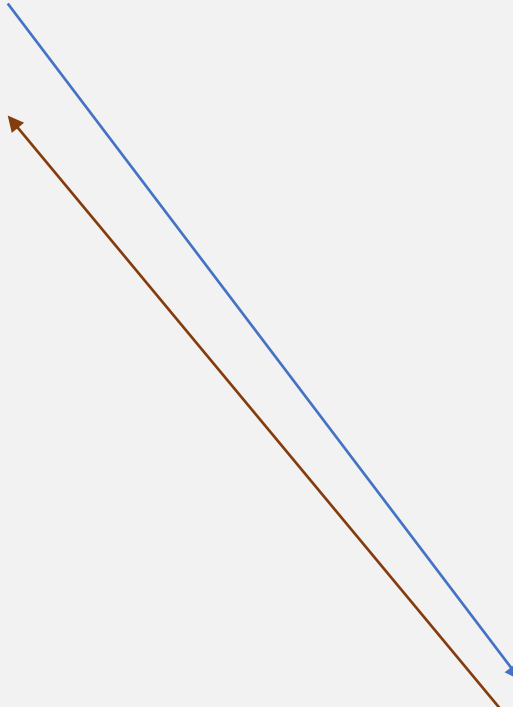


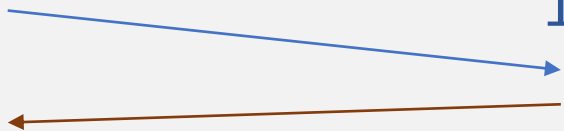
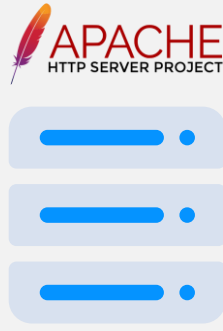
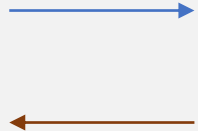
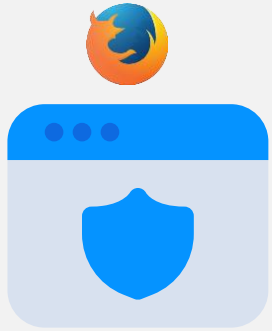
GET index.php

HTTP/2 200

```
<html>  
  <head>  
  ...
```

```
css/  
images/  
js/  
includes/  
  - db.inc  
templates/  
  - footer.php  
  - header.php  
  - nav.php  
libraries/  
  - phpmailer/  
    - phpmailer.php  
index.php  
login.php  
user.php  
register.php
```



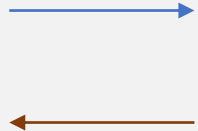
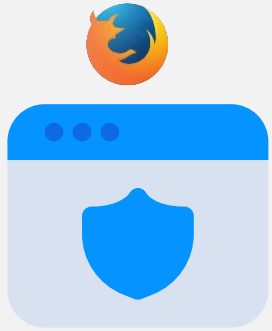


```
css/  
images/  
js/  
includes/  
- db.inc  
templates/  
- footer.php  
- header.php  
- nav.php  
libraries/  
- phpmailer/  
  - phpmailer.php  
index.php  
login.php  
user.php  
register.php
```

GET includes/db.inc

HTTP/2 200





GET includes/db.inc

HTTP/2 200

```
css/  
images/  
js/  
includes/  
- db.inc  
templates/  
- footer.php  
- header.php  
- nav.php  
libraries/  
- phpmailer/  
- phpmailer.php  
index.php  
login.php  
user.php  
register.php
```



```
css/  
images/  
js/  
includes/  
  - db.inc  
templates/  
  - footer.php  
  - header.php  
  - nav.php  
libraries/  
  - phpmailer/  
    - phpmailer.php  
index.php  
login.php  
user.php  
register.php
```



```
app/  
  - src/  
  - templates/  
    - footer.twig | php  
    - header.twig | php  
    - nav.twig | php  
config/  
  - settings.php  
  - default.settings.php  
public/  
  - index.php  
  - css/  
  - images/  
  - js/  
vendor/  
  - phpmailer/phpmailer/  
composer.json  
composer.lock
```

```
css/  
images/  
js/  
includes/  
  - db.inc  
templates/  
  - footer.php  
  - header.php  
  - nav.php  
libraries/  
  - phpmailer/  
    - phpmailer.php  
index.php  
login.php  
user.php  
register.php
```



```
app/  
  - src/  
  - templates/  
    - footer.twig | php  
    - header.twig | php  
    - nav.twig | php  
config/  
  - settings.php  
  - default.settings.php  
public/  
  - index.php  
  - css/  
  - images/  
  - js/  
vendor/  
  - phpmailer/phpmailer/  
composer.json  
composer.lock
```

```
css/  
images/  
js/  
includes/  
  - db.inc  
templates/  
  - footer.php  
  - header.php  
  - nav.php  
libraries/  
  - phpmailer/  
    - phpmailer.php  
index.php  
login.php  
user.php  
register.php
```



```
app/  
  - src/  
  - templates/  
    - footer.twig | php  
    - header.twig | php  
    - nav.twig | php  
config/  
  - settings.php  
  - default.settings.php  
public/  
  - index.php  
  - css/  
  - images/  
  - js/  
vendor/  
  - phpmailer/phpmailer/  
composer.json  
composer.lock
```



```
css/  
images/  
js/  
includes/  
- db.inc  
templates/  
- footer.php  
- header.php  
- nav.php  
libraries/  
- phpmailer/  
  - phpmailer.php  
index.php  
login.php  
user.php  
register.php
```



```
app/ _____  
- src/  
- templates/  
  - footer.twig | php  
  - header.twig | php  
  - nav.twig | php  
config/  
- settings.php  
- default.settings.php  
public/  
- index.php  
- css/  
- images/  
- js/  
vendor/  
- phpmailer/phpmailer/  
composer.json  
composer.lock
```

*At the top*

```
css/  
images/  
js/  
includes/  
  - db.inc  
templates/  
  - footer.php  
  - header.php  
  - nav.php  
libraries/  
  - phpmailer/  
    - phpmailer.php  
index.php  
login.php  
user.php  
register.php
```



```
app/  
  - src/  
  - templates/  
    - footer.twig | php  
    - header.twig | php  
    - nav.twig | php  
config/  
  - settings.php  
  - default.settings.php  
public/  
  - index.php  
  - css/  
  - images/  
  - js/  
vendor/  
  - phpmailer/phpmailer/  
composer.json  
composer.lock
```

```
css/  
images/  
js/  
includes/  
  - db.inc  
templates/  
  - footer.php  
  - header.php  
  - nav.php  
libraries/  
  - phpmailer/  
    - phpmailer.php  
index.php  
login.php  
user.php  
register.php
```



```
app/  
  - src/  
  - templates/  
    - footer.twig | php  
    - header.twig | php  
    - nav.twig | php  
config/  
  - settings.php  
  - default.settings.php  
public/  
  - index.php  
  - css/  
  - images/  
  - js/  
vendor/  
  - phpmailer/phpmailer/  
composer.json  
composer.lock
```

*Front Controller*

```
css/  
images/  
js/  
includes/  
  - db.inc  
templates/  
  - footer.php  
  - header.php  
  - nav.php  
libraries/  
  - phpmailer/  
    - phpmailer.php  
index.php  
login.php  
user.php  
register.php
```



```
app/  
  - src/  
  - templates/  
    - footer.twig | php  
    - header.twig | php  
    - nav.twig | php  
config/  
  - settings.php  
  - default.settings.php  
public/  
  - index.php  
  - css/  
  - images/  
  - js/  
tests/  
vendor/  
  - phpmailer/phpmailer/  
composer.json  
composer.lock
```

Application logic

Configuration

Assets and Static Files

Tests

Dependencies

```
app/  
  - src/  
  - templates/  
    - footer.twig | php  
    - header.twig | php  
    - nav.twig | php  
config/  
  - settings.php  
  - default.settings.php  
public/  
  - index.php  
  - css/  
  - images/  
  - js/  
tests/  
vendor/  
  - phpmailer/phpmailer/  
composer.json  
composer.lock
```

Application logic

Configuration

Assets and Static Files

Tests

Dependencies

Build files

```
app/  
- src/  
- templates/  
  - footer.twig | php  
  - header.twig | php  
  - nav.twig | php
```

```
config/  
- settings.php  
- default.settings.php
```

```
.env  
public/  
- index.php  
- css/  
- images/  
- js/
```

```
tests/  
phpunit.xml
```

```
vendor/  
  - phpmailer/phpmailer/  
composer.json  
composer.lock  
package.json
```

```
.github/workflows/ci.yml  
.gitignore  
.gitattributes  
.phpstan
```

# Consolidate Configuration



```
$db = new PDO('mysql:host=localhost;dbname=mydb', 'myuser', 'mypass');
```



```
$db = new PDO('mysql:host=localhost;dbname=mydb', 'myuser', 'mypass');
```



```
$uploaddir = '/var/www/uploads/';  
$uploadfile = $uploaddir . basename($_FILES['userfile']['name']);  
  
move_uploaded_file($_FILES['userfile']['tmp_name'], $uploadfile);
```

```
$uploaddir = '/var/www/uploads/';  
$uploadfile = $uploaddir . basename($_FILES['userfile']['name']);  
  
move_uploaded_file($_FILES['userfile']['tmp_name'], $uploadfile);
```

```
header("Location: https://example.com/new-page");
```



```
header("Location: https://example.com/new-page");
```



```
app/  
  - src/  
  - templates/  
    - footer.twig | php  
    - header.twig | php  
    - nav.twig | php  
config/  
  - settings.php  
  - default.settings.php  
.env  
public/  
  - index.php  
  - css/  
  - images/  
  - js/  
tests/  
phpunit.xml  
vendor/  
  - phpmailer/phpmailer/  
composer.json  
composer.lock  
package.json  
.github/workflows/ci.yml  
.gitignore  
.gitattributes  
.phpstan
```

```
app/  
  - src/  
  - templates/  
    - footer.twig | php  
    - header.twig | php  
    - nav.twig | php  
config/  
  - settings.php  
  - default.settings.php  
.env  
public/  
  - index.php  
  - css/  
  - images/  
  - js/  
tests/  
phpunit.xml  
vendor/  
  - phpmailer/phpmailer/  
composer.json  
composer.lock  
package.json  
.github/workflows/ci.yml  
.gitignore  
.gitattributes  
.phpstan
```

```
config/  
  - settings.php  
  - default.settings.php  
.env  
.gitignore
```





```
config/  
  - settings.php  
  - default.settings.php  
.env  
.gitignore
```

```
vendor/  
.env  
config/settings.php
```



*Highly opinionated*

config/  
- settings.php  
- default.settings.php  
.env  
.gitignore

vendor/  
.env  
config/settings.php



```
config/  
  - settings.php  
  - default.settings.php  
.env  
.gitignore
```

```
<?php  
  
return [  
    'database' => [  
        'dsn' => 'mysql:host=localhost;dbname=mydb',  
        'username' => 'myuser',  
        'password' => 'mypass',  
    ],  
    'siteUrl' => 'https://example.com',  
    'uploadDir' => __DIR__ . '/../..../uploads',  
];
```



```
<?php

return [
    'database' => [
        'dsn' => 'mysql:host=localhost;dbname=mydb',
        'username' => 'myuser',
        'password' => 'mypass',
    ],
    'siteUrl' => 'https://example.com',
    'uploadDir' => __DIR__ . '/../../uploads',
];
```

```
$db = new PDO(
    'mysql:host=localhost;dbname=mydb',
    'myuser',
    'mypass',
);
```

```
<?php

return [
    'database' => [
        'dsn' => 'mysql:host=localhost;dbname=mydb',
        'username' => 'myuser',
        'password' => 'mypass',
    ],
    'siteUrl' => 'https://example.com',
    'uploadDir' => __DIR__ . '/../../uploads',
];
```

```
$settings = require __DIR__ . '/../../config/settings.php';
$db = new PDO(
    'mysql:host=localhost;dbname=mydb',
    'myuser',
    'mypass',
);
```

```
<?php

return [
    'database' => [
        'dsn' => 'mysql:host=localhost;dbname=mydb',
        'username' => 'myuser',
        'password' => 'mypass',
    ],
    'siteUrl' => 'https://example.com',
    'uploadDir' => __DIR__ . '/../../uploads',
];
```

```
$settings = require __DIR__ . '/../../config/settings.php';
$db = new PDO(
    $settings['dsn'],
    $settings['database']['username'],
    $settings['database']['password']
);
```



```
<?php

return [
    'database' => [
        'dsn' => 'mysql:host=localhost;dbname=mydb',
        'username' => 'myuser',
        'password' => 'mypass',
    ],
    'siteUrl' => 'https://example.com',
    'uploadDir' => __DIR__ . '/../../uploads',
];
```

```
$settings = require __DIR__ . '/../../config/settings.php';
```

```
$uploaddir = $settings['uploadDir'];
```

```
$uploadfile = $uploaddir . basename($_FILES['userfile']['name']);
```

```
move_uploaded_file($_FILES['userfile']['tmp_name'], $uploadfile);
```

```
<?php
```

```
return [  
    'database' => [  
        'dsn' => 'mysql:host=localhost;dbname=mydb',  
        'username' => 'myuser',  
        'password' => 'mypass',  
    ],  
    'siteUrl' => 'https://example.com',  
    'uploadDir' => __DIR__ . '/../..../uploads',  
];
```



*Opcache*

```
$settings = require __DIR__ . '/../..../config/settings.php';  
header("Location: ".$settings['siteUrl'].'/new-page');
```





# Composer & Autoloading



# Why use Composer?



# Why use Composer?

composer.json

```
{
    "name": "you/your-app",
    "description": "Lorem ipsum",
    "type": "project",
    "license": "proprietary",
    "require": {
        "php": "^8.2",
        "ext-pdo": "*",
        "ext-mbstring": "*",
        "erusev/parsedown": "^1.7.4",
    },
    "autoload": {
        "psr-4": {
            "App \\": "app/src/"
        }
    }
}
```



# Why use Composer?

composer.json

```
{  
  "name": "you/your-app",  
  "description": "Lorem ipsum",  
  "type": "project",  
  "license": "proprietary",  
  "require": {  
    "php": "^8.2",  
    "ext-pdo": "*",  
    "ext-mbstring": "*",  
    "erusev/parsedown": "^1.7.4",  
  },  
  "autoload": {  
    "psr-4": {  
      "App\\": "app/src/"  
    }  
  }  
}
```

**Unified** way to communicate application **information**



# Why use Composer?

composer.json

```
{
  "name": "you/your-app",
  "description": "Lorem ipsum",
  "type": "project",
  "license": "proprietary",
  "require": {
    "php": "^8.2",
    "ext-pdo": "*",
    "ext-mbstring": "*",
    "erusev/parsedown": "^1.7.4",
  },
  "autoload": {
    "psr-4": {
      "App \\": "app/src/"
    }
  }
}
```

**Declare** dependencies and enforce it



# Why use Composer?

composer.json

```
{
  "name": "you/your-app",
  "description": "Lorem ipsum",
  "type": "project",
  "license": "proprietary",
  "require": {
    "php": "^8.2",
    "ext-pdo": "*",
    "ext-mbstring": "*",
    "erusev/parsedown": "^1.7.4",
  },
  "autoload": {
    "psr-4": {
      "App \\": "app/src/"
    }
  }
}
```

**Dependency management**

# Why use Composer?

composer.json

```
{
  "name": "you/your-app",
  "description": "Lorem ipsum",
  "type": "project",
  "license": "proprietary",
  "require": {
    "php": "^8.2",
    "ext-pdo": "*",
    "ext-mbstring": "*",
    "erusev/parsedown": "^1.7.4",
  },
  "autoload": {
    "psr-4": {
      "App\\": "app/src/"
    }
  }
}
```

## Dependency management

- Resolves dependency tree
- Easy updates within ver constraints
- Autoload support
- Security auditing
- Statistics and analysis

# Why use Composer?

composer.json

```
{
  "name": "you/your-app",
  "description": "Lorem ipsum",
  "type": "project",
  "license": "proprietary",
  "require": {
    "php": "^8.2",
    "ext-pdo": "*",
    "ext-mbstring": "*",
    "erusev/parsedown": "^1.7.4",
  },
  "autoload": {
    "psr-4": {
      "App \\": "app/src/"
    }
  }
}
```

**Autoload support for custom classes, interfaces, traits, and enums**



# Why use Composer?

composer.json

```
{  
    "name": "you/your-app",  
    "description": "Lorem ipsum",  
    "type": "project",  
    "license": "proprietary",  
    "require": {  
        "php": "^8.2",  
        "ext-pdo": "*",  
        "ext-mbstring": "*",  
        "erusev/parsedown": "^1.7.4",  
    },  
    "autoload": {  
        "classmap": [  
            "app/src/"  
        ]  
    }  
}
```

**Autoload support for custom classes, interfaces, traits, and enums**

*Scans the entire folder*

## composer.json

```
{
  "name": "you/your-app",
  "description": "Lorem ipsum",
  "type": "project",
  "license": "proprietary",
  "require": {
    "php": "^8.2",
    "ext-pdo": "*",
    "ext-mbstring": "*",
    "erusev/parsedown": "^1.7.4",
  },
  "autoload": {
    "classmap": [
      "app/src/"
    ]
  }
}
```

```
app/
  - src/
  - templates/
    - footer.twig | php
    - header.twig | php
    - nav.twig | php
config/
  - settings.php
  - default.settings.php
.env
public/
  - index.php
  - css/
  - images/
  - js/
tests/
phpunit.xml
vendor/
  - phpmailer/phpmailer/
composer.json
composer.lock
package.json
.github/workflows/ci.yml
.gitignore
.gitattributes
.phpstan
```

## composer.json

```
{
  "name": "you/your-app",
  "description": "Lorem ipsum",
  "type": "project",
  "license": "proprietary",
  "require": {
    "php": "^8.2",
    "ext-pdo": "*",
    "ext-mbstring": "*",
    "erusev/parsedown": "^1.7.4",
  },
  "autoload": {
    "classmap": [
      "app/src/"
    ]
  }
}
```

```
app/
- src/
  - user.class.php
  - page.class.php
  - dashboard.class.php
  - UserInterface.php
  - MyClass.inc
```



```
<?php

require_once 'includes/db.php';
require_once 'includes/user.class.php';

?>
<?php if (user_logged_in()): ?>
    <?php require_once 'includes/page.class.php' ?>
    <?php
        $page = new Page();
        echo $page->getTitle();
    ?>
<?php endif ?>
```

```
<?php

require __DIR__ . '/vendor/autoload.php';
require_once 'includes/db.php';
require_once 'includes/user.class.php';

?>

<?php if (user_logged_in()): ?>
    <?php require_once 'includes/page.class.php' ?>
    <?php
        $page = new Page();
        echo $page →getTitle();
    ?>
<?php endif ?>
```

```
<?php

require __DIR__ . '/vendor/autoload.php';
require_once 'includes/db.php';
require_once 'includes/user.class.php';

?>

<?php if (user_logged_in()): ?>
    <?php require_once 'includes/page.class.php' ?>
    <?php
        $page = new Page();
        echo $page →getTitle();
    ?>
<?php endif ?>
```

```
<?php

require __DIR__ . '/vendor/autoload.php';
require_once 'includes/db.php';

?>
<?php if (user_logged_in()): ?>
    <?php
        $page = new Page();
        echo $page →getTitle();
    ?>
<?php endif ?>
```

```
<?php

require __DIR__ . '/../app/bootstrap.php';
require_once 'includes/db.php';

?>
<?php if (user_logged_in()): ?>
    <?php
        $page = new Page();
        echo $page->getTitle();
    ?>
<?php endif ?>
```

```
app/
- src/
- bootstrap.php
- templates/
    - footer.twig | php
    - header.twig | php
    - nav.twig | php
```



app/

- src/

- bootstrap.php

- templates/

  - footer.twig | php

  - header.twig | php

  - nav.twig | php

```
<?php
```

```
require __DIR__ . '/../vendor/autoload.php';
```



# Refactor only when

- File contains only one class/interface/trait/enum per file
- Does not modify state
- Constructor has no side-effects



# Error Handling



```
mysqli_connect('') or die('Error establishing database connection');
```

```
mysqli_connect('') or die('Error establishing database connection');
```

```
$settings = require __DIR__ . '/../../config/settings.php';  
$db = new PDO(  
    $settings['dsn'],  
    $settings['database']['username'],  
    $settings['database']['password']  
);
```

```
$settings = require __DIR__ . '/../../config/settings.php';  
$db = new PDO(  
    $settings['dsn'],  
    $settings['database']['username'],  
    $settings['database']['password']  
);
```

**PDOException** SQLSTATE[HY000] [2002] No connection could be made because the target machine actively refused it.

# Error display and logging

Do not display sensitive information to end users. Enable error logging and log all errors by setting following `php.ini` values.

Do not **display** errors

```
display_errors = Off
```

Do **log** errors

```
log_errors = On  
error_reporting = E_ALL
```



# Global error and exception handlers

```
<?php
```

```
set_error_handler( 'APPNAME_error_handler' );
```

```
function APPNAME_error_handler($error_level, $message): void {  
    // Log externally?  
    //  
}
```

# Global error and exception handlers

```
<?php

set_exception_handler([ExceptionHandler::class . '::handleException']);

class ExceptionHandler {
    public function __invoke() {
        // Log it?
        // Send 500 response?
        // Send HTML|JSON|XML response?
    }
}
```

# Global error and exception handlers

PHP 8.1 Goodness! [🔗](#)

```
<?php

set_exception_handler('ExceptionHandler::handleException( ... )');

class ExceptionHandler {
    public function __invoke() {
        // Log it?
        // Send 500 response?
        // Send HTML|JSON|XML response?
    }
}
```

# Global error and exception handlers

app/bootstrap.php

```
<?php
```

```
require __DIR__ . '/../vendor/autoload.php';
```

```
set_error_handler(\App\ErrorHandling\ErrorHandler::class);
```

```
set_exception_handler(\App\ErrorHandling\ExceptionHandler::class);
```

```
mysqli_connect('') or die('Error establishing database connection');
```

```
$connection = mysqli_connect(...);  
  
if ($connection === false) {  
    throw new RuntimeException('Error establishing database connection');  
}
```

```
mysqli_report(MYSQLI_REPORT_ERROR|MYSQLI_REPORT_STRICT);  
$connection = mysqli_connect('');
```

# Consolidate Classes and Functions





```
<?php
namespace App\ShoppingCart\PaymentMethods;

class IdealPaymentHandler implements PaymentProviderInterface {

    private HttpClient $httpClient;

    public function __construct(HttpClient $httpClient) {
        $this->httpClient = $httpClient;
    }

    public function authorizePayment(Customer $customer, Order $order): bool {
        $this->httpClient->post();
        // ...
    }
}
```

```
<?php
namespace App\ShoppingCart\PaymentMethods;

class IdealPaymentHandler implements PaymentProviderInterface {

    private HttpClient $httpClient;

    public function __construct(HttpClient $httpClient) {
        $this->httpClient = $httpClient;
    }

    public function authorizePayment(Customer $customer, Order $order): bool {
        $this->httpClient->post();
        // ...
    }
}
```

← One class, one file

App/src/IDEalPaymentHandler.php

```
<?php
namespace App\ShoppingCart\PaymentMethods;
class IDEalPaymentHandler implements PaymentProviderInterface {
    private HttpClient $httpClient;

    public function __construct(HttpClient $httpClient) {
        $this->httpClient = $httpClient;
    }

    public function authorizePayment(Customer $customer, Order $order): bool {
        $this->httpClient->post();
        // ...
    }
}
```

← One class, one file

App/src/ShoppingCart/PaymentMethods/IDealPaymentHandler.php

```
<?php
namespace App\ShoppingCart\PaymentMethods;

class IDealPaymentHandler implements PaymentProviderInterface {

    private HttpClient $httpClient;

    public function __construct(HttpClient $httpClient) {
        $this->httpClient = $httpClient;
    }

    public function authorizePayment(Customer $customer, Order $order): bool {
        $this->httpClient->post();
        // ...
    }
}
```

← One class, one file

App/src/ShoppingCart/PaymentMethods/IDEalPaymentHandler.php

```
<?php
namespace App\ShoppingCart\PaymentMethods;

class IDEalPaymentHandler implements PaymentProviderInterface {

    private HttpClient $httpClient;

    public function __construct(HttpClient $httpClient) {
        $this->httpClient = $httpClient;
    }

    public function authorizePayment(Customer $customer, Order $order): bool {
        $this->httpClient->post();
        // ...
    }
}
```

One class, one file

App/src/ShoppingCart/PaymentMethods/IDEalPaymentHandler.php

App/src/ShoppingCart/PaymentMethods/PaymentProviderInterface.php

```
<?php
namespace App\ShoppingCart\PaymentMethods;
class IDEalPaymentHandler implements PaymentProviderInterface {
    private HttpClient $httpClient;

    public function __construct(HttpClient $httpClient) {
        $this->httpClient = $httpClient;
    }

    public function authorizePayment(Customer $customer, Order $order): bool {
        $this->httpClient->post();
        // ...
    }
}
```

One class, one file

App/src/ShoppingCart/PaymentMethods/IDEalPaymentHandler.php

```
<?php
namespace App\ShoppingCart\PaymentMethods;

class IDEalPaymentHandler implements PaymentProviderInterface {
    private HttpClient $httpClient;

    public function __construct(HttpClient $httpClient) {
        $this->httpClient = $httpClient;
    }

    public function authorizePayment(Customer $customer, Order $order): bool {
        $this->httpClient->post();
        // ...
    }
}
```

One class, one file

Dependencies declared at constructor

App/src/ShoppingCart/PaymentMethods/IDEalPaymentHandler.php

```
<?php
namespace App\ShoppingCart\PaymentMethods;

class IDEalPaymentHandler implements PaymentProviderInterface {
    private HttpClient $httpClient;

    public function __construct(HttpClient $httpClient) {
        $this->httpClient = $httpClient;
    }

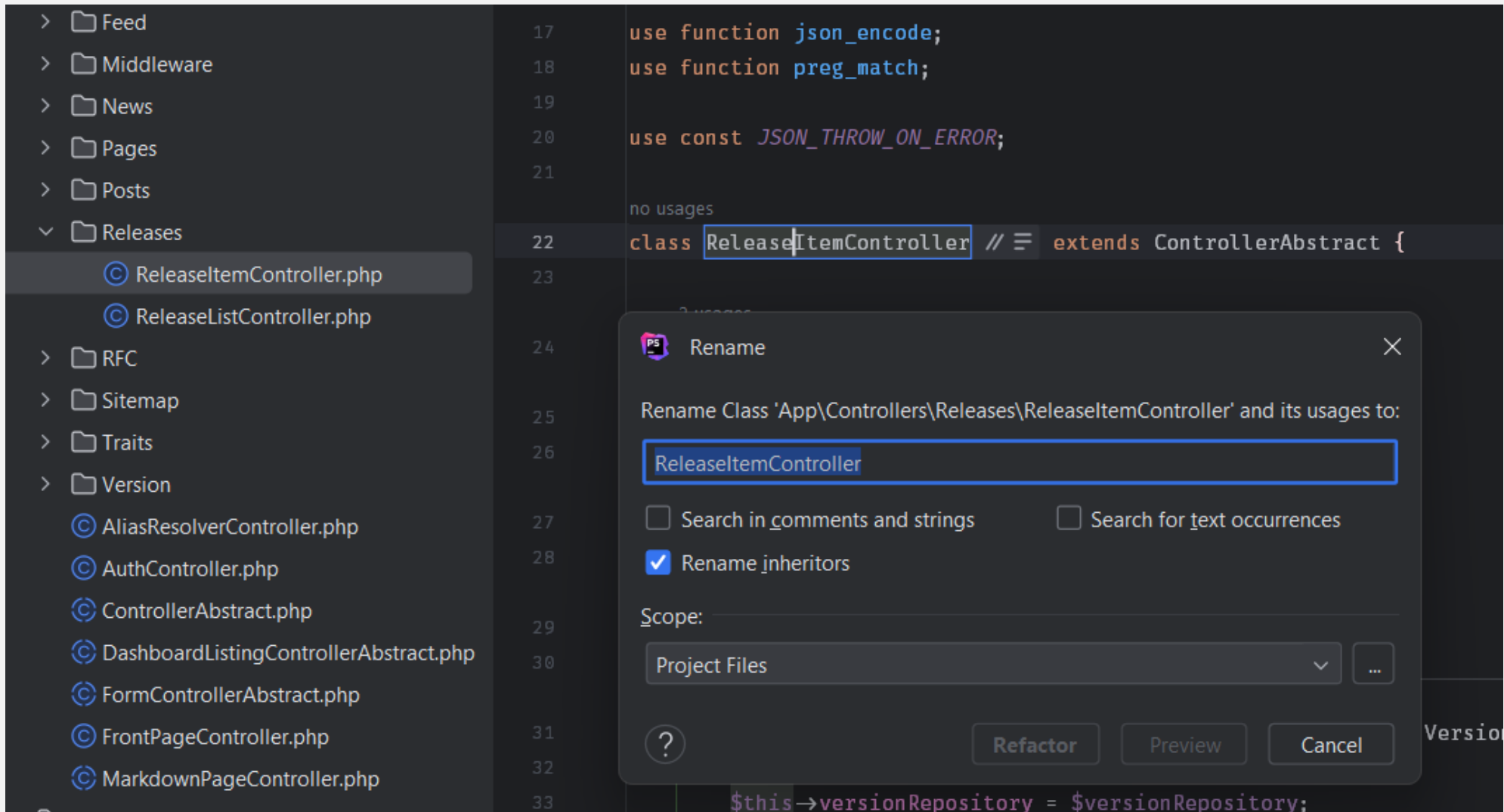
    public function authorizePayment(Customer $customer, Order $order): bool {
        $this->httpClient->post();
        // ...
    }
}
```

One class, one file

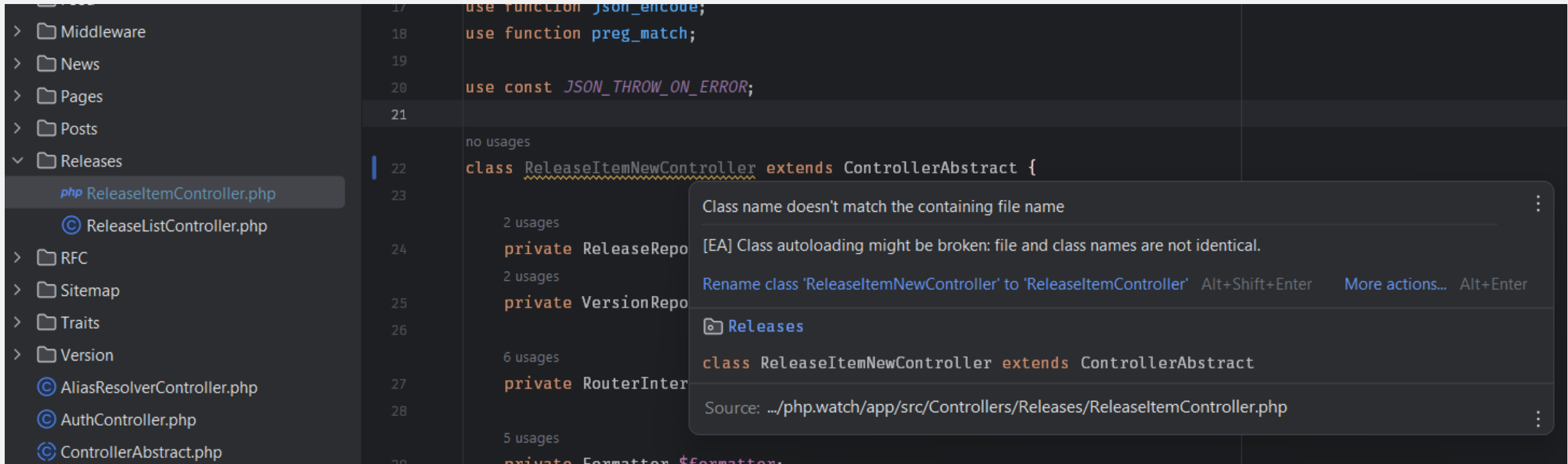
Dependencies declared at constructor

No global state or variables





PHPStorm has a built-in refactor tool that automatically updates all references to a class name



PHPStorm inspection if the file name does not match the class name

```
function cache_get() {}
```

```
function cache_set() {}
```

```
function cache_purge() {}
```

```
function cache_exists() {}
```

```
function cache_get() {}  
function cache_set() {}  
function cache_purge() {}  
function cache_exists() {}
```



```
namespace App\Cache;  
  
class CacheHandler {  
    public static function get() {}  
    public static function set() {}  
    public static function purge() {}  
    public static function exists() {}  
}
```



```
function cache_get() {}  
function cache_set() {}  
function cache_purge() {}  
function cache_exists() {}
```



```
namespace App\Cache;  
  
class CacheHandler {  
    public static function get() {}  
    public static function set() {}  
    public static function purge() {}  
    public static function exists() {}  
}
```



```
function cache_get() {}  
function cache_set() {}  
function cache_purge() {}  
function cache_exists() {}
```



```
namespace App\Cache;  
  
class CacheHandler {  
    public static function get() {}  
    public static function set() {}  
    public static function purge() {}  
    public static function exists() {}  
}
```

App/src/Cache/CacheHandler.php

```
function cache_get() {}  
function cache_set() {}  
function cache_purge() {}  
function cache_exists() {}
```



```
namespace App\Cache;  
  
class CacheHandler {  
    public static function get() {}  
    public static function set() {}  
    public static function purge() {}  
    public static function exists() {}  
}
```

App/src/Cache/CacheHandler.php

```
function cache_get() {}  
function cache_set() {}  
function cache_purge() {}  
function cache_exists() {}
```



```
namespace App\Cache;  
  
class CacheHandler {  
    public static function get() {}  
    public static function set() {}  
    public static function purge() {}  
    private static function exists() {}  
}
```

Possibility to add visibility restrictions



# Dependency Injection Container



*What if we consolidated these?*

```
$settings = require __DIR__ . '/../../config/settings.php';  
$db = new PDO(  
    $settings['dsn'],  
    $settings['database']['username'],  
    $settings['database']['password']  
);
```

```
$db = new PDO(  
    $settings['dsn'],  
    $settings['database']['username'],  
    $settings['database']['password']  
);
```

```
$mailer = new Mailer('smtp.example.com', 'username', 'password');
```

```
$cache = new MemoryCache();  
$cache->setPartition('myApp');
```

# Using PHP-DI

```
$container = new DI\Container([  
    // place your definitions here  
]);
```



# Using PHP-DI

```
$appContainer = new DI\Container([
    'database' => static function (Container $container) {
        return new PDO(
            $settings['database']['dsn'],
            $settings['database']['username'],
            $settings['database']['password']
        );
    },
]);
```



# Using PHP-DI

```
$appContainer = new DI\Container([  
    'database' => static function (Container $container) {  
        return new PDO(  
            $settings['database']['dsn'],  
            $settings['database']['username'],  
            $settings['database']['password']  
        );  
    },  
]);
```

```
$container->get('database');  
// PDO object
```





# Using PHP-DI

```
$appContainer = new DI\Container([
    'database' => static function (Container $container) {
        return new PDO(
            $settings['database']['dsn'],
            $settings['database']['username'],
            $settings['database']['password']
        );
    },
]);
```

*Service locator pattern*

```
$container →get('database');  
// PDO object
```

# Using PHP-DI

```
$appContainer = new DI\Container([
    'database' => static function (Container $container) {
        $settings = $container->get('settings')['database'];
        return new PDO(
            $settings['dsn'],
            $settings['username'],
            $settings['password']
        );
    },
]);
```



# Using PHP-DI

```
$appContainer = new DI\Container([
    'database' => static function (Container $container) {
        $settings = $container->get('settings')['database'];
        return new PDO(
            $settings['dsn'],
            $settings['username'],
            $settings['password']
        );
    },
]);
```



# Using PHP-DI

```
$appContainer = new DI\Container([
    'database' => static function (Container $container) {
        $settings = $container->get('settings')['database'];
        return new PDO(
            $settings['dsn'],
            $settings['username'],
            $settings['password']
        );
    },
    'mailer' => static function (Container $container): Mailer {
        return new Mailer('smtp.example.com', 'username', 'password');
    },
]);
```



# Using PHP-DI

```
$appContainer = new DI\Container([
    'database' => static function (Container $container) {
        $settings = $container->get('settings')['database'];
        return new PDO(
            $settings['dsn'],
            $settings['username'],
            $settings['password']
        );
    },
    'mailer' => static function (Container $container): Mailer {
        $s = $container->get('settings')['mailer'];
        return new Mailer($s['host'], $s['user'], $s['pw']);
    },
]);
```



# Using PHP-DI

```
$appContainer = new DI\Container([
    'database' => static function (Container $container) {
        $settings = $container->get('settings')['database'];
        return new PDO(
            $settings['dsn'],
            $settings['username'],
            $settings['password']
        );
    },
    'mailer' => static function (Container $container): Mailer {
        $s = $container->get('settings')['mailer'];
        return new Mailer($s['host'], $s['user'], $s['pw']);
    },
]);
```



# Using PHP-DI

```
$appContainer = new DI\Container([
    'database' => static function (Container $container) {
        $settings = $container->get('settings')['database'];
        return new PDO(
            $settings['dsn'],
            $settings['username'],
            $settings['password']
        );
    },
    'settings' => static function () {
        return include __DIR__ . '/../config/settings.php';
    },
    'mailer' => static function (Container $container): Mailer {
        $s = $container->get('settings')['mailer'];
        return new Mailer($s['host'], $s['user'], $s['pw']);
    },
]);
```



# Using PHP-DI

```
$appContainer = new DI\Container([
    'database' => static function (Container $container) {
        $settings = $container->get('settings')['database'];
        return new PDO(
            $settings['dsn'],
            $settings['username'],
            $settings['password']
        );
    },
    'settings' => static function () {
        return include __DIR__ . '/../config/settings.php';
    },
    'mailer' => static function (Container $container): Mailer {
        $s = $container->get('settings')['mailer'];
        return new Mailer($s['host'], $s['user'], $s['pw']);
    },
]);
```



## app/dependencies.php

```
return [  
    'database' => static function (Container $container) {  
        $settings = $container->get('settings')['database'];  
        return new PDO(  
            $settings['dsn'],  
            $settings['username'],  
            $settings['password']  
        );  
    },  
    'settings' => static function () {  
        return include __DIR__ . '/../config/settings.php';  
    },  
    'mailer' => static function (Container $container): Mailer {  
        $s = $container->get('settings')['mailer'];  
        return new Mailer($s['host'], $s['user'], $s['pw']);  
    },  
];
```

# Using PHP-DI

## app/dependencies.php

```
return [  
    'database' => static function (Container $container) {  
        $settings = $container->get('settings')['database'];  
        return new PDO(  
            $settings['dsn'],  
            $settings['username'],  
            $settings['password']  
        );  
    },  
    'settings' => static function () {  
        return include __DIR__ . '/../config/settings.php';  
    },  
    'mailer' => static function (Container $container): Mailer {  
        $s = $container->get('settings')['mailer'];  
        return new Mailer($s['host'], $s['user'], $s['pw']);  
    },  
];
```

## app/bootstrap.php

```
require __DIR__ . '/../vendor/autoload.php';  
  
set_error_handler(  
    \App\ErrorHandling\ErrorHandler::class  
);  
set_exception_handler(  
    \App\ErrorHandling\ExceptionHandler::class  
);
```



# Using PHP-DI

## app/dependencies.php

```
return [  
    'database' => static function (Container $container) {  
        $settings = $container->get('settings')['database'];  
        return new PDO(  
            $settings['dsn'],  
            $settings['username'],  
            $settings['password']  
        );  
    },  
    'settings' => static function () {  
        return include __DIR__ . '/../config/settings.php';  
    },  
    'mailer' => static function (Container $container): Mailer {  
        $s = $container->get('settings')['mailer'];  
        return new Mailer($s['host'], $s['user'], $s['pw']);  
    },  
];
```

## app/bootstrap.php

```
require __DIR__ . '/../vendor/autoload.php';  
  
set_error_handler(  
    \App\ErrorHandling\ErrorHandler::class  
);  
set_exception_handler(  
    \App\ErrorHandling\ExceptionHandler::class  
);  
  
$dependencies = require __DIR__ .  
    '/dependencies.php';  
  
$container = new DI\Container($dependencies);
```

# Separate Business Logic and View Logic



## ./profile.php

```
<?php
require __DIR__ . '/../../includes/db.php';

if (empty($_SESSION['uid'])) {
    die();
}

include __DIR__ . '/../../includes/user.class.php';
$user = new User($_SESSION['uid']);

if (empty($user)) {
    die();
}

if (!empty($_POST['email'])) {
    $user->email = $_POST['email'];
    $user->save();
}
?>
<div>
    Username: <?= $user->name ?>
    Email: <?= $user->email ?>
</div>
```

## ./profile.php

```
<?php
require __DIR__ . '/../../includes/db.php';

if (empty($_SESSION['uid'])) {
    die();
}

include __DIR__ . '/../../includes/user.class.php';
$user = new User($_SESSION['uid']);

if (empty($user)) {
    die();
}

if (!empty($_POST['email'])) {
    $user->email = $_POST['email'];
    $user->save();
}
?>
<div>
    Username: <?= $user->name ?>
    Email: <?= $user->email ?>
</div>
```

Changes application state

## ./profile.php

```
<?php
require __DIR__ . '/../../includes/db.php';

if (empty($_SESSION['uid'])) {
    die();
}

include __DIR__ . '/../../includes/user.class.php';
$user = new User($_SESSION['uid']);

if (empty($user)) {
    die();
}

if (!empty($_POST['email'])) {
    $user->email = $_POST['email'];
    $user->save();
}
?>
<div>
    Username: <?= $user->name ?>
    Email: <?= $user->email ?>
</div>
```

Changes application state

Error handling

## ./profile.php

```
<?php
require __DIR__ . '/../../includes/db.php';

if (empty($_SESSION['uid'])) {
    die();
}

include __DIR__ . '/../../includes/user.class.php';
$user = new User($_SESSION['uid']);

if (empty($user)) {
    die();
}

if (!empty($_POST['email'])) {
    $user->email = $_POST['email'];
    $user->save();
}
?>
<div>
    Username: <?= $user->name ?>
    Email: <?= $user->email ?>
</div>
```

Changes application state

Error handling

Database access



## ./profile.php

```
<?php
require __DIR__ . '/../../includes/db.php';

if (empty($_SESSION['uid'])) {
    die();
}

include __DIR__ . '/../../includes/user.class.php';
$user = new User($_SESSION['uid']);

if (empty($user)) {
    die();
}

if (!empty($_POST['email'])) {
    $user->email = $_POST['email'];
    $user->save();
}
?>
<div>
    Username: <?= $user->name ?>
    Email: <?= $user->email ?>
</div>
```

Changes application state

Error handling

Database access

## ./profile.php

```
<?php
require __DIR__ . '/../../includes/db.php';

if (empty($_SESSION['uid'])) {
    die();
}

include __DIR__ . '/../../includes/user.class.php';
$user = new User($_SESSION['uid']);

if (empty($user)) {
    die();
}

if (!empty($_POST['email'])) {
    $user->email = $_POST['email'];
    $user->save();
}
?>
<div>
    Username: <?= $user->name ?>
    Email: <?= $user->email ?>
</div>
```

Changes application state

Error handling

Database access

Data validation

# ./profile.php

```
<?php
require __DIR__ . '/../../includes/db.php';

if (empty($_SESSION['uid'])) {
    die();
}

include __DIR__ . '/../../includes/user.class.php';
$user = new User($_SESSION['uid']);

if (empty($user)) {
    die();
}

if (!empty($_POST['email'])) {
    $user->email = $_POST['email'];
    $user->save();
}
?>
<div>
    Username: <?= $user->name ?>
    Email: <?= $user->email ?>
</div>
```

Changes application state

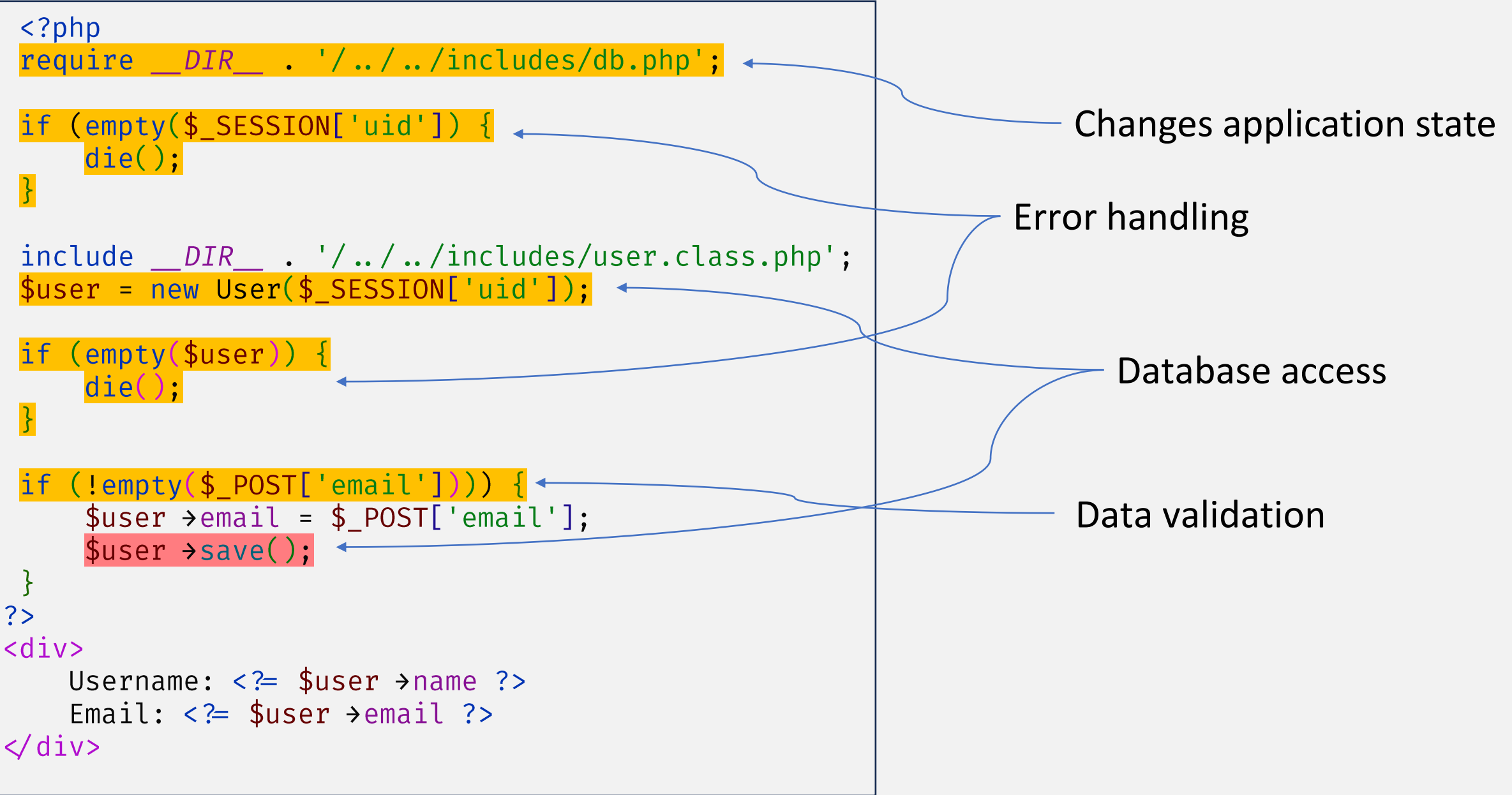
Error handling

Database access

Data validation



# ./profile.php



Separate Business Logic and View Logic



- Throw exceptions when appropriate
- Handle HTTP methods (GET/POST) separately
- Strictly no database calls or state changes in the view layer
- Consolidate data validation rules
- Abstract \$\_GET, \$\_POST, and other super globals

# Using PlatesPHP

```
$templates = new League\Plates\Engine('/path/to/templates');  
  
// Render a template  
echo $templates->render('profile', ['name' => $user]);
```



# Using PlatesPHP

```
$templates = new League\Plates\Engine('/path/to/templates');
```

```
// Render a template
```

```
echo $templates →render('profile', ['name' ⇒ $user]);
```

```
<div>
```

```
    Username: <?= $user →name ?>
```

```
    Email: <?= $user →email ?>
```

```
</div>
```



# Using PlatesPHP

```
$templates = new League\Plates\Engine('/path/to/templates');
```

```
// Render a template
```

```
echo $templates->render('profile', ['name' => $user]);
```

```
<div>
```

```
    Username: <?= $this->e($user->name) ?>
```

```
    Email: <?= $this->e($user->email) ?>
```

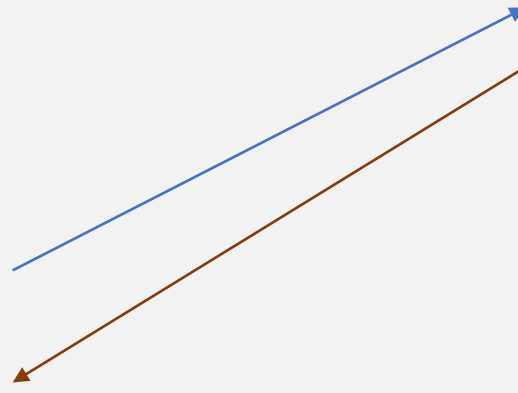
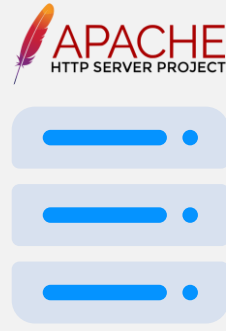
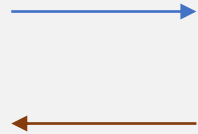
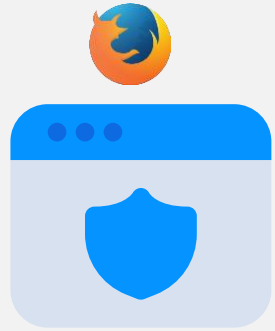
```
</div>
```





# Front Controller



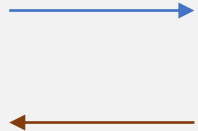
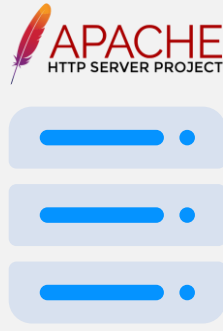
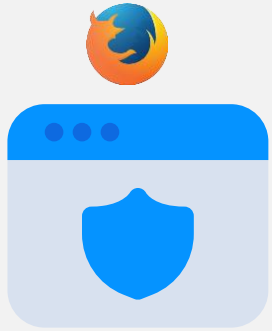


```
css/  
images/  
js/  
includes/  
  - db.inc  
templates/  
  - footer.php  
  - header.php  
  - nav.php  
libraries/  
  - phpmailer/  
    - phpmailer.php  
index.php  
login.php  
user.php  
register.php
```

GET css/styles.css

HTTP/2 200

```
html {  
  font-size: 16px;  
}
```

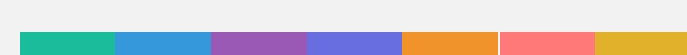


GET /login.php

HTTP/2 200

```
html {  
  font-size: 16px;  
}
```

```
app/  
- src/  
- templates/  
  - footer.twig | php  
  - header.twig | php  
  - nav.twig | php  
config/  
- settings.php  
- default.settings.php  
public/  
- index.php  
- css/  
- images/  
- js/  
vendor/  
  - phpmailer/phpmailer/  
composer.json  
composer.lock
```





```
<VirtualHost *:443>
  Protocols h2 http/1.1
  ServerName example.com
  DocumentRoot /var/www/example.com/public

  <Directory /var/www/php.watch/public>
    RewriteEngine on

    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteCond %{REQUEST_FILENAME} !-d
    RewriteCond %{REQUEST_FILENAME} !\.(jpg|png|css|js|txt)$
    RewriteRule ^ index.php [QSA,L]

  </Directory>
</VirtualHost>
```



```
<VirtualHost *:443>
  Protocols h2 http/1.1
  ServerName example.com
  DocumentRoot /var/www/example.com/public

  <Directory /var/www/php.watch/public>
    RewriteEngine on

    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteCond %{REQUEST_FILENAME} !-d
    RewriteCond %{REQUEST_FILENAME} !\.(jpg|png|css|js|txt)$
    RewriteRule ^ index.php [QSA,L]

  </Directory>
</VirtualHost>
```

```
<VirtualHost *:443>
  Protocols h2 http/1.1
  ServerName example.com
  DocumentRoot /var/www/example.com/public

  <Directory /var/www/php.watch/public>
    RewriteEngine on

    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteCond %{REQUEST_FILENAME} !-d
    RewriteCond %{REQUEST_FILENAME} !\.(jpg|png|css|js|txt)$
    RewriteRule ^ index.php [QSA,L]

  </Directory>
</VirtualHost>
```

```
app/
  - src/
  - templates/
    - footer.twig | php
    - header.twig | php
    - nav.twig | php
config/
  - settings.php
  - default.settings.php
public/
  - index.php
  - css/
  - images/
  - js/
vendor/
  - phpmailer/phpmailer/
composer.json
composer.lock
```

```
<VirtualHost *:443>
  Protocols h2 http/1.1
  ServerName example.com
  DocumentRoot /var/www/example.com/public

  <Directory /var/www/php.watch/public>
    RewriteEngine on

    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteCond %{REQUEST_FILENAME} !-d
    RewriteCond %{REQUEST_FILENAME} !\.(jpg|png|css|js|txt)$
    RewriteRule ^ index.php [QSA,L]

  </Directory>
</VirtualHost>
```

```
app/
  - src/
  - templates/
    - footer.twig | php
    - header.twig | php
    - nav.twig | php
config/
  - settings.php
  - default.settings.php
public/
  - index.php
  - css/
  - images/
  - js/
vendor/
  - phpmailer/phpmailer/
composer.json
composer.lock
```

```
<VirtualHost *:443>
  Protocols h2 http/1.1
  ServerName example.com
  DocumentRoot /var/www/example.com/public
```

```
<Directory /var/www/php.watch/public>
  RewriteEngine on
```

```
  RewriteCond %{REQUEST_FILENAME} !-f
  RewriteCond %{REQUEST_FILENAME} !-d
  RewriteCond %{REQUEST_FILENAME} !\.(jpg|png|css|js|txt)$
  RewriteRule ^ index.php [QSA,L]
```

```
</Directory>
</VirtualHost>
```

```
app/
  - src/
  - templates/
    - footer.twig | php
    - header.twig | php
    - nav.twig | php
config/
  - settings.php
  - default.settings.php
public/
  - index.php
  - css/
  - images/
  - js/
vendor/
  - phpmailer/phpmailer/
composer.json
composer.lock
```





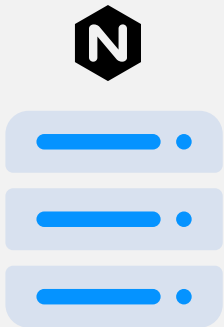
```
<VirtualHost *:443>
  Protocols h2 http/1.1
  ServerName example.com
  DocumentRoot /var/www/example.com/public

  <Directory /var/www/php.watch/public>
    RewriteEngine on

    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteCond %{REQUEST_FILENAME} !-d
    RewriteCond %{REQUEST_FILENAME} !\.(jpg|png|css|js|txt)$
    RewriteRule ^ index.php [QSA,L]

  </Directory>
</VirtualHost>
```



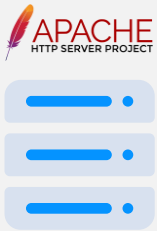


```
server {  
    listen 443;  
    server_name example.com;  
    root /var/www/example.com/public;  
  
    location / {  
        try_files $uri $uri/ /index.php?$query_string;  
    }  
}
```



```
example.com {  
    root * /var/www/example.com/public  
  
    file_server  
  
    php_fastcgi unix//run/php/php8.3-fpm.sock  
}
```





index.php



Request method



Request path



GET /login.php

GET /login.php



index.php

Request method

Request path



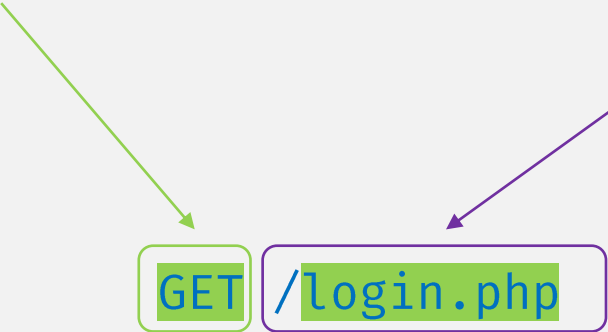
GET /login.php

```
GET login.php --> App\Controllers\User\LoginPageController  
GET register.php --> App\Controllers\User\RegisterPageController  
POST login.php --> App\Controller\User\LoginPageSubmitController  
POST register.php --> App\Controllers\User\RegisterPageSubmitController
```

index.php

Request method

Request path



GET /login.php

The diagram shows two arrows pointing to a rounded rectangular box. A green arrow from the text 'Request method' points to the word 'GET' inside the box. A purple arrow from the text 'Request path' points to the text '/login.php' inside the box. The 'GET' and '/login.php' are highlighted with green and purple backgrounds respectively.

```
GET login.php --> App\Controllers\User\LoginPageController  
GET register.php --> App\Controllers\User\RegisterPageController  
POST login.php --> App\Controller\User\LoginPageSubmitController  
POST register.php --> App\Controllers\User\RegisterPageSubmitController
```

# Using Slim PHP

```
$app →get('/login.php', App\Controllers\User\LoginPageController::class);
```

```
$app →get('/register.php', App\Controllers\User\RegisterPageController::class);
```

```
$app →post('login.php', App\Controllers\User\LoginPageSubmitController::class);
```

```
$app →post('register.php', App\Controllers\User\RegisterPageSubmitController::class);
```

# Using Slim PHP

routes.php

```
$app →get('/login.php', App\Controllers\User\LoginPageController::class);
```

```
$app →get('/register.php', App\Controllers\User\RegisterPageController::class);
```

```
$app →post('login.php', App\Controllers\User\LoginPageSubmitController::class);
```

```
$app →post('register.php', App\Controllers\User\RegisterPageSubmitController::class);
```



## app/dependencies.php

```
return [  
    'database' => static function (Container $container) {  
        $settings = $container->get('settings')['database'];  
        return new PDO(  
            $settings['dsn'],  
            $settings['username'],  
            $settings['password']  
        );  
    },  
    App\Controllers\User\LoginPageController::class => static function(Container $container) {  
        return new App\Controllers\User\LoginPageController(  
            new TemplateEngine(),  
            $container->get('database'),  
        );  
    }  
];
```

## app/dependencies.php

```
return [  
    'database' => static function (Container $container) {  
        $settings = $container->get('settings')['database'];  
        return new PDO(  
            $settings['dsn'],  
            $settings['username'],  
            $settings['password']  
        );  
    },  
    App\Controllers\User\LoginPageController::class => static function(Container $container) {  
        return new App\Controllers\User\LoginPageController(  
            new TemplateEngine(),  
            $container->get('database'),  
        );  
    }  
];
```

*Dependency Injection!*  
~~*Service locator pattern*~~

```

namespace App\Controllers\User;

use Psr\Http\Message\ResponseInterface;

class LoginPageController {

    private TemplateEngine $templateEngine;
    private PDO $database;

    public function __construct(
        TemplateEngine $templateEngine,
        PDO $database)
    {
        $this->engine = $templateEngine;
        $this->database = $database;
    }

    public function __invoke(): ResponseInterface {
        if (! $db->query( ... )) {
            throw new UserNotFoundException();
        }

        return $this->templateEngine->process('login-form');
    }
}

```

Dependencies clearly declared  
 Separate business and presentation logic  
 Throws exceptions on unexpected cases

```
namespace App\Controllers\User;

use Psr\Http\Message\ResponseInterface;

class LoginPageController {

    private TemplateEngine $templateEngine;
    private PDO $database;

    public function __construct(
        TemplateEngine $templateEngine,
        PDO $database)
    {
        $this->engine = $templateEngine;
        $this->database = $database;
    }

    public function __invoke(): ResponseInterface {
        if (! $db->query(...)) {
            throw new UserNotFoundException();
        }

        return $this->templateEngine->process('login-form');
    }
}
```

**Dependencies clearly declared**

Separate business and presentation logic

Throws exceptions on unexpected cases

```

namespace App\Controllers\User;

use Psr\Http\Message\ResponseInterface;

class LoginPageController {

    private TemplateEngine $templateEngine;
    private PDO $database;

    public function __construct(
        TemplateEngine $templateEngine,
        PDO $database)
    {
        $this->engine = $templateEngine;
        $this->database = $database;
    }

    public function __invoke(): ResponseInterface {
        if (! $db->query( ... )) {
            throw new UserNotFoundException();
        }

        return $this->templateEngine->process('login-form');
    }
}

```

Dependencies clearly declared  
**Separate business and presentation logic**  
 Throws exceptions on unexpected cases

```

namespace App\Controllers\User;

use Psr\Http\Message\ResponseInterface;

class LoginPageController {

    private TemplateEngine $templateEngine;
    private PDO $database;

    public function __construct(
        TemplateEngine $templateEngine,
        PDO $database)
    {
        $this->engine = $templateEngine;
        $this->database = $database;
    }

    public function __invoke(): ResponseInterface {
        if (! $db->query(...)) {
            throw new UserNotFoundException();
        }

        return $this->templateEngine->process('login-form');
    }
}

```

Dependencies clearly declared  
 Separate business and presentation logic  
**Throws exceptions on unexpected cases**

```
app/  
  - src/  
  - dependencies.php  
  - routes.php  
  - bootstrap.php  
  - templates/  
    - footer.twig | php  
    - header.twig | php  
    - nav.twig | php  
config/  
  - settings.php  
  - default.settings.php  
public/  
  - index.php  
  - css/  
  - images/  
  - js/  
vendor/  
  - phpmailer/phpmailer/  
composer.json  
composer.lock
```

```
app/  
- src/  
- dependencies.php  
- routes.php  
- bootstrap.php  
- templates/  
  - footer.twig | php  
  - header.twig | php  
  - nav.twig | php  
config/  
- settings.php  
- default.settings.php  
public/  
- index.php  
- css/  
- images/  
- js/  
vendor/  
- phpmailer/phpmailer/  
composer.json  
composer.lock
```

```
return [  
    'database' => static function (Container $container) {  
        $settings = $container->get('settings')['database'];  
        return new PDO(  
            $settings['dsn'],  
            $settings['username'],  
            $settings['password']  
        );  
    },  
    'settings' => static function () {  
        return include __DIR__ . '/../config/settings.php';  
    },  
    'mailer' => static function (Container $container): Mailer {  
        $s = $container->get('settings')['mailer'];  
        return new Mailer($s['host'], $s['user'], $s['pw']);  
    },  
];
```



```
app/  
  - src/  
  - dependencies.php  
  - routes.php  
  - bootstrap.php  
  - templates/  
    - footer.twig | php  
    - header.twig | php  
    - nav.twig | php  
config/  
  - settings.php  
  - default.settings.php  
public/  
  - index.php  
  - css/  
  - images/  
  - js/  
vendor/  
  - phpmailer/phpmailer/  
composer.json  
composer.lock
```

```
$app →get('/login.php',  
  App\Controllers\User\LoginPageController::class  
);  
  
$app →get('/register.php',  
  App\Controllers\User\RegisterPageController::class  
);  
  
$app →post('login.php',  
  App\Controllers\User\LoginPageSubmitController::class  
);  
  
$app →post('register.php',  
  App\Controllers\User\RegisterPageSubmitController::class  
);
```

```
app/  
- src/  
- dependencies.php  
- routes.php  
- bootstrap.php  
- templates/  
  - footer.twig | php  
  - header.twig | php  
  - nav.twig | php  
config/  
- settings.php  
- default.settings.php  
public/  
- index.php  
- css/  
- images/  
- js/  
vendor/  
- phpmailer/phpmailer/  
composer.json  
composer.lock
```

```
use App\Slim\Container;  
use Slim\App;  
  
require __DIR__ . '/../vendor/autoload.php';  
  
set_error_handler(  
    \App\ErrorHandling\ErrorHandler::class  
);  
set_exception_handler(  
    \App\ErrorHandling\ExceptionHandler::class  
);  
  
$settings = require __DIR__ . '/../config/settings.php';  
  
$container = new Container($settings);  
$app = new App($container);  
require __DIR__ . '/../app/dependencies.php';  
require __DIR__ . '/../app/routes.php';  
  
$app →run();
```

Application logic

Configuration

Assets and Static Files

Tests

Dependencies



```
app/  
  - src/  
  - templates/  
    - footer.twig | php  
    - header.twig | php  
    - nav.twig | php  
config/  
  - settings.php  
  - default.settings.php  
public/  
  - index.php  
  - css/  
  - images/  
  - js/  
tests/  
vendor/  
  - phpmailer/phpmailer/  
composer.json  
composer.lock
```

# Unit and Integration Tests

- PHPUnit, PestPHP
- Behat
- Codeception
- Selenium with PHP Web Driver

# External tools

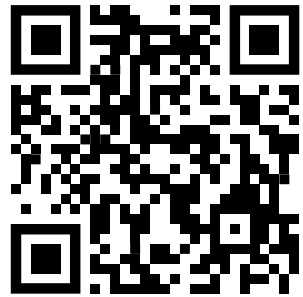
- Grunt/Webpack/Gulp for HTML/CSS/JS minification and aggregation
- Headless Chrome to generate PDF instead of DOMPdf
- GitHub Actions / GitLab CI to replace Jenkins
- Docker Compose files to quickly setup dev/staging environments
- Virtual Private Clouds/Networks to bypass web service authentication
- FFI

# Further Resources

- <https://aye.sh/talk/dpc2023-modernize-php>
- <https://www.martinfowler.com/eaCatalog/index.html>
- <https://refactoring.com/>
- <https://www.oreilly.com/library/view/working-effectively-with/0131177052/>
- <https://www.manning.com/books/re-engineering-legacy-software>
- <https://www.packtpub.com/product/modernizing-legacy-applications-in-php/9781787124707>
- <https://phpdelusions.net/articles/paths>
- [https://phpdelusions.net/articles/error\\_reporting](https://phpdelusions.net/articles/error_reporting)
- [https://en.wikipedia.org/wiki/The\\_Mythical\\_Man-Month](https://en.wikipedia.org/wiki/The_Mythical_Man-Month)

# Questions?

No question is too small



**#DPC23 @Ayeshlive** [ayesh@php.watch](mailto:ayesh@php.watch)

<https://aye.sh/talk/dpc2023-modernize-php>

arigatô paldies dziękuję Ďakujem tak  
diolch dankie děkuji mahalo kop khun  
cảm ơn bạn хвала شکرالك köszönöm  
a dank gràcies ngiyabonga tänan Баярлалаа dhanyavād  
Дякую ευχαριστώ **THANK YOU** Благодарам  
спасибо takk благодаря  
grazie Mh'gōi Dank u Благодаря ти gracias  
mulțumesc takk ස්තූතියි ačiū நன்றி הודת.  
danke teşekkür ederim choukrane faleminderit 谢谢  
ՀնրհաԿալըԹյոԼս obrigado kiitos  
terima kasih hvala grazzi



# THANK YOU

smakelijk eten!

arigatô      paldies      dziękuję      Ďakujem      tak  
diolch      dankie      хвала      Баярлалаа      شكرا لك      kop khun  
cảm ơn bạn      ngiyabonga      tänan      köszönöm  
a dank      gràcies      dhanyavād  
Дякую      ευχαριστώ      Благодарам  
спасибо      takk      благодаря  
grazie      Mh'gōi      Dank u      ačiū      Благодаря ти      gracias  
mulțumesc      danke      takk      ස්තූතියි      faleminderit      நன்றி      הודת.  
teşekkür ederim      choukrane      obrigado      kiitos      谢谢  
Հնրհապալութիւնս      terima kasih      hvala      grazzi



How to eat an elePHPant

# Modernize Legacy PHP Applications

