

TEXT

---

## ABOUT ME

- ❖ Andrew Woods
- ❖ [andrewwoods.net](http://andrewwoods.net)
- ❖ @awoods
- ❖ Slides: <http://bit.ly/2srPNC4>
- ❖ Code: <http://bit.ly/2igcHak>



<http://pnwphp.com>



***pnwphp***

# NULL OBJECT PATTERN

---

Portland PHP Meetup, 2017 Jun 20

Andrew Woods

TEXT

---

## THE CODE

<https://github.com/andrewwoods/null-pattern-talk>

```
<?php
```

```
$account = new stdClass();
```

```
$account->name = 'Portland';
```

```
?>
```

```
<span>Hello <?=$account->name ?></span>
```

```
<?php
```

```
$account = null;
```

```
?>
```

```
<span>Hello <?=$account->name ?></span>
```

PHP Notice: Trying to get property of non-object in /Users/awoods/code/example-01b.php on line 9

PHP Stack trace:

PHP 1. {main}() /Users/awoods/code/example-01b.php:0

Notice: Trying to get property of non-object in /Users/awoods/code/example-01b.php on line 9

Call Stack:

0.0019 229944 1. {main}() /Users/awoods/code/example-01b.php:0

## PROBLEMS CAUSED BY NULL

- ❖ Need to add conditional code
- ❖ Errors get thrown when not used properly



TEXT

---

## CONDITIONALS

- ❖ Simple and it works
- ❖ It's ok when once or twice
- ❖ dirties your code

```
$account = getUserById($userId);
```

```
if ($account != null) {  
    echo "Hello {$account->name}\n";  
}
```

# NULL OBJECT PATTERN

---

## WHAT IS IT?

- ❖ It's a design pattern
- ❖ help handle the null case more smoothly
- ❖ avoid special if blocks for do nothing code

## HOW DO YOU USE IT?

- ❖ Often used with Factory Pattern
- ❖ Return an object with empty values instead of null
- ❖ Code to an interface
- ❖ Multiple classes

```
$admin = AccountFactory::create(
    'admin', 'awoods', 'andrew@seaphp.com'
);
$author = AccountFactory::create(
    'user', 'alena', 'alena@pdxphp.com'
);
$guest = AccountFactory::create(
    'manager', 'jason', 'jason@bourne.com'
);

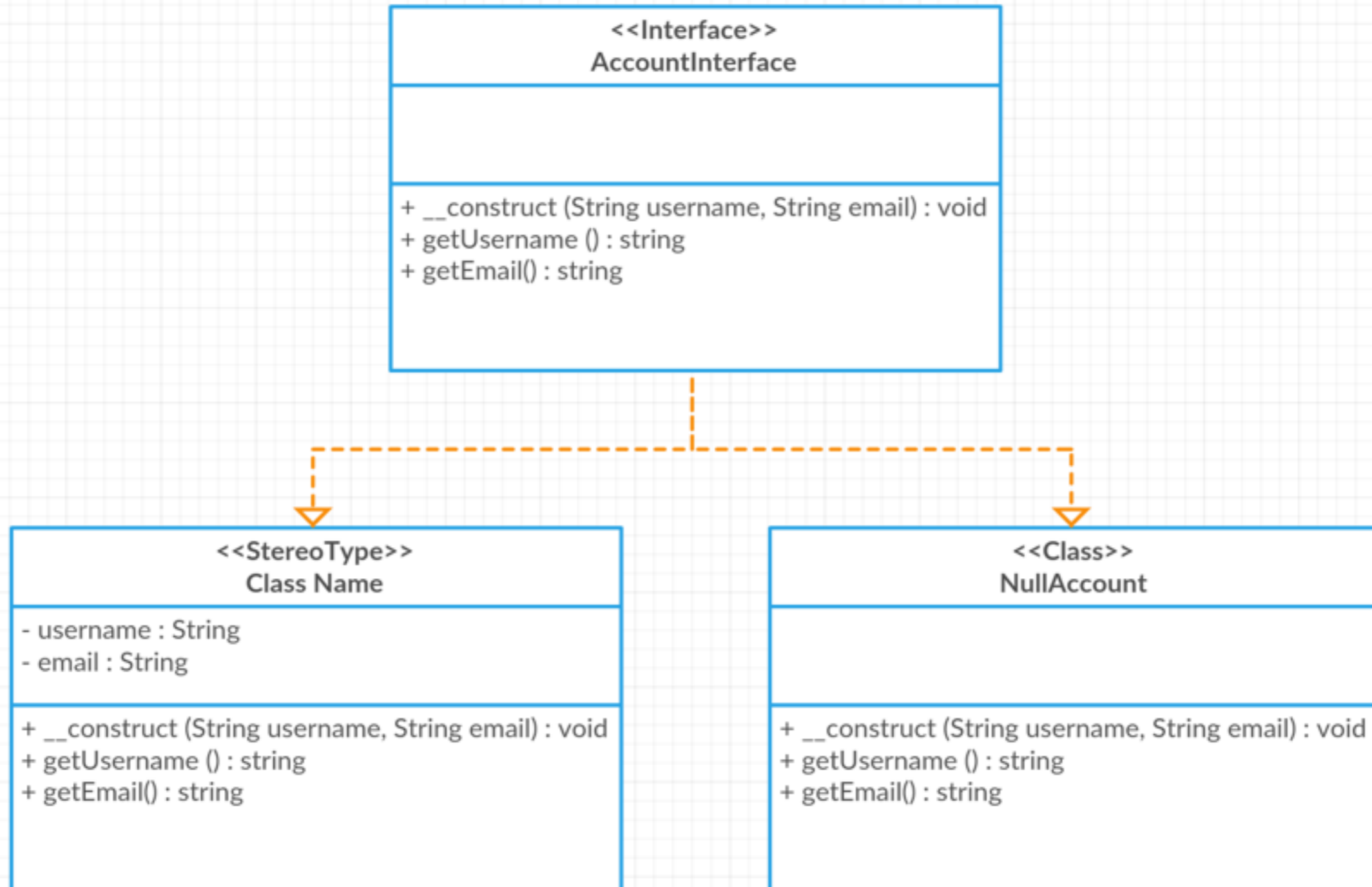
print_r($admin);
print_r($author);
print_r($guest);
```

```
class AccountFactory
{
    public static function create($role, $login, $email)
    {
        if ('user' === $role) {
            return new UserAccount($login, $email);
        }

        if ('admin' === $role) {
            return new AdminAccount($login, $email);
        }

        return null;
    }
}
```

## CLASS DIAGRAM





```
class AccountFactory
{
    public static function create($role, $login, $email)
    {
        if ('user' === $role) {
            return new UserAccount($login, $email);
        }

        if ('admin' === $role) {
            return new AdminAccount($login, $email);
        }

        return new NullAccount($login, $email);
    }
}
```

```
class NullAccount implements AccountInterface
{
    public function getUsername()
    {
        return "";
    }

    public function getEmail()
    {
        return "";
    }
}
```

# USING DATABASES

---

## PHP STANDARD CLASS

2 Reasons I used stdClass in my example

- ❖ Easy to Create
- ❖ MySQL uses it for results with `fetch_object()`

## NORMAL SELECT RESULT

- ❖ Prepared statement
- ❖ Bind parameter
- ❖ Check the number of items found
- ❖ Return NULL for Zero records
- ❖ see `example-sql-result.php`

```
$query = $db->prepare($sql);  
if (! $query){  
    die();  
}  
$query->bind_param('i', $id);  
$query->execute();  
$result = $query->get_result();  
if (0 === $result->num_rows){  
    return null;  
}  
$data = $result->fetch_object();  
$query->close();
```

# NULL OBJECT PATTERN SELECT RESULT

- ❖ Same approach as normal SQL result
- ❖ The function acts like a factory
- ❖ SQL fields map to public properties
- ❖ Provides consistency between code and db layers
- ❖ see `example-sql-result-alt.php`

```
$query = $db->prepare($sql);  
if (! $query){  
    die();  
}  
$query->bind_param('i', $id);  
$query->execute();  
  
$result = $query->get_result();  
if ($result->num_rows){  
    return new NullAccount();  
}  
  
$data = $result->fetch_object('UserAccount');  
$result->close();  
$query->close();
```





## MAGIC

- ❖ Implement `__set()`
- ❖ Use with related fields but different names
  - ▶ Zip vs Zipcode vs Postal Code
  - ▶ Login vs Username

## CONSTRAINTS

- ❖ No Constructor Injection
- ❖ `__set` does not get called by `fetch_object`
- ❖ see `example-sql-result-null-object.php`

```
$result = $query->get_result();  
if (0 === $result->num_rows){  
    $nullEmail = '';  
    $nullLogin = 'guest';  
  
    $nullAccount = new NullAccount($nullLogin, $nullEmail);  
  
    return $nullAccount;  
}  
  
$record = $result->fetch_object();  
  
$data = new UserAccount($record->login, $record->email);  
$data->load($record);
```

TEXT

---

# THANK YOU

❖ Andrew Woods

❖ [andrewwoods.net](http://andrewwoods.net)

❖ @awoods

❖ Code: <http://bit.ly/2igcHak>

