# PHP/MySQL development

## a Twitter-like application

Maxime Martineau

Polytech Tours Département Informatique

December 13, 2016

# Outline

The project

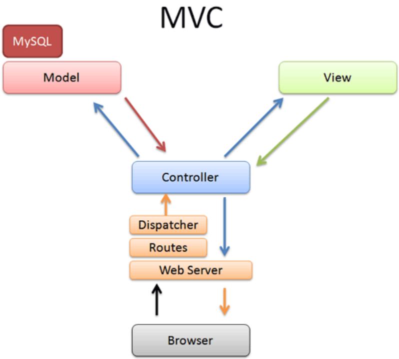
PHP

PHP and MySQL

Twitter application

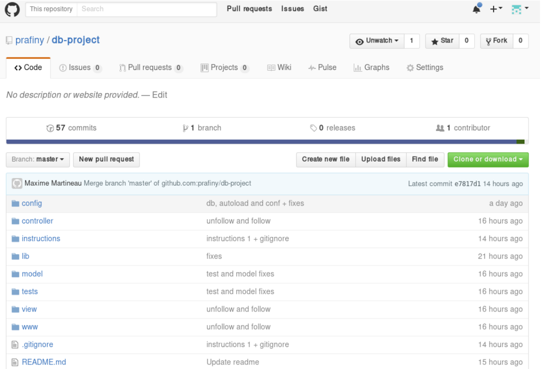
- Users : can post, follow a user, . . .
- Post : message from a user
- Notification system
- Hashtag : to mark a given topic (ex: #DevelopersBeLike)

# Project structure



# Code

http://github.com/prafiny/db-project



prafiny / db-project

No description or website provided. — Edit

57 commits · 1 branch · 0 releases · 1 contributor

Maxime Martineau Merge branch 'master' of github.com:prafiny/db-project — Latest commit e7817d1 14 hours ago

| | | |
|---|---|---|
| config | db, autoload and conf + fixes | a day ago |
| controller | unfollow and follow | 16 hours ago |
| instructions | instructions 1 + gitignore | 14 hours ago |
| lib | fixes | 21 hours ago |
| model | test and model fixes | 16 hours ago |
| tests | test and model fixes | 16 hours ago |
| view | unfollow and follow | 16 hours ago |
| www | unfollow and follow | 16 hours ago |
| .gitignore | instructions 1 + gitignore | 14 hours ago |
| README.md | Update readme | 15 hours ago |

# Code

`http://github.com/prafiny/db-project`

→ Instructions in `instructions/` → Read 0setup.pdf for environment installation → Read 1user.pdf for first work
→ Models to be completed (files in `model`)

# PHP : Data

- Declare a variable

```php
$a = 23; // $a is an int
$b = 2.3; // $b is a float
$c = true; // $c is a boolean
$d = array("a", 1, "b"); // $d is an array
```

# PHP: Data

- Deal with arrays

```php
$a = array("a", "b", 1);
echo $a[0]; // prints "a"
$a[0] = "A";
echo $a[0]; // ?
$a[] = 2; // $a == ?
```

- Associative arrays

```php
$fruit = array("name" => "orange", "color" => "orange");
echo $fruit["name"]; // ?
echo $fruit["color"]; // ?
$fruit["color"] = "black";
```

# PHP : Data

- Create and use a stdClass object

```
$fruit = (object) array("name" => "orange", "color" =>
    "orange");
echo $fruit->name; // ?
echo $fruit->color; // ?
```

# PHP : Controls

- If stanzas

```php
if($a == 1) {
    echo "yoohoo :D";
}
else {
    echo "oh :(";
}
```

# PHP : Controls

- While loops

```php
$i = 0;
while($i <= 10) {
    echo $i;
    $i++;
}
```

- For loops

```php
for($i = 0; $i <= 10; $i++) {
    echo $i;
}
```

- Foreach loops

```php
$arr = array(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
foreach($arr as $i) {
    echo $i;
}
```

# PHP : Functions

- Create a function

```
function inverse($nb) {
    return 1/$nb;
}
```

- Use a function

```
echo inverse(3); // ?
```

# PHP : Error handling with exceptions

- Throw an exception

```
function inverse($nb) {
    if($nb == 0) { throw new Exception(’Division by
        zero.’); }
    return 1/$nb;
}
echo inverse(0);
echo "I can has display ?"; // ?
```

# PHP : Error handling with exceptions

- Throw an exception

```
function inverse($nb) {
    if($nb == 0) { throw new Exception(’Division by
        zero.’); }
    return 1/$nb;
}
echo inverse(0);
echo "I can has display ?"; // ?
```

- Catch an exception

```
try {
    echo inverse(0);
}
catch(Exception $e) {
    echo $e->getMessage();
}
echo "I can has display ?"; // ?
```

# PHP and MySQL : PDO

## PHP Data Object

- Connection

```
$db = new PDO('mysql:host=SERVER_ADDR;dbname=DB_DBNAME',
    DB_USER, DB_PASS);
```

- Queries

```
$sql = 'SELECT name, color, calories FROM fruit ORDER BY
    name';
foreach ($db->query($sql) as $row) {
   print $row['name'] . "\t";
   print $row['color'] . "\t";
   print $row['calories'] . "\n";
}
```

# PHP and MySQL : PDO

PHP Data Object

- Prepare query (with variables)

```
$sql = ’SELECT name, colour, calories FROM fruit WHERE
    calories < :calories AND colour = :colour’;
$sth = $dbh->prepare($sql);
$sth->execute(array(’:calories’ => 150, ’:colour’ =>
    ’red’));
```

# PHP and MySQL : PDO

PHP Data Object

- Prepare query (with variables)

```
$sql = ’SELECT name, colour, calories FROM fruit WHERE
    calories < :calories AND colour = :colour’;
$sth = $dbh->prepare($sql);
$sth->execute(array(’:calories’ => 150, ’:colour’ =>
    ’red’));
```

- Error handling (with Exceptions)

```
try {
    $sql = ’SELECT name, color, calories FROM fruit ORDER
        BY name’;
    foreach ($db->query($sql) as $row)
    { do stuff; }
} catch (\PDOException $e) { print $e->getMessage(); }
```