PHP/MySQL development

a Twitter-like application

Maxime Martineau

Polytech Tours Département Informatique

December 15, 2016

Outline

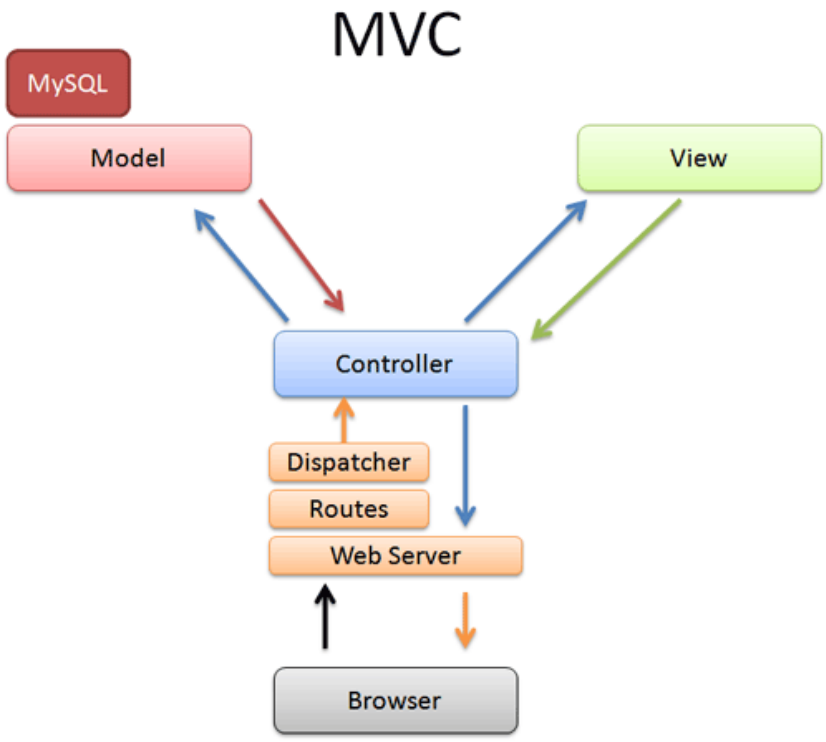
Twitter application

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

Frame

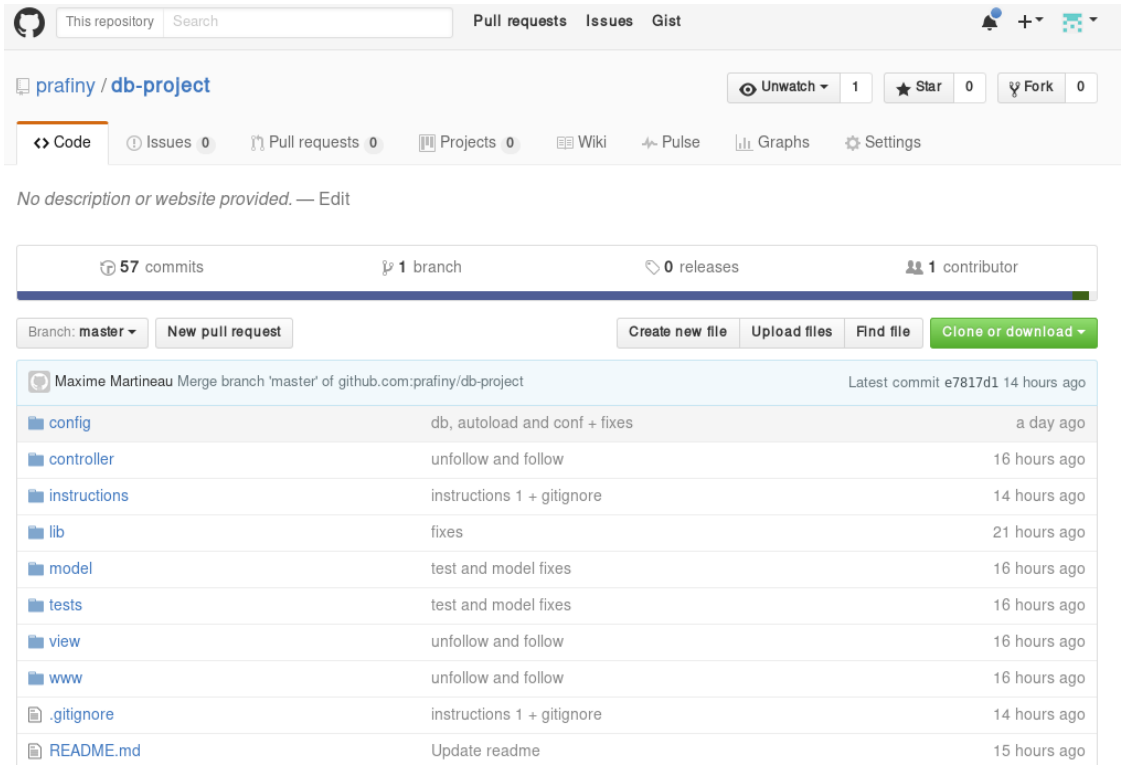
- 6 practical works (12h)
- Work in pairs
- Code to give back at the end (the database schema + the `model/` files)

Project structure



Code

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



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" => "strawberry", "color" => "red");
echo $fruit["name"]; // ?
echo $fruit["color"]; // ?
$fruit["color"] = "black";
```

PHP : Data

- Create and use a stdClass object

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

PHP : Controls

- If stanzas

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

PHP : Controls

- While loops

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

- For loops

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

- Foreach loops

```
$arr = array(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
foreach($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);
$sth->execute(array(':calories' => 150, ':colour' =>
    'red'));
foreach($sth->fetchAll() as $row) {
    print $row['name'] . "\t";
    print $row['color'] . "\t";
    print $row['calories'] . "\n";
}
```

PHP and MySQL : PDO

- Error handling (with Exceptions)

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

Code

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

1. Application can be run in browser (see instructions)
2. Unit tests can be run to check the functions (see instructions)
3. Instructions (in the folder `instructions/`)
4. The goal is to fill out the models (files in folder `model/`)
   - 4.1 Read 0setup.pdf for environment installation
   - 4.2 Read 1user.pdf for first work