PHP - MVC

Using Model-View-Controller in PHP

Aplicações para a Internet

Engenharia Informática – 2015/2016

instituto politécnico de leiria

Copyright

Author(s):

- **Vítor Carreira (vitor.carreira@ipleiria.pt)**

Contributor(s):

- **Fernando Silva** (fernando.silva@ipleiria.pt)
- **Alexandrino Gonçalves (alex@ipleiria.pt)**
- **Norberto Henriques (norberto.henriques@ipleiria.pt)**

Revised on: March, 2016

- **Fernando Silva** (fernando.silva@ipleiria.pt)

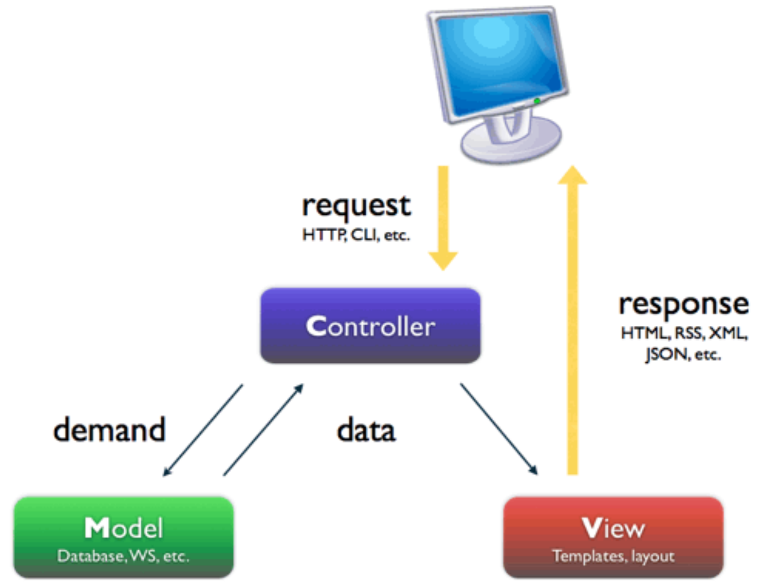
Syllabus

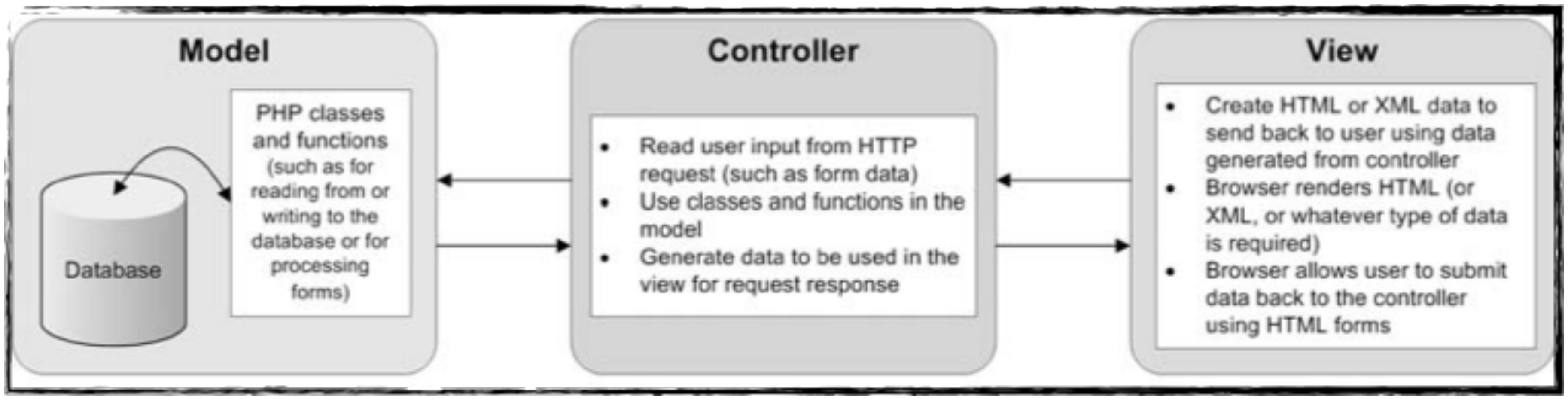
- Model-View-Controller
- PHP Frameworks

MVC 1/10

- Model–view–controller (MVC) is an architectural pattern that isolates "domain logic" from user interface
  - **The model** is responsible to manage the data; it stores and retrieves entities used by an application, usually from a database, and contains the logic implemented by the application
  - **The view** (presentation) is responsible to display the data provided by the model in a specific format (html, xml, etc). Multiple views can exist for a single model for different purposes
  - **The controller** handles the model and view layers to work together. The controller receives a request from the client, invoke the model to perform the requested operations and send the data to the View

MVC 2/10





PHP classes and functions (such as for reading from or writing to the database or for processing forms)

- Read user input from HTTP request (such as form data)
- Use classes and functions in the model
- Generate data to be used in the view for request response

- Create HTML or XML data to send back to user using data generated from controller
- Browser renders HTML (or XML, or whatever type of data is required)
- Browser allows user to submit data back to the controller using HTML forms

MVC 3/10

- Practical example:
  - Create a web page that shows a list of articles stored on a array
    - Note that the data could also be on a database
  - Solution 1 - No MVC
  - Solution 2 - MVC architecture

MVC 4/10

- Solution 1 - No MVC

```
<?php
    $articles = ['Understanding MVC in PHP' => 'This article covers the basics of MVC web frameworks.',
     'MVC in PHP' => 'The model view controller pattern is the most used pattern for today’s world web applications.'];
    $title = "List of Articles";
?>
<!doctype html>
<html>
    <head>
        <meta charset="utf-8">
        <title><?= $title ?></title>
    </head>

    <body>
    <table>
    <thead>
        <tr>
            <th>Title</th>
            <th>Content</th>
        </tr>
    </thead>
    <tbody>
    <?php foreach ($articles as $title => $content) { ?>
        <tr>
            <td><?= htmlspecialchars($title) ?></td>
            <td><?= htmlspecialchars($content) ?></td>
        </tr>
    <?php } ?>
    </tbody>
    </table>
    </body>
</html>
```

MVC 5/10

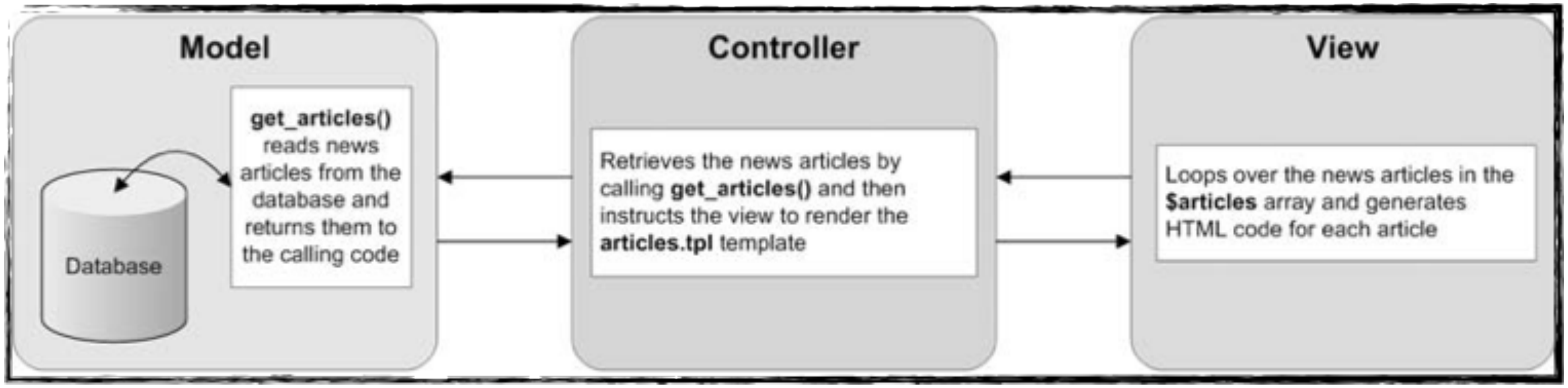
- Solution 1 - No MVC
  - Issues:
    - Hard to maintain (data access code will be scattered across the page)
    - If more than one page displays articles, all the code must be duplicated and kept consistent

MVC 6/10

- Solution 2 - Transforming to MVC
  - Model - create a class with the code that accesses the data (array). This class can be reused when required.
  - View - Loops the article list and creates the HTML page.
  - Controller - separates the code that fetches the data (model) from the code that writes the HTML (view).

MVC 7/10

- Solution 2 - MVC


Model
get_articles() reads news articles from the database and returns them to the calling code
Database
Controller
Retrieves the news articles by calling get_articles() and then instructs the view to render the articles.tpl template
View
Loops over the news articles in the $articles array and generates HTML code for each article

MVC – Model

File: “Model\Article.php”

```
<?php
namespace Model;

class Article
{
    public $title;
    public $content;

    public function __construct($title=null, $content=null)
    {
        $this->title = $title;
        $this->content = $content;
    }

    public static function all()
    {
        return [
            1 => new Article('Understanding MVC in PHP', 'This article covers the
            basics of MVC web frameworks.'),
            2 => new Article('MVC in PHP', 'The model view controller pattern is the
            most used pattern for today’s world web applications.'),
        ];
    }
}
```

MVC – Controller

PSR-2 (Coding Style Guide)

- When present, all **use** declarations MUST go after the namespace declaration.
- There MUST be one use keyword per declaration.
- There MUST be one blank line after the use block.

File: “articles.php”

```php
<?php
spl_autoload_register();

use Controllers\ArticleController;

$controller = new ArticleController;
$articles = $controller->getArticles();
$title = "List of Articles";

require('views/header.view.php');
require('views/articles.view.php');
require('views/footer.view.php');
```

File: “Controllers\ArticleController.php”

```php
<?php
namespace Controllers;

use Model\Article;

class ArticleController
{
    public function getArticles()
    {
        // EXTRA: handle filters, sort, etc...
        $articles = Article::all();
        return $articles;
    }
}
```

MVC – View

File: “views\header.view.php”

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title><?= $title ?></title>
  </head>
  <body>
```

File: “views\articles.view.php”

```
<?php if (count($articles)) { ?>
    <table>
    <thead>
        <tr>
            <th>Title</th>
            <th>Content</th>
        </tr>
    </thead>
    <tbody>
    <?php foreach ($articles as $article) { ?>
        <tr>
            <td><?= htmlspecialchars($article->title) ?></td>
            <td><?= htmlspecialchars($article->content) ?></td>
        </tr>
    <?php } ?>
    </tbody>
    </table>
<?php } else { ?>
    <h2>No articles found</h2>
<?php } ?>
```

File: “views\footer.view.php”

```
  </body>
</html>
```

PHP Frameworks

- Some PHP frameworks that support MVC, templates, ORM and other advanced features:
  - Laravel (http://www.laravel.com/)
  - Zend Framework (http://framework.zend.com/)
  - Symfony (http://www.symfony-project.org/)
  - Yii Framework (http://www.yiiframework.com/)
  - CakePHP (http://cakephp.org/)

References

- PHP and MySQL Web Development (4th Edition)
  - Luke Welling and Laura Thomson, Addison-Wesley 2009
- PHP Objects, Patterns, and Practice (2nd Edition)
  - Matt Zandstra, APress 2008
- Object Oriented PHP Concepts Techniques and Code
  - Peter Lavin, No Starch Press 2006
- PHP Documentation
  - Manual: http://www.php.net/manual/en/
- PSR – PHP Standard Recommendations
  - http://www.php-fig.org/psr/

Questions?

?