

MVCs and You!

Zend Framework 2 Model-View Controller

Modular Solutions for Rapid
Deployment of PHP-Based
Web Applications



Zend Framework 2

- ❖ Open-source web framework using PHP 5
- ❖ Licensed under MIT Open Source Initiative
- ❖ Free like speech & free like beer
- ❖ Aims to reduce time spent on development
- ❖ Thoroughly object-oriented (Mihai 20)
- ❖ Enables rapid prototype deployment
- ❖ Allows for multiple isolated apps within one project

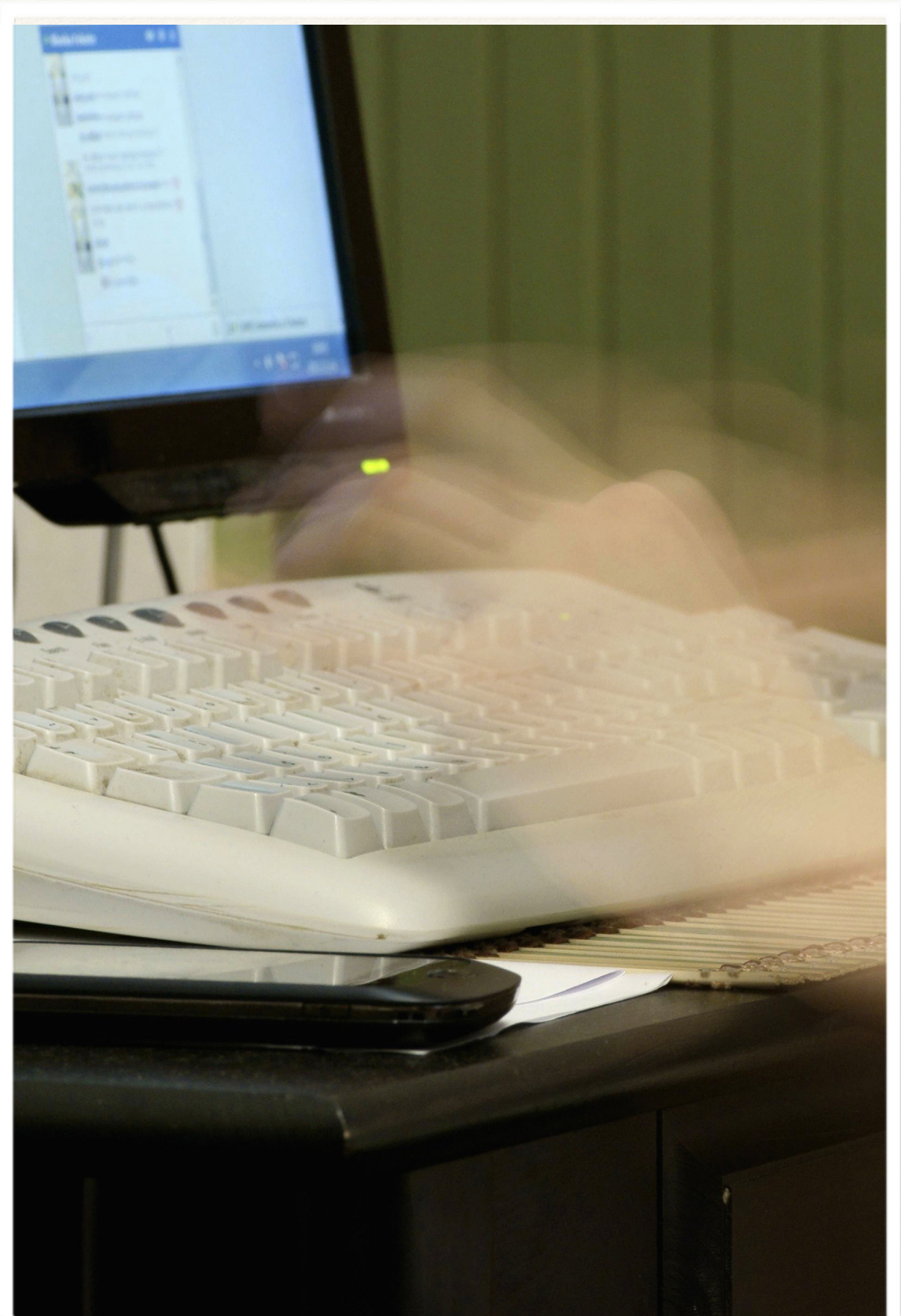
Model-View Controllers

- ❖ Model (Data) - an assembly of classes. Includes data retrieval, manipulation, and preprocessing
- ❖ View (Presentation) - The format and design of the graphic user interface; templates
- ❖ Controller (Routing) - Receives input, eg. from the view, and checks conditions to activate different states; application flow; application logic

Necessary Architecture

Dependencies

- Server Software - Apache 2.4
- Language - PHP 5
- Database - MySQL
- Testing - PHPUnit
- Dependency Management - Composer
- Version Control - BitBucket
- Cache-ing - Zend
- PHP Templates - Twig
- Object-Relational Mapping - Doctrine



Concepts

- ❖ The Model-View-Controller ("MVC") design pattern decouples the model from its view, enabling loose coupling and the ability to change one without affecting the other. (Wong)
- ❖ View Helpers are class objects built by the developers of the Zend Framework. These classes help web developers to avoid repeatedly writing the same code over and over. (Mihai-Marius 49)

Twig Template

weather.twig.php

```
1 {% do headTitle('NOC Weather') %}
2 {% do headLink().appendStylesheet(basePath('/css/Noc/weather.css')) %}
3 {% do headScript().appendFile(basePath('/js/Ext/lib/ext-all-dev.js')).appendFile(basePath('/js/Noc/Ext/weather.js')) %}
4 {% if locations|length == 0 %}
5     {% do headMeta().appendHttpEquiv('Refresh','15') %}
6     <div class="loading">Weather data is downloading. Please wait...</div>
7 {% endif %}
8
9 <table>
10     {% for location in locations %}
11         {% if loop.index == 1 %}
12             <tr>
13                 <th></th>
14                 <th>
15                     <h2>Current</h2>
16                 </th>
17                 {% for forecast in location.forecast %}
18                     <th>
19                         <h2> {{ forecast['day'] ~ ', ' ~ forecast['month'] ~ ' ' ~ forecast['date'] }}</h2>
20                     </th>
21                 {% endfor %}
22             </tr>
23             {% endif %}
24             ...
25         {% endfor %}
26     </table>
```

Alternatives

❖ MVP - Model-View Presenter

- In an MVP, the user communicates with the View which then communicates with the Presenter
- In an MVC, the user communicates with the Controller, which handles routing

❖ MVVM - Model-View ViewModel

- Like MVP, only many views can be mapped to one ViewModel.

MEAN IO

- ❖ Stands for MongoDB, Express.js, Angular.js, and Node.js
- ❖ MEAN is a fullstack JavaScript platform for modern web applications
- ❖ Uses a single language (JS) across the entire development stack, enabling rapid development as well as client-side speed gains



Works Cited

- ❖ Chauhan, Shailendra. "Understanding MVC, MVP and MVVM Design Patterns." "Understanding MVC, MVP and MVVM Design Patterns. Dot Net Tricks, 13 Mar. 2014. Web. 7 Oct. 2014. <<http://www.dotnet-tricks.com/Tutorial/designpatterns/2FMM060314-Understanding-MVC,-MVP-and-MVVM-Design-Patterns.html>>.
- ❖ Mihai-Marius, Vlăsceanu. "Zend Framework Functionality And Web Application Methodology." (2012): OAIster. Web. 6 Oct. 2014. <http://www.theseus.fi/bitstream/handle/10024/51741/zend_methodology_vlasceanu_mihai_2012.pdf?sequence=1>.
- ❖ Qureshi, M. Rizwan Jameel, and Fatima Sabir. "A Comparison Of Model View Controller And Model View Presenter." (2014): arXiv. Web. 6 Oct. 2014. <<http://arxiv.org/pdf/1408.5786v1.pdf>>
- ❖ Wong, Stephen. "Model-View-Controller Design Pattern." (2009): OAIster. Web. 6 Oct. 2014. <http://cnx.org/contents/f56fc589-7980-4450-8234-31c7501f3a8f@2/Model-View-Controller_Design_P>