

SCranfield

by Shannon Cranfield

FILE	598_SHANNON_CRANFIELD_SCRANFIELD_178145_1210842520.DOCX (205.93K)		
TIME SUBMITTED	08-APR-2019 09:47 PM (UTC+0100)	WORD COUNT	1962
SUBMISSION ID	104404155	CHARACTER COUNT	12460



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

'Part 2'

Advanced Web Development
M6X01086

Shannon Cranfield
P149078
Michael Dacey

Table of contents

1. Introduction.....	1
2. Controllers.....	1
2.1. The Front Controller	1
2.2. The Page Controller:	2
3. Web presentation patterns	2
3.1. Template Pattern	2
4. Symfony4	3
5. Conclusion	5
6. References	6
Appendix A.....	7
Appendix B	Error! Bookmark not defined.

Advanced Web Development

1. Introduction

Martin Fowler a software engineer specializing dominantly in object oriented analysis and patterns, he wrote a book called “Patterns of Enterprise Application Architecture”. In this book, he intelligibly explains that websites that are complex have similar things that need to be done when controlling a request, some of which are internationalization, security and proving certain views for specific users. If the controller input behavior is dispersed widely across manifold objects most of this behavior can result in duplication and is problematic to change behavior runtime [1]. This report will be using this as a reference to evaluate and explain the use of page and front controllers used in web applications, specifically amplifying how this approach incorporates into the Modal View Controller framework. Additionally, an exploration of web presentation patterns looking mainly at the template pattern will be explored. The report will end by looking at PHP web application framework – Symfony.

2. Controllers

Controllers are classes that can be reached through the Uniform Resource Locator and is in charge of controlling the request. Other classes within the application and models are called to fetch information by the controller [2]. The controller joins the model and the views together contingent on the resources request. A controller is what is interacting with the user, every time a request is sent to the server, the controller is what is receiving the request by processing the request, this could be a GET/PUT/DESTROY/POST request [3].

2.1. The Front Controller:

The front controller stipulates Single PHP file which is in control of handling requests. Centralizing the control of an application flow is extremely beneficial, it supports how the operation of a large complex system run. Therefore, the front controller is perfect for this [4]. A single part of a website that is responsible for handing all requests incoming to the application fully, is needed to use the front controller. A Front Controller is used mostly for larger systems and provides grander flexibility in the management of views and commands [5]. The URL in the front controller approach acts like a command line.

As changes are easily made in the front controller, it is better sorted for the front controller to be handling all the requests and apply it to all the incoming requests, frameworks such as Symfony and laravel uses a more complex front controller [6].

2.2. The Page Controller:

A page controller is an object that is in control of the requests for a specific page or action on a website. The main responsibilities of the page controllers are determining which view must be displayed and forward the information the model has. Additionally, the page controller must produce and invoke objects that the model has so that processing of the data can happen. Data that is applicable should be passed from the HTML request to the model, this is done so that objects from the model doesn't need any connection to the html request [5].

Pages on website holds their own document on the server. Dynamic pages have a more intricate association between path names and the file that responds. For each logical page of the website the page controller has one input, the controller could control a page or an object that is linked to that page. Each page within the web application acts as a controller for an individual view or many. Each action gives a different view but are all related to the user .



3. Web presentation patterns

In a lifetime, patterns have a huge role in one's life for example breakfast, work, bedtime and then repeating them. Just like patterns play a huge role in a human's life, patterns play an important part in web development. Design patterns are solutions to problems that commonly occur in software and web design [7].

3.1. Template Pattern

The template pattern's main purpose is to avoid code duplication by creating abstract classes, this classes will need to extend. Inside these abstract classes, organisation of the code happens instead of duplication. Code from the sub-classes will inherit from the abstract class which allow methods to be reused of the parent class. This ensures that

sub classes doesn't need to duplicate the same code. The template pattern is used mainly to reduce copy and paste [8].

The template method is evidence that inheritance is a prevailing methodology when it comes to reusing blocks of code/logic. The template pattern declares methods in the abstract class, these methods either are actual template methods that call methods in a specific order or that act as steps of an algorithm.

In appendix A, an example of the template pattern is shown. The main abstract class has the main methods that each sub-class will inherit. The sub classes which extends the Sandwich abstract class, which allows the addition of its own implementation for the addfillings(), in this case adding cheese for the cheese sandwich and peanut butter for the peanut butter sandwich, however both these sandwiches can do a few methods that a the same.



4. Symfony

Symfony is a set of PHP decoupled and reusable components, a web application framework, a philosophy and a community that work conjointly together [9]. Symfony's philosophy is embracing standards and practices but predominantly interoperability among applications meaning components in Symfony may work with other frameworks. The Symfony community is one of the largest PHP communities with over 600,000 developers around 120 countries, this shows that PHP can be used in large enterprises [10]. Symfony can aid with overcoming tasks such as routing, templating and much more. There are many PHP frameworks that aid in the making of web applications, this section of the report will explore some of the reasons for using and for not using Symfony.

Symfony Imposes best practices by driving users to work in a certain way using folder structures, putting specific files into folders, having a good folder structure makes the code more organized making team collaboration easier also making the code easily maintainable. Therefore, users don't have to worry about tools being created but just concentrating on the application creation [11].

Software developers prefer to decide which framework to use according to its flexibility and easy to use format. Symfony is accessible for novice and export developers. Due to the fact that Symfony has a great functional environment the process and experience of web development is great [12]. The user-friendly interface along with effective methodology succors in the creating of dynamic and complex applications that have great flexibility. MCV components is integrated giving developers control over

configuration and other features of development. Additionally, Symfony is compatible with other databases. Symfony has a large range of bundles also known as plugins. Bundles (plugins) is modular blocks of code which have their own functionality. The purpose of bundles is to heighten the frameworks functionality and features. These bundles can be used in project other than the ones in Symfony [12].

Symfony makes use of a great templating system. Views are stored in the templates to show to user. These views can be PHP or Twig. Twig used mostly by developers is a templating system which is used to write concise codes. It is the securest, reliable and quickest PHP template engine that Symfony uses.

A third-party library is used called Doctrine as Symfony does not provide a component to work with databases. Doctrine is an object relational mapper, ORM is a method used in programming languages when dealing with databases for translating incompatible data types in relational databases. Doctrine which is placed above the database abstraction layer. The main advantage or one of its key features is that it allows for database to be written in Doctrine Query language an alternative to SQL that keeps flexibility without requiring needless code repetition [13]. Furthermore, Doctrine will treat the databases as if they were object oriented.

In Symfony testing is made easy by the privatization of tools for behavioral, functional and unit testing. Unit independent library is offered, this fuels HTTP requests and authenticates the output without any script using testing tools [14]. Due to the use of PHP Unit Independent library testing is made easy. Additionally, functional testing is self-operated to inflate a developer's routine [15]. Maintainability and testing is made better by reusing bundles and the creating design patterns. There are tools that assist with fixing errors within the code and issues with security. These tools support improving productivity and ensuring that development is fast and essentially bug-free.

Symfony allows for dependency injection and has a component which is centralized to allow ways in which objects are constructed. Class and parameters are added to create a service. A latest update of Symfony allows for services to be automatically loaded which means developers don't have to define services individually but do however define classes from folders as a service [16].

Symfony may be used for any applications, it is not designed for small scaled applications since symphony allows customization of options [17].

Although, Symfony has great documentation and is user friendly for both novice and expert developers, it has a very steep learning curve at the start. Time is needed for developers to learn and get familiarized with the framework.

Overall, Symfony is lovely framework which makes development of applications quick and easy and is recommended highly using the framework.

5. Conclusion

This report explored the use of page and front controllers within a web applications. A web presentation pattern was identified. The main purpose of the template pattern is to provide a common place where code recites and all the implementation has this same code, so repetition dos not occur.

6. References

- [1] M. Fowler, *Patterns of enterprise application architecture (Паттерны корпоративных приложений)*. 2003.
- [2] “Controller - General - FuelPHP Documentation.” [Online]. Available: <https://fuelphp.com/docs/general/controllers/base.html>. [Accessed: 20-Mar-2019].
- [3] DevMarketer, *What is programming MVC?* 2016.
- [4] J. E. Sweat, *PHP Architect's Guide to PHP Design Patterns*. 2005.
- [5] M. Dacey, “PHP Frameworks.” 2012.
- [6] “The Front Controller Design Pattern.” [Online]. Available: <https://webdevetc.com/blog/the-front-controller-design-pattern-in-php>. [Accessed: 08-Apr-2019].
- [7] Snook Jonathan, “Web Presentation Patterns,” Jan-2016.
- [8] “The template pattern.” [Online]. Available: <https://phpenthusiast.com/blog/the-template-pattern-the-power-of-abstraction>. [Accessed: 03-Apr-2019].
- [9] “What is Symfony.” [Online]. Available: <https://symfony.com/what-is-symfony>. [Accessed: 24-Mar-2019].
- [10] Bernardo Pineda, *What is Symfony?* .
- [11] Zenva, *What is Symfony, Symfony, Twig and Doctrine Beginners Tutorial - YouTube* . .
- [12] “10 Advantages Of The Symfony PHP Framework | Software Developer India.” [Online]. Available: <https://www.software-developer-india.com/10-advantages-of-the-symfony-php-framework/>. [Accessed: 03-Apr-2019].
- [13] “Doctrine ORM for PHP Guide to Doctrine for PHP Doctrine 1.1 License: Creative Commons Attribution-Share Alike 3.0 Unported License Version: manual-1,” 2009.
- [14] “The 9 Incredible Reasons to Use Symfony Framework.” [Online]. Available: <https://www.techcronus.com/blog/reasons-to-use-symfony-framework-with-features-advantages/>. [Accessed: 24-Mar-2019].
- [15] “7 Good Reasons to Use Symfony Framework for Your Project.” [Online]. Available: <https://hackernoon.com/7-good-reasons-to-use-symfony-framework-for-your-project-265f96dcf759>. [Accessed: 24-Mar-2019].
- [16] “Symfony 4 and Php 7 - WayfareWayfare.” [Online]. Available: <https://wayfare.ro/symfony-4-php-7/>. [Accessed: 03-Apr-2019].
- [17] Mounika, “Know about Laravel and Symfony frameworks of PHP.” [Online]. Available: <https://krify.co/tag/disadvantages-of-symfony/>. [Accessed: 24-Mar-2019].

Appendix A

```
<?php
```

```
namespace TemplatePattern;
```

```
abstract class Sandwich {
```

```
    protected function makeSandwich()
```

```
    {
```

```
        return $this ->placeBread()
```

```
            ->spreadButter()
```

```
            ->addFillings()
```

```
            ->closeBread()
```

```
            ->eat();
```

```
    }
```

```
    protected function placeBread()
```

```
    {
```

```
        echo('place bread');
```

```
        return $this;
```

```
    }
```

```
    protected function spreadButter()
```

```
    {
```

```
        echo('spread butter');
```

```
        return $this;
```

```
    }
```

```
    protected function addFillings()
```

```
    {
```

```

        echo('close bread ');

    return $this;
}

protected function eat(){
    echo('EAT');

    return $this;
}
}

class PeanutBuuterSandwhich extends Sandwhich {
protected function addFillings()
{
    echo ('add peanut butter');

    return $this;
}
}

class CheeseSandwhich extends Burger{
protected function addFillings()
{
    echo('add cheese');

    return $this;
}
}

?>

```

FINAL GRADE

50/100

GENERAL COMMENTS

Instructor

Investigation of Fowler (30) 22

Investigation of Symfony (20) 16 Quality (20) 2

View handling (10) 4

Database handling (15) 6

Software testing (5) 0

I would like to have seen more evaluation of benefits and drawbacks of Fowlers patterns. Also, you should have discussed the Template View approach.

Good discussion of Symfony, again I felt that more evaluation should have been provided. Also, some basic examples to illustrate some of the features would have been useful.

Code is using basic PHP, there is small piece of isolated code that using MVC.

PAGE 1

PAGE 2

PAGE 3

PAGE 4



Comment 1

What are the drawbacks of this approach, if any?



Comment 2

This is not associated with Fowler, you should have been considering the Template View approach.

I will give you some marks for as this could be a useful pattern in web development.



Comment 3

expert



Comment 4

You need to justify this