



Веб разработка на PHP Symfony

Twig



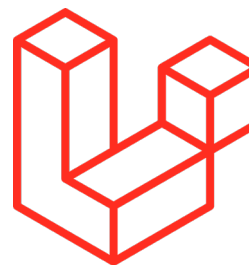
Веб разработка на PHP Symfony

Introduction



Демьян Костельный
Middle PHP Developer

 demian-kostelny-613b90151



Веб разработка на PHP Symfony

Тема урока

Twig

Веб разработка на PHP Symfony

План урока

1. Знакомство с Twig
2. Настройка Twig
3. Генерация шаблона, переменные
4. Глобальные переменные в Twig
5. Оформление страницы ошибок с помощью Twig
6. Twig filters and extensions

Веб разработка на PHP Symfony

Знакомство с Twig

Веб разработка на PHP Symfony

Знакомство с Twig

Twig представляет из себя шаблонизатор для PHP. Symfony использует внутри себя Twig для построения шаблонов. Данный шаблонизатор уже по умолчанию установлен в новом приложении Symfony и готов сразу же к использованию. В предыдущих уроках мы уже видели то какая структура у файлов этого шаблонизатора.

Веб разработка на PHP Symfony

Знакомство с Twig

Шаблонизатор позволяет использовать php функции прямо внутри шаблонов, а также кроме этого подгружать другие шаблоны прямо внутрь шаблона. Пример:

```
<h1>{{ some_value }}</h1>
<h2>Пример использования математических операторов: {{ 2 + 2 / 2 * 2 }}</h2>
<h2>Использование логической конструкции</h2>
{% if ( my_val == true ) %}
    <p>Этот текст выведется если переменная my_val будет иметь значение true</p>
{% endif %}
```

В примере были показаны не все возможности шаблонизатора с которыми мы ознакомимся дальнейшем.

Настройки Twig

Веб разработка на PHP Symfony

Настройки Twig

Шаблонизатор Twig можно полностью настраивать в своём приложении, посмотреть файл конфигурации можно в папке `config/packages/twig.yaml`. Но кроме этого посмотреть настройки Twig можно с помощью следующих команд в командной строке.

Следующая команда показывает конфигурацию по умолчанию:

```
php bin/console config:dump-reference twig
```

Следующая же команда показывает актуальный конфиг в нашем приложении:

```
php bin/console debug:config twig
```

Веб разработка на PHP Symfony

Настройки Twig

Заглядывая в документацию мы можем увидеть множество параметров, которые мы можем настроить в файле конфигурации Twig. Они могут отвечать за самые разные части работы в Twig. Например за кодирование символов - или формат даты.

Configuration ¶

- `auto_reload`
- `autoescape`
- `autoescape_service`
- `autoescape_service_method`
- `base_template_class`
- `cache`
- `charset`
- `date`
 - `format`
 - `interval_format`
 - `timezone`
- `debug`
- `default_path`
- `form_themes`
- `globals`
- `number_format`
 - `decimals`
 - `decimal_point`
 - `thousands_separator`
- `optimizations`
- `paths`
- `strict_variables`

Веб разработка на PHP Symfony

Настройки Twig

Если нам нужно изменить какие-то настройки в Twig, мы просто берём интересующий нас параметр, добавляем его в файл конфигурации и конечно же задаём ему значение.

Например так можно отключить кэширование в Twig:

```
twig:  
    default_path: '%kernel.project_dir%/templates'  
    cache: true
```

Веб разработка на PHP Symfony

Генерация шаблона, переменные

Веб разработка на PHP Symfony

Генерация шаблонов, переменные

Как нам уже известно, всегда при генерации новых контроллеров - к контроллеру также создаётся папка с шаблоном, который имеет такое же название, как и контроллер. Кроме этого можно абсолютно спокойно создавать новые файлы в любом удобном месте и использовать их либо в самих шаблонах, либо в контроллерах.

```
{% extends 'base.html.twig' %}

{% block title %}Hello BlogController!{% endblock %}

{% block body %}
<style>
  .example-wrapper { margin: 1em auto; max-width: 800px; width: 95%; font: 18px/1.5 sans-serif; }
  .example-wrapper code { background: #F5F5F5; padding: 2px 6px; }
</style>

<div class="example-wrapper">
  <h1>Hello {{ controller_name }}! ☒ </h1>

  This friendly message is coming from:
  <ul>
    <li>Your controller at <code><a href="{{ 'C:/Dev/my_project_name/src/Controller/BlogController.php'|file_link(0) }}">src/Controller/
    BlogController.php</a></code></li>
    <li>Your template at <code><a href="{{ 'C:/Dev/my_project_name/templates/blog/index.html.twig'|file_link(0) }}">templates/blog/
    index.html.twig</a></code></li>
  </ul>
</div>
{% endblock %}
```

Веб разработка на PHP Symfony

Генерация шаблонов, переменные

В уроке о контроллерах было уже показано как передаются переменные в шаблоны. Переменные могут иметь любые типы данных и быть переданы в шаблон и там уже выведены либо обработаны с помощью PHP функций, если нужно.

```
/**
 * @Route("/blog", name="blog")
 */
public function index(): Response
{
    return $this->render('blog/index.html.twig', [
        'controller_name' => 'BlogController',
    ]);
}
```

```
<div class="example-wrapper">
    <h1>Hello {{ controller_name }}! ☒ </h1>
```

Веб разработка на PHP Symfony

Генерация шаблонов, переменные

Сейчас мы рассмотрим больше основного функционала Twig, к примеру, как подключать CSS/JS файлы, как подгружать другие шаблоны в шаблон и т.п.

Глобальные переменные в Twig

Веб разработка на PHP Symfony

Глобальные переменные в Twig

В Twig можно настроить глобальные переменные, которые имеют постоянное значение и их можно использовать в любом шаблоне. Чтобы создать новую переменную нужно открыть файл конфигурации Twig `config/packages/twig.yaml` и добавить параметр массив `globals`:

```
twig:
  default_path: '%kernel.project_dir%/templates'
  # ...

  globals:
    tracking_id: 'Some id' # Данные переменные могут иметь самые разные типы данных
    my_global_value: 123  # и кроме этого их можно будет вызвать в любом шаблоне.
                          # Глобальные переменные хорошо использовать например в
                          # в трекарах аналитики.
```

Оформление страницы ошибок с помощью Twig

Веб разработка на PHP Symfony

Оформление страниц ошибок с помощью Twig

Часто нам нужно полностью изменить дизайн страниц ошибок, таких как, например, 404 или 500. Для того чтобы это сделать - нужно установить twig-pack в приложение через Composer:

```
composer require symfony/twig-pack
```

После этого, нам нужно создать файлы шаблонов ошибок. Давайте сделаем это вместе.

Twig filters and extensions

Веб разработка на PHP Symfony

Twig filters and extensions

Фильтры в Twig представляют из себя стандартные PHP функции, которые можно использовать для того чтобы работать с данными в Twig. Например, есть фильтр `upper`, который делает строку `uppercase`. Пример использования фильтра:

```
{{ 'hello world!' |upper }} <!-- Фильтр надо указывать после символа "|"
                                     и результатом использования фильтра будет HELLO WORLD!
-->
```

Веб разработка на PHP Symfony

Twig filters and extensions

Список всех фильтров можно найти в официальной документации по ссылке:
<https://twig.symfony.com/doc/3.x/filters/index.html>

Filters ¶

- [abs](#)
- [batch](#)
- [capitalize](#)
- [column](#)
- [convert_encoding](#)
- [country_name](#)
- [currency_name](#)

Веб разработка на PHP Symfony

Twig filters and extensions

Если нам нужно написать какой-то собственный фильтр, то можно воспользоваться `extensions` (расширениями) для Twig. Чтобы создать своё расширение нужно создать файл по адресу `src/Twig/AppExtension.php` и уже внутри файла писать код собственного расширения. Рассмотрим небольшой и простой пример.

Веб разработка на PHP Symfony

Twig filters and extensions

Сначала нам нужно как-то назвать наше расширение. Пусть оно будет называться `custom_name` и будет оно сначала зарегистрировано через функцию `getFilters()`, в которой в `return` указываются собственные фильтры со ссылками на PHP функции фильтров, которые и будут выполнять заданные действия фильтра.

```
public function getFilters()
{
    return [
        new TwigFilter('custom_name', [$this, 'customName']), // Здесь указываем название PHP функции которая будет создана
                                                                // и будет выполнять действия для фильтра который мы назвали
                                                                // custom_name
        // в этом классе
    ];
}
```


Веб разработка на PHP Symfony

Twig filters and extensions

После регистрации нашего расширения, настало время написать функцию, которая и будет выполнять действия для нашего фильтра. Наша функция `customName` принимает внутри себя строку `$name`, которую мы будем обрабатывать и добавлять текст “+ this string is customized”.

```
public function customName($name)
{
    $name = $name . ' + this string is customized';

    return $name;
}
```

Веб разработка на PHP Symfony

Twig filters and extensions

В конечном итоге мы сможем использовать наш фильтр теперь в шаблонах приложения:

```
{{ 'test' |custom_name('test string') }}
```

Информационный видеосервис для разработчиков программного обеспечения



Проверка знаний

TestProvider.com



Проверьте как Вы усвоили данный материал на TestProvider.com

TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.

Веб разработка на PHP Symfony

Спасибо за внимание! До новых встреч!



Демьян Костельный
Middle PHP Developer

