

Что такое шаблонизатор Twig и зачем он нужен?

На весь экран

[Главная](#) > [Блог](#) > [Programming](#) > Что такое шаблонизатор Twig и зачем он нужен?



Новченкова Варвара
программист агентства
RACURS



9196



03.06.2021

Содержание статьи

- 1. [Зачем нужны шаблонизаторы?](#)
- 2. [История создания](#)
- 3. [Особенности Twig](#)
- 4. [Подключение Twig и рендер первого шаблона](#)
- 5. [Оформление кода](#)
- 6. [Циклы и условия в Twig](#)
- 7. [Фильтры в Twig](#)
- 8. [Заключение](#)

Оглавление

Зачем нужны шаблонизаторы?

Создание качественного продукта, будь то сайт или приложение – результат сплоченной работы целой команды. Это программисты, верстальщики, дизайнеры. Чем сложнее проект, тем важнее отделить его логику от представления. Только при успешном их разделении backend и frontend-разработчики смогут эффективно сотрудничать. Всего этого можно достигнуть благодаря шаблонизаторам.

На данный момент существует множество способов генерации конечных html-страниц. В том числе большое количество шаблонизаторов. Вот некоторые из них: Blade, Smarty, Twig, Volt. Мы остановимся подробнее на Twig.

История создания

Twig был написан в 2008 году австрийским разработчиком Армином Ронахером, как обработчик шаблонов с открытым исходным кодом, созданный на языке программирования php. Вскоре он занялся разработкой на Python и больше не возвращался к шаблонизатору. Однако труды были не напрасны. Идею Twig поддержал и развил ведущий разработчик фреймворка Symfony – Фабьен Потенсье.

Заказать бесплатный SEO аудит

Оставить заявку



своим личным опытом уверен отметил. «Twig не только один из самых функциональных шаблонизаторов, но и самый быстрый».

Особенности Twig

К основным особенностям шаблонизатора относятся:

- высокая скорость работы
- наследование шаблонов (возможность определить базовый и дочерние шаблоны)
- понятный и компактный синтаксис
- оптимизированный php код
- безопасный режим «песочницы» (пользователи имеют ограниченный набор тегов, фильтров и методов объектов, определенных разработчиком)
- возможность создания и использования макросов
- полностью документирован (и API тоже)
- простая отладка (при синтаксических ошибках шаблонизатор выведет сообщение об имени файла с ошибкой и номером строки)

На весь экран



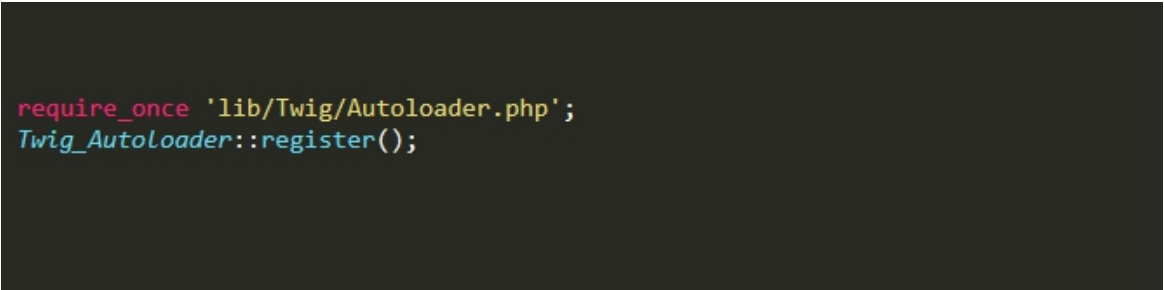
Подключение Twig и рендер первого шаблона

Один из самых простых способов установки – скачать компонент с GitHub. Далее следует распаковать архив и каталог lib поместить в папку с рабочим проектом.



Всё необходимое для того, чтобы подключить Twig находится в файле index.php. В каталоге templates размещены файлы шаблонов. В нашем случае base.html и books.html. compilation_cache используется для хранения php-файлов, созданных шаблонизатором для работы.

1. Подключаем Twig.



2. Сообщаем шаблонизатору местонахождение шаблонов. В качестве аргумента передаем путь к папке templates.

[Заказать бесплатный SEO аудит](#)[Оставить заявку](#)

3. Создаем объект шаблонизатора, указав необходимые опции **cache** и **auto_reload**.

```
$twig = new Twig_Environment($loader, array(
    'cache' => 'compilation_cache',
    'auto_reload' => true
));
```

[На весь экран](#)

Параметру **cache** передается путь к папке **compilation_cache**.

Вторую опцию активируем, прописав **true**. Это необходимо при обновлении страницы, чтобы Twig заново обработал шаблон. В противном случае шаблонизатор закеширует страницы. В будущем, после завершения работы над сайтом, данную опцию можно отключить.

4. Для наглядной работы с шаблонизатором сформируем небольшой массив со списком книг вида ключ-значение.

```
$books = array(
    array('number' => 'Книга 1', 'title' => 'Этюд в багровых тонах', 'date' => '1887'),
    array('number' => 'Книга 2', 'title' => 'Тайна Боскомской долины', 'date' => '1891'),
    array('number' => 'Книга 3', 'title' => 'Случай с переводчиком', 'date' => '1893'),
    array('number' => 'Книга 4', 'title' => 'Пенсне в золотой оправе', 'date' => '1904'),
);
```

5. Используем метод **render()** для передачи данных в шаблон и вывода html результата.

```
echo $twig->render('books.html', array('books' => $books));
```

Аргументы: название файла шаблона и созданный ранее массив.

Общий вид файла:

```
$loader = new Twig_Loader_Filesystem('templates');

$twig = new Twig_Environment($loader, array(
    'cache' => 'compilation_cache',
    'auto_reload' => true
));

$books = array(
    array('number' => 'Книга 1', 'title' => 'Этюд в багровых тонах', 'date' => '1887'),
    array('number' => 'Книга 2', 'title' => 'Тайна Боскомской долины', 'date' => '1891'),
    array('number' => 'Книга 3', 'title' => 'Случай с переводчиком', 'date' => '1893'),
    array('number' => 'Книга 4', 'title' => 'Пенсне в золотой оправе', 'date' => '1904'),
);

echo $twig->render('books.html', array('books' => $books));
```

6. Настройка **base.html** (базовый шаблон)

Наследование шаблонов позволяет вынести общие элементы сайта, такие как **header**, **footer**, **sidebar** в базовый шаблон. Он определяет костяк html-документа. Каждый блок, в свою очередь, может быть переопределен дочерним шаблоном.

```
<head>
  <title>{% block title %}{% endblock %}</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
</head>
<body>

<div id="content">
  {% block content %}
  {% endblock %}
</div>

</body>
</html>
```

В нашем примере два блока: title и content.

7. Настройка books.html (дочерний шаблон)

На весь экран

а) указываем родительский шаблон

```
{% extends "base.html" %}
```

б) вписываем название дочерней страницы в наш блок title. Оно появится между тегами title в base.html

```
{% block title %}Серия книг о Шерлоке Холмсе{% endblock %}
```

в) заполняем содержание, оно будет помещено в блок content базового шаблона

```
{% block content %}
<h1>Серия романов о Шерлоке Холмсе</h1>

<div id="books">
  {% for book in books %}
  <div class="book">
    <strong>{{book.number}}</strong>. "<em>{{book.title}}</em>" - {{book.date}}
  </div>
  {% endfor %}
</div>
{% endblock %}
```

Цикл **for** проходит по данным, которые были переданы в шаблон посредством метода **render()**. Для извлечения значений необходимо обратиться к элементам по их ключам. В нашем случае это: `{{ book.number }}` и `{{ book.title }}`.

Выстраивается следующая цепочка:

Срабатывает метод `$twig->render('books.html', array('books' => $books))`



Twig подгружает шаблон books.html и формирует html-содержание



Распознает, что books.html это дочерний шаблон base.html и помещает туда сформированное содержание



Шаблонизатор возвращает полноценную страницу html

Оформление кода

Правило 1. Только один пробел после начала разделителя (`{{`, `{%`, и `{#`) и перед концом разделителя (`}}`, `%}` и `#}`):

```
{% if foo %}{% endif %}
```

Правило 2. Не ставить пробелов:

- после открытия круглой скобки и перед закрытием круглой скобки в выражениях
- перед и после разделителей строк
- перед и после следующих операторов: `|`, `..`, `..`, `[]`
- перед и после круглой скобки, используемой для филт функций

На весь экран
- перед и после открытия и закрытия массивов и хэшей

```
{{ 1 + (2 * 3) }}
```

```
{{ 'foo' }}
{{ "foo" }}
```

```
{{ foo|upper|lower }}
{{ user.name }}
{{ user[name] }}
{% for i in 1..12 %}{% endfor %}
```

```
{{ foo|default('foo') }}
{{ range(1..10) }}
```

Правило 3. В названиях переменных использовать маленькие буквы и символы подчеркивания.

Правило 4. Делать отступы внутри тегов.

Циклы и условия в Twig

В Twig есть такие базовые конструкции как циклы и условия. В примере с выводом книг был задействован цикл `for`:

```
{% block content %}
<h1>Серия романов о Шерлоке Холмсе</h1>

<div id="books">
  {% for book in books %}
    <div class="book">
      <strong>{{book.number}}</strong>. "<em>{{book.title}}</em>" - {{book.date}}
    </div>
  {% endfor %}
</div>
{% endblock %}
```

Одной из особенностей шаблонизатора является использование **оператора двоеточие** (`..`). Это альтернатива `range` в php для создания массива, содержащего диапазон элементов.

Пример перебора цифр:

```
{% for number in 0..100 %}
  {{ number }}
{% endfor %}
```

Кроме того, полезной является возможность добавить условие в определение цикла.

Пример – фильтрация продуктов по цене:

```
<tbody>
  {% for product in products if product.value < 150 %}
  <tr>
    <td>{{ product.name }}</td>
    <td>{{ product.description }}</td>
    <td>{{ product.value }}</td>
    <td>{{ product.date_register|date("m/d/Y") }}</td>
  </tr>
  {% endfor %}
</tbody>
```

На весь экран

Условия, поддерживаемые Twig: **if**, **elseif**, **if not** и **else**:

```
<tbody>
  {% for product in products %}
  {% if product.value > 300 %}
  <tr>
    <td>{{ product.name }}</td>
    <td>{{ product.description }}</td>
    <td>{{ product.value }}</td>
    <td>{{ product.date_register|date("m/d/Y") }}</td>
  </tr>
  {% endif %}
  {% endfor %}
</tbody>
```

Фильтры в Twig

Фильтры – более продвинутые средства Twig. Эти инструменты позволяют форматировать данные перед выводом, определять в каком формате и как будет передана информация в шаблон. Вот несколько из них:

Date и **date_modify** – форматируют и модифицируют дату

```
<td>{{ product.date_register|date_modify("+3 day")|date("m/d/Y") }}</td>
```

Format – преобразует placeholder’ы, строки, которые начинаются со знака %, заменяя на подставляемые значения.

```
<td>{{ "Описание товара: %s" | format(product.description) }}</td>
```

Striptags – удаляет html теги, все пробелы, находящиеся рядом, заменяет одним.

```
{{ <alert>alert("Contrary to popular belief, Lorem Ipsum is not simply random text.")</alert> | striptags }}
```

Escape – экранирует строку для безопасного отображения.

Заключение

Twig – это мощное средство для создания шаблонов сайта, которое хорошо документировано и полностью протестировано. Он особенно полезен для командного ведения проекта, позволяя комфортно работать программистам и

Заказать бесплатный SEO аудит

Оставить заявку



Написать комментарий:

Имя

e-mail

Вам понравилась статья ?

- ★
- ★
- ★
- ★
- ★

Ваш отзыв

На весь экран

Отправить

Рекомендуемые статьи

- [Дубли страниц – почему это плохо и как с ними бороться](#)
- [SERM или как перестать беспокоиться и начать жить бизнесу в интернет](#)
- [Особенности продвижения сайтов B2B-сегмента](#)

Написать нам

[Документы](#)

Оглавление