Developer & Code

9. Twig функции для ссылок. Блог на PHP

Червень 18, 2021

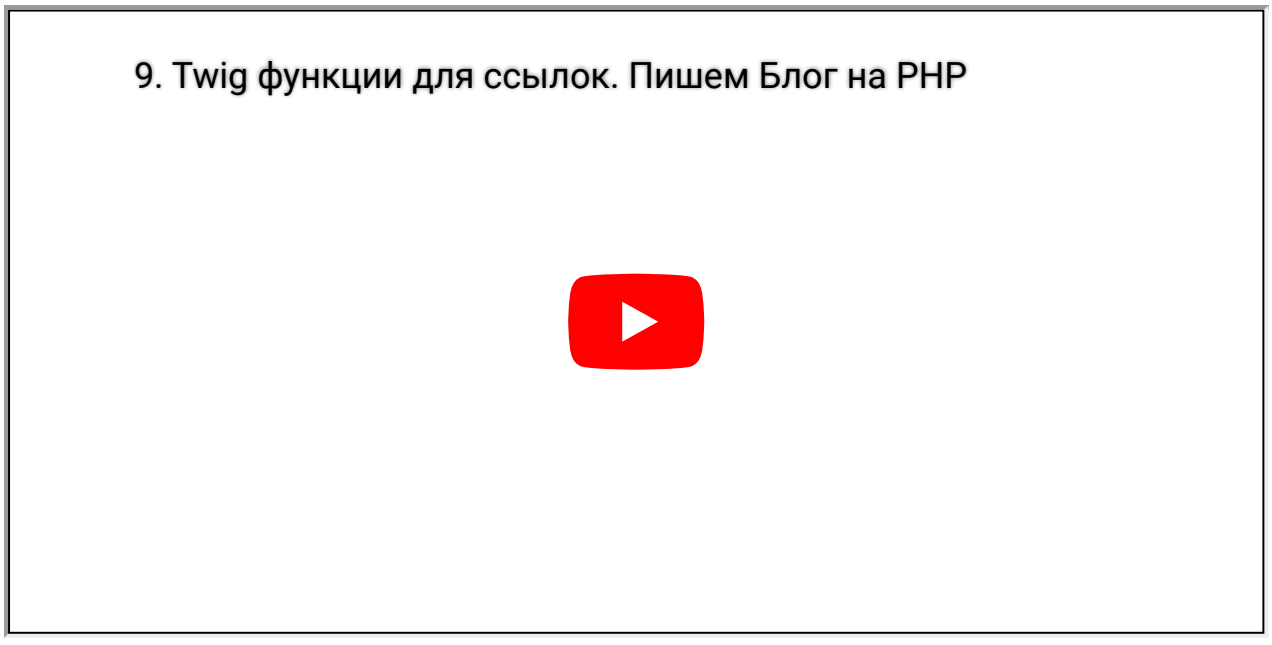
В этом уроке, мы создадим 3 функции, которые будем использовать в шаблонах Twig. Эти функции помогут нам решить вопрос с картинками.

Какие темы мы разберем

В 9. Twig функции для ссылок – Блог на PHP, уроке мы рассмотрим следующие темы:

- Добавим 3 twig функции для работы со ссылками
- Обновим существующие шаблоны с использованием новых twig функций

Приятного просмотра:



В рамках урока, мы напишем следующий код.

В `index.php` файле, давайте добавим обработчик `TwigMiddleware`.

```php
<?php

use Blog\Slim\TwigMiddleware;

//инициализация...

$app->add(new TwigMiddleware($view));
```

Реализация класса `TwigMiddleware`.

```php
<?php

declare(strict_types=1);

namespace Blog\Slim;

use Blog\Twig\AssetExtension;
use Psr\Http\Message\ResponseInterface;
use Psr\Http\Message\ServerRequestInterface;
use Psr\Http\Server\MiddlewareInterface;
use Psr\Http\Server\RequestHandlerInterface;
use Twig\Environment;

class TwigMiddleware implements MiddlewareInterface
{
    /**
     * @var Environment
     */
    private Environment $environment;

    /**
     * TwigMiddleware constructor.
     * @param Environment $environment
     */
    public function __construct(Environment $environment)
    {
```

```
        $this->environment = $environment;
    }

    /**
     * @param ServerRequestInterface $request
     * @param RequestHandlerInterface $handler
     * @return ResponseInterface
     */
    public function process(
        ServerRequestInterface $request,
        RequestHandlerInterface $handler
    ): ResponseInterface {
        $this->environment
            ->addExtension(new AssetExtension($request));
        return $handler->handle($request);
    }
}
```

Также, давайте создадим класс `AssetExtension`, с помощью которого, мы добавим 3 Twig функции: `getUrl()`, `getAssetUrl()`, и `getBaseUrl()`.

```
<?php

declare(strict_types=1);

namespace Blog\Twig;

use Psr\Http\Message\ServerRequestInterface;
use Twig\Extension\AbstractExtension;
use Twig\TwigFunction;

class AssetExtension extends AbstractExtension
{
    /**
     * @var ServerRequestInterface
     */
    private ServerRequestInterface $request;

    /**
     * AssetExtension constructor.
     * @param ServerRequestInterface $request
```

```
     */
    public function __construct(ServerRequestInterface $request)
    {
        $this->request = $request;
    }

    /**
     * @return array|TwigFunction[]
     */
    public function getFunctions()
    {
        return [
            new TwigFunction('asset_url', [$this, 'getAssetUrl']),
            new TwigFunction('url', [$this, 'getUrl']),
            new TwigFunction('base_url', [$this, 'getBaseUrl']),
        ];
    }

    /**
     * @param string $path
     * @return string
     */
    public function getAssetUrl(string $path): string
    {
        return $this->getBaseUrl() . $path;
    }

    /**
     * @return string
     */
    public function getBaseUrl(): string
    {
        $params = $this->request->getServerParams();
        return $params['REQUEST_SCHEME'] . '://' . $params['HTTP_HOST'
    }

    /**
     * @param string $path
     * @return string
     */
    public function getUrl(string $path): string
    {
        return $this->getBaseUrl() . $path;
```

```
    }
}
```

И мы можем использовать новые функции в twig шаблонах.

Файл `index.twig`, давайте обновим ссылку на Блог используя функцию `url()`, которую мы объявили и реализовали в классе `AssetExtension`.

```
<a href="{{ url('blog') }}">View all</a>
```

В файле `templates/post.twig`, с помощью функции `asset_url()`, починим ссылку на картинку поста.

```
<img src="{{ asset_url(post.image_path) }}" class="img-fluid" alt="{{
```

В файле `section/navigation.twig` вместо ”/” атрибута `href`:

```
<a class="navbar-brand" href="/">Blog</a>
```

добавим функцию `base_url()`:

```
<a class="navbar-brand" href="{{ base_url() }}">Blog</a>
```

И правильные ссылки в навигации с помощью функции `url()`:

```
        <ul class="navbar-nav">
          <li class="nav-item">
            <a class="nav-link" href="{{ url('blog') }}">Blog</a>
          </li>
          <li class="nav-item">
```

```
            <a class="nav-link" href="{{ url('about') }}">About</a>
          </li>
        </ul>
```

Для файла `section/posts.twig`, давайте также используем функцию `url()`:

```
<div class="row mb-2 bg-light" style="border: 1px solid #ddd;">
  <div class="p-3">
    {% if post.image_path is not null %}
      <img src="{{ asset_url(post.image_path) }}" class="card-img-to
    {% endif %}
    <div class="mt-3">
      <a href="{{ url(post.url_key) }}"><h5 class="card-title">{{ po
      <span class="text-muted">Max Pronko | {{ post.published_date|d
      <p class="mt-2">{{ post.description|escape }}</p>
      <a href="{{ url(post.url_key) }}" class="">Read more</a>
    </div>
  </div>
</div>
```

Скачать код данного урока можно по ссылке.

Підписуйтеся на канал “Спільнота програмістів – Developer & Code” в телеграмі

Макс Пронько – Програміст, CEO компанії Pronko Consulting, розбираємо Веб технології. Автор YouTube каналу Макс Пронько. Телеграм група сайту.

← 8. Страница Постов и пагинация. Блог на PHP

10. Пагинация постов. Блог на PHP →