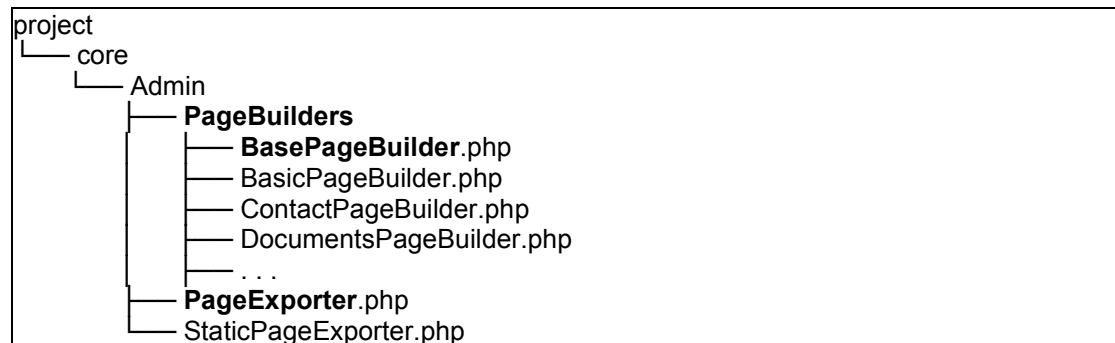


Svaki custom **PageBuilder** za određenu stranicu u project/core/Admin/PageBuilders direktorijumu nasledjuje osnovnu klasu BasePageBuilder, što omogućava da sve stranice dele zajedničku logiku za generisanje HTML sadržaja, uključivanje CSS/JS resursa i pripremu podataka iz modela



Da bi se kreirao novi **PageBuilder**, dovoljno je napraviti novu **PHP** klasu, npr. **NazivStraniceBuilder.php**, koja nasleđuje BasePageBuilder, definisati CSS i HTML specifičan za tu stranicu, kao i dodatni PHP kod za dohvat podataka iz odgovarajućeg modela (App\Models\NazivModela). Nakon toga, implementira se metoda buildPage() koja kombinuje sve delove i vraća kompletan HTML sadržaj stranice, spreman za prikaz. Na ovaj način svaka stranica ostaje modularna i lako proširiva, dok BasePageBuilder osigurava konzistentnu strukturu i zajedničke funkcionalnosti.

```

<?php
namespace App\Admin\PageBuilders;
use App\Models\NazivModela; // Ovde stavite model koji odgovara
stranici

class NazivStraniceBuilder extends BasePageBuilder

    * CSS stilovi specifični za ovu stranicu
    * Ovde ide sav CSS koji stilizuje izgled stranice.
    * Može uključivati gridove, hover efekte, dugmad, boje, fontove,
      animacije itd.

    protected string $css = <<<CSS
    /* Primer CSS-a */
    .example-container {
    primer: 2rem;
    }
    CSS;

    * HTML struktura stranice
    * Ovde ide glavni HTML sadržaj stranice.
    * Može uključivati petlje za prikaz podataka iz baze, sekcije, dugmad,
      tabele itd.

    protected string $html = <<<'HTML '
    <main>
    <section class="example-container">
    <h2 class="text-2xl font-bold mb-4">Naslov Stranice</h2>
    <!-- Ovde možete iterirati podatke iz modela -->

```

```

<?php foreach ($items as $item): ?>
<div class="example-item">
<h3><?= htmlspecialchars($item["title"]) ?></h3>
<p><?= htmlspecialchars($item["description"]) ?></p>
</div>
<?php endforeach; ?>
</section>
</main>
HTML;

```

* Metoda za generisanje sadržaja stranice

* Ovde ide dodatni PHP kod za pripremu podataka koje stranica koristi.

```

public function buildPage(): string
{
    // Dodatni PHP kod specifičan za stranicu
    $additionalPHP = <<<PHP
    // Kreirajte instancu modela i učitajte podatke
    \ $model = new NazivModela();
    [\ $items, \ $totalCount] = \ $model->list(
    limit: 10, // primer paginacije
    offset: 0
    );
    PHP;

    // Generisanje sadržaja stranice
    $content = $this->getHeader($this->css, $additionalPHP);
    $content .= $this->getCommonIncludes(); // zajednički JS/CSS ili
template uključivanja
    $content .= $this->html; // glavni HTML sadržaj
    $content .= $this->getFooter(); // footer stranice

    return $content;
}
}

```

Za implementaciju novo kreiranih stranica u **PageExporter.php**, koriste se metode koje automatski određuju tip stranice i kreiraju odgovarajući PageBuilder. `determinePageType` analizira naziv stranice i vraća tip na osnovu ključnih reči, dok `getPageBuilder` instancira odgovarajuću klasu. Ako tip nije prepoznat, koristi se podrazumevani `BasicPageBuilder`, što omogućava modularno i jednostavno generisanje novih stranica.

```

private function determinePageType(string $name): string
{
    $name = strtolower($name);
    if (strpos($name, 'galerija') !== false) {
        return 'galerija';
    } elseif (strpos($name, ...) !== false) {
        return ....;
    }

    return 'basic';
}

```

```
}  
private function getPageBuilder(string $pageType, string $name):  
BasePageBuilder  
{  
    switch (strtolower($pageType)) {  
  
        case 'galerija':  
            return new GalleryPageBuilder($name, $this->data);  
        case ...:  
            return new ...($name, $this->data);  
        default:  
            return new BasicPageBuilder($name, $this->data);  
    }  
}
```