10

Consolidando o seu conhecimento

Chegou a hora de você seguir todos os passos realizados por mim durante esta aula. Caso já tenha feito, excelente. Se ainda não, é importante que você execute o que foi visto nos vídeos para poder continuar com a próxima aula.

Obs: Para facilitar disponibilizamos o projeto inicial já criado com um primeiro controlador. Você pode baixar esse projeto aqui (https://caelum-online-public.s3.amazonaws.com/1246-phpmvc/07/1246-phpmvc-psrs-inicial.zip) e ir diretamente para o desafio da aula.

1) No seu editor de código crie um novo projeto (pasta) chamado de `psrs`.

2) Na raiz do novo projeto crie um novo arquivo `composer.json` com o conteúdo abaixo:

```
{
    "autoload": {
        "psr-4": {
            "Alura\\Cursos\\": "src/"
        }
    },
    "require": {
        "doctrine/orm": "^2.6",
        "psr/http-message": "^1.0",
        "nyholm/psr7": "^1.1",
        "nyholm/psr7-server": "^0.3.0",
        "psr/http-server-handler": "^1.0",
        "php-di/php-di": "^6.0"
    }
}
```

3) Abra um terminal e entre na pasta `psrs`. Execute o comando:

```
composer install
```

4) Na raiz do novo projeto crie as pastas:

- `config`
- `public`
- `src`
- `view`

5) Agora dentro do `src` crie as pastas:

- `Controller`
- `Entity`
- `Helper`
- `Infra`

6) Na pasta `src\Infra` copie o arquivo `EntityManagerCreator` do projeto anterior:

```
<?php

namespace Alura\Cursos\Infra;

use Doctrine\ORM\EntityManager;
use Doctrine\ORM\EntityManagerInterface;
use Doctrine\ORM\Tools\Setup;

class EntityManagerCreator
{
    public function getEntityManager(): EntityManagerInterface
    {
        $paths = [__DIR__ . '/../Entity'];
        $isDevMode = false;

        $dbParams = array(
            'driver' => 'pdo_sqlite',
            'path' => __DIR__ . '/../../db.sqlite'
        );

        $config = Setup::createAnnotationMetadataConfiguration(
            $paths,
            $isDevMode
        );
        return EntityManager::create($dbParams, $config);
    }
}
```

7) Na raiz do seu novo projeto copie o arquivo `db.sqlite` do projeto anterior.

8) Dentro da pasta `config` crie o arquivo `routes.php` :

```
<?php

use Alura\Cursos\Controller\FormularioInsercao;

return [
    '/novo-curso' => FormularioInsercao::class
];
```

9) Ainda na pasta `config` crie mais um arquivo PHP, o `dependencies.php` :

```
<?php

$builder = new DI\ContainerBuilder();
$builder->addDefinitions([
    \Doctrine\ORM\EntityManagerInterface::class => function () {
        return (new \Alura\Cursos\Infra\EntityManagerCreator())
            ->getEntityManager();
    }
]);
$container = $builder->build();
```

```
return $container;
```

10) Na pasta `public` do seu projeto crie o arquivo `index.php` com o conteúdo abaixo:

```
<?php

require __DIR__ . '/../vendor/autoload.php';

use Nyholm\Psr7\Factory\Psr17Factory;
use Nyholm\Psr7Server\ServerRequestCreator;
use Psr\Container\ContainerInterface;
use Psr\Http\Server\RequestHandlerInterface;

$caminho = $_SERVER['PATH_INFO'];
$rotas = require __DIR__ . '/../config/routes.php';

if (!array_key_exists($caminho, $rotas)) {
    http_response_code(404);
    exit();
}

session_start();

//$ehRotaDeLogin = stripos($caminho, 'login');
//if (!isset($_SESSION['logado']) && $ehRotaDeLogin === false) {
//    header('Location: /login');
//    exit();
//}

$psr17Factory = new Psr17Factory();

$creator = new ServerRequestCreator(
    $psr17Factory, // ServerRequestFactory
    $psr17Factory, // UriFactory
    $psr17Factory, // UploadedFileFactory
    $psr17Factory  // StreamFactory
);

$serverRequest = $creator->fromGlobals();

$classeControladora = $rotas[$caminho];
/** @var ContainerInterface $container */
$container = require __DIR__ . '/../config/dependencies.php';
/** @var RequestHandlerInterface $controlador */
$controlador = $container->get($classeControladora);

$resposta = $controlador->handle($serverRequest);

foreach ($resposta->getHeaders() as $name => $values) {
    foreach ($values as $value) {
        header(sprintf('%s: %s', $name, $value), false);
    }
}
```

```
echo $resposta->getBody();
```

11) Dentro da pasta `src\Controller` crie um novo arquivo `FormularioInsercao` :

```
<?php

namespace Alura\Cursos\Controller;

use Alura\Cursos\Helper\RenderizadorDeHtmlTrait;
use Nyholm\Psr7\Response;
use Psr\Http\Message\ResponseInterface;
use Psr\Http\Message\ServerRequestInterface;
use Psr\Http\Server\RequestHandlerInterface;

class FormularioInsercao implements RequestHandlerInterface
{
    private $entityManager:

    public function __construct(EntityManagerInterface $entityManager)
    {
        $this-> entityManager = $entityManager;
    }

    public function handle(ServerRequestInterface $request): ResponseInterface
    {
        //var_dump($this->entityManager);
        $html = "teste";
        return new Response(200, [], $html);
    }
}
```

12) Na linha de comando, na raiz do seu projeto rode o servidor:

```
php -S localhost:8080 -t public
```

Teste a URL abaixo no navegador:

```
http://localhost:8080/novo-curso
```