UTFPR

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DIO CEZAR GO

Composer

O QUE É?

- o composer é uma ferramenta para gerenciamento de dependências para o PHP;
- com poucas linhas de configurações você define todas as bibliotecas de terceiros ou até mesmo suas próprias bibliotecas;
- o composer pode ser considerado um `npm` para o PHP;

COMO COMEÇO?

- a primeira parte é fazer o download do phar do composer;
- phar é um empacotamento de uma aplicação;
  - é pegar e usar!
- o que deve ser feito na verdade é a instalação da biblioteca em seu sistema operacional para que seja possível o acesso ao comando: `composer`

COMO INSTALAR?

- instalando via terminal:
  - `curl -sS https://getcomposer.org/installer | php`

CONCEITOS BÁSICOS

- o composer facilita o gerenciamento de dependência em seus projetos;
  - isso surge com a necessidade padronização e interoperabilidade entre vários frameworks disponíveis no mercado;
- mas o não limita-se ao uso de frameworks;
- pode-se utilizá-lo para obter bibliotecas, classes outras funcionalidades para se trabalhar com o seu PHP puro, se for o caso;
- uma recomendação é: seguir a http://www.php-fig.org
  - propõe uma série de recomendações para o desenvolvimento em PHP;
  - entre elas o uso de namespaces e autoloaders;
  - https://github.com/php-fig/fig-standards/blob/master/accepted/PSR-4-autoloader.md

MÃOS NA MASSA

- o primeiro passo é a criação de um arquivo chamado composer.json;
- este arquivo possuirá as configurações de dependências de sua aplicação no formato JSON;

EXEMPLO BÁSICO DO COMPOSER.JSON

COMPOSER.JSON

```
{
    "name": "Nome do projeto",
    "description": "Breve descrição do que a aplicação se propoe a fazer",
    "authors": [
        {
            "name": "Seu nome",
            "email": "seu-email@seu-dominio.com"
        }
    ],
    "require": {
        "php": ">=5.2.8"
    }
}
```

se a versão do PHP for abaixo da 5.2.8 uma mensagem de erro será lançada ao instalar as dependências informando que não é possível prosseguir

E AGORA?

- agora já temos o esqueleto e precisamos incluir alguns pacotes;
- o composer utiliza como repositório o packagist (https://packagist.org/)
- no qual, qualquer desenvolvedor pode criar seus pacotes e disponibilizá-los para a comunidade;

UM EXEMPLO

vamos instalar uma biblioteca chamada slug.php;

COMPOSER.JSON

```
...
"require": {
        "php": ">=5.2.8",
        "kevinlebrun/slug.php": "1.*"
}
...
```

E AGORA?

- agora estamos prontos para ver o composer em ação;
- na pasta raíz da sua aplicação (onde está o composer.json)
  rode o comando → `composer install`
- este comando irá ler todas as dependências e instalá-las automaticamente;
- se algum dos pacotes instalados, for dependente de outro pacote, ele também será instalado;
- note que agora, existe uma pasta chamada `vendor`
  - é nessa pasta em que os pacotes estão instalados;

PRÓXIMO PASSO

- com tudo pronto e baixado, podemos criar um exemplo para utilização da biblioteca em questão;
- o composer utiliza um sistema automático de inclusão das classes e bibliotecas que foram instaladas;
- para isso, basta chamar → `require 'vendor/autoload.php';`

ESCREVENDO O PHP

agora podemos escrever nosso código usando a biblioteca em questão:

INDEX.PHP

```
<?php
     header('Content-Type: text/html; charset=utf-8');
     require 'vendor/autoload.php';
     $slugifier = new \Slug\Slugifier();
     $slugifier->setTransliterate(true);
     $frase = 'Frase com acentuação para teste de criação de slug';
     $slug = $slugifier->slugify($frase);
     echo '<b>Frase natural: </b>' . $frase . "<br /><br />";
     echo '<b>Frase com aplicação de slug: </b>' . $slug . "<br /><br />";
?>
```

E PRONTO!

- sempre que for necessário a atualização de pacotes em seu `composer.json` basta chamar o comando → `composer update`
- além disso, o próprio composer pode fazer sua atualização, com o comando `composer self-update`

MATERIAIS COMPLEMENTARES

- https://tableless.com.br/composer-para-iniciantes/