Fluxo dos dados (estrutura MVC)

Na tela do filme selecionado do catalogo é solicitado a função static do controller para criar o objeto contendo as informações do filme.

// Criação do objeto do com dados do filme

if(isset($\_GET['id'])){

$idFilme = $\_GET['id'];

$idFilmeJson = json\_encode($idFilme);

$novo\_obj = controller::getFilmePorId($idFilme);

ob\_end\_clean();

$especificacoes = get\_object\_vars($novo\_obj);

$idImg = $especificacoes['episodio\_id'];

$spec = json\_encode($especificacoes);

?>

<script>

console.log(<?php echo $spec; ?>);

var idFilme = <?php echo $idFilmeJson; ?>

</script>

No arquivo controller é feito a requisição da api e é criado o objeto com os dados do filme através da classe “filme”.

// CRIAÇÃO DO OBJETO FILMES QUE FAZ A REQUISIÇÃO DA API

try {

// Inicializa o cURL

$ch = curl\_init();

curl\_setopt($ch, CURLOPT\_URL, $api\_url);

curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 1);

// Executa a requisição

$jsonData = curl\_exec($ch);

// Verifica se ocorreu algum erro

if (curl\_errno($ch)) {

throw new Exception(curl\_error($ch));

}

curl\_close($ch);

// Decodifica o JSON recebido

$data = json\_decode($jsonData, true);

// Cria um objeto com os dados do arquivo JSON

$obj\_filme = new filme(

$data['title'],

$data['episode\_id'],

$data['opening\_crawl'],

$data['release\_date'],

$data['director'],

$data['producer'],

$data['characters']

);

return $obj\_filme;

} catch (Exception $e) {

// Lida com possíveis exceções e exibe uma mensagem de erro

echo 'Erro ao acessar a API: ' . $e->getMessage();

}

O arquivo listarFilmes é o endpoint para requisitar a api, aqui é criado um objeto json com os dados do filme, que será a resposta para o controller.

<?php

require 'api\_logic.php';

try {

if(isset($\_GET['id'])){

$id = $\_GET['id'];

$filme = listarFilmes($id);

ob\_start();

echo json\_encode($filme);

}

} catch (Exception $e) {

echo 'Erro: ' . $e->getMessage();

}

?>

O arquivo api logic é onde ocorre a conexão com a api externa de star wars.

<?php

/\* LISTAR FILMES ---------------------------------------------------------------------------------- \*/

function listarFilmes($filme\_id){

$url = "<http://swapi.py4e.com/api/films/>";

$ch = curl\_init($url);

curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, true);

curl\_setopt($ch, CURLOPT\_SSL\_VERIFYPEER, false);

$filme = json\_decode(curl\_exec($ch), true);

$count = $filme['count'];

for($i = 0; $i<$count; $i++){

if($filme['results'][$i]['episode\_id'] == $filme\_id){

// echo $filme['results'][$i]['title'] . "<br>";

// echo $filme['results'][$i]['opening\_crawl'];

$listaFilmes = $filme['results'][$i];

return $listaFilmes;

}

}

}

Envio dos logs de requisição da api:

para funcionar é necessário instalar o gerenciador de dependências composer e a biblioteca monolog.

No prompt execute:

composer require monolog/monolog

<?php

require\_once "../vendor/autoload.php";

use Monolog\Logger;

use Monolog\Handler\StreamHandler;

use Monolog\Formatter\JsonFormatter;

$logger = new Logger('api\_access');

$stream = new StreamHandler('api\_access.log', Logger::INFO);

$stream->setFormatter(new JsonFormatter());

$logger->pushHandler($stream);

function log\_api\_request($method, $endpoint, $params, $response\_code, $response\_body, $ip, $user\_agent) {

$context = [

'method' => $method,

'endpoint' => $endpoint,

'params' => $params,

'response\_code' => $response\_code,

'response\_body' => $response\_body,

'ip' => $ip,

'user\_agent' => $user\_agent

];

return $context;

}

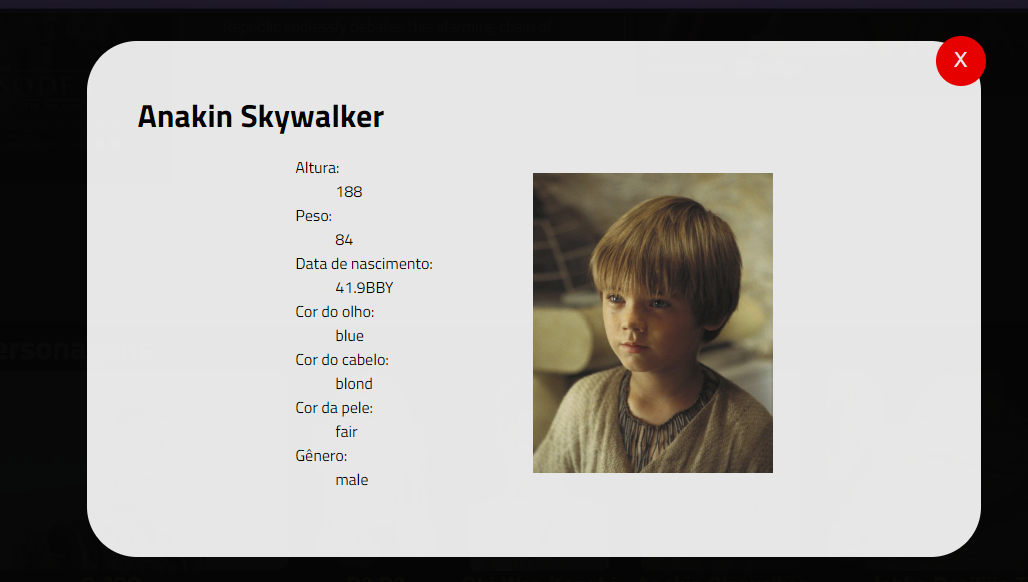
?>

Melhorias

Botão para alternar ordem do catálogo entre ordem de lançamento ou cronológica



Janela modal para exibir dados dos personagens



Exibição do trailer do filme

