

Criando a API em PHP

1) Criando a API

Vamos criar agora o arquivo api.php na mesma pasta do arquivo index.html. Copie o código abaixo para este novo arquivo:

```
<?php

$frase = "Esta é uma frase de teste";
$qt_chars = strlen($frase);
$meu_ip = $_SERVER['REMOTE_ADDR'];
date_default_timezone_set('America/Sao_Paulo');
$hoje = date("d/m/Y H:i:s");

echo 'A frase "' . $frase . '" contém ' . $qt_chars . ' caracteres<br>';
echo 'IP: ' . $meu_ip . '<br>';
echo 'Agora é ' . $hoje;
```

?>

Veja o que cada função acima faz:

strlen() : retorna o número de caracteres de uma string

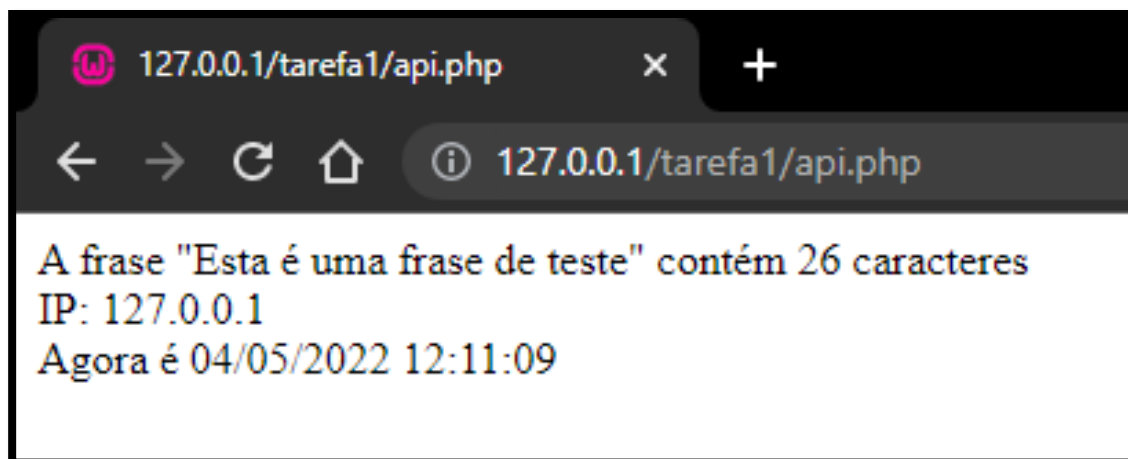
\$_SERVER['REMOTE_ADDR'] : retorna o valor do IP do servidor de conexão à internet do cliente.

date_default_timezone_set('America/Sao_Paulo') : defini o horário de acordo com a região no globo.

date("d/m/Y H:i:s") : retorna a data e hora atual no formato desejado.

Para “rodar” o programa acima, no navegador digite o endereço:

http://127.0.0.1/tarefa1/api.php



127.0.0.1 é por padrão o endereço do servidor de internet local (o Apache do Wampp).

2) Modificando a API - retorno no formato JSON

Vamos adotar o formato JSON que se trata apenas de um formato padrão para troca de informações.

Comente as linhas “echo” e inclua a conversão para JSON:

```
<?php

$frase = "Esta é uma frase de teste";
$qt_chars = strlen($frase);
$meu_ip = $_SERVER['REMOTE_ADDR'];
$hoje = date("d/m/Y H:i:s");

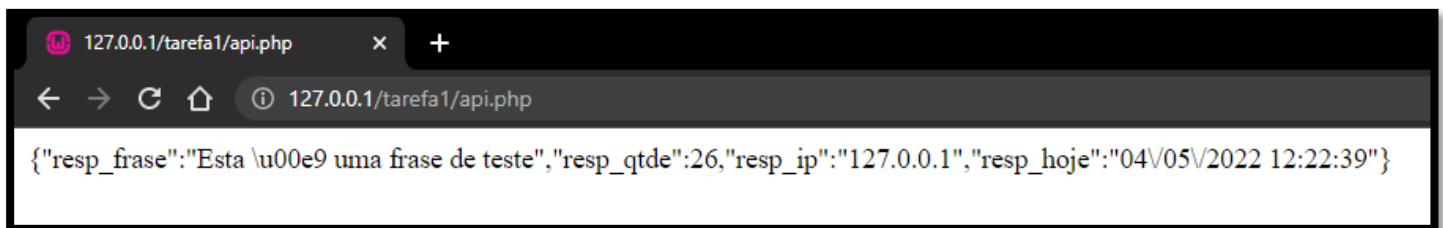
// echo 'A frase "' . $frase . '" contém ' . $qt_chars . ' caracteres<br>';
// echo 'IP: ' . $meu_ip . '<br>';
// echo 'Agora é ' . $hoje;

$response = array(
    "resp_frase"=>$frase,
    "resp_qtde"=>$qt_chars,
    "resp_ip"=>$meu_ip,
    "resp_hoje"=>$hoje,
);

echo json_encode($response);

?>
```

A variável \$response será um ARRAY formado pelas “chaves” e respectivos valores. A função json_encode() transforma um ARRAY no formato JSON. O resultado na tela deverá ser:



Repare que o formato JSON coloca os valores entre CHAVES {} e internamente exibe uma chave e um valor associado:

```
{
    "resp_frase" : "Esta \u00e9 uma frase de teste",
    "resp_qtde" : 26,
    "resp_ip" : "127.0.0.1",
    "resp_hoje" : "04/05/2022 12:22:39"
}
```

3) “Chamando” a API com JQuery

Referência: https://www.w3schools.com/jquery/ajax_post.asp

Edite agora o seu index.html, troque o conteúdo da função enviar_api() pelo código:

```

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script> (1)

<script>

function enviar_api(){
    frase_digitada = $("#um_texto_qualquer").val(); //(2)

    $.post("api.php", {frase_a_enviar: frase_digitada}, function(result){ //(3)
        console.log(result); //(4)

        obj = JSON.parse(result); //(5)

        console.log(obj.resp_qtde); //(6)
        console.log(obj.resp_ip);
        console.log(obj.resp_hoje);

        $("#div_resposta_data").html(obj.resp_hoje); //(7)
        $("#div_resposta_ip").html(obj.resp_ip);
        $("#div_resposta_qtde").html(obj.resp_qtde);

    });

}
</script>

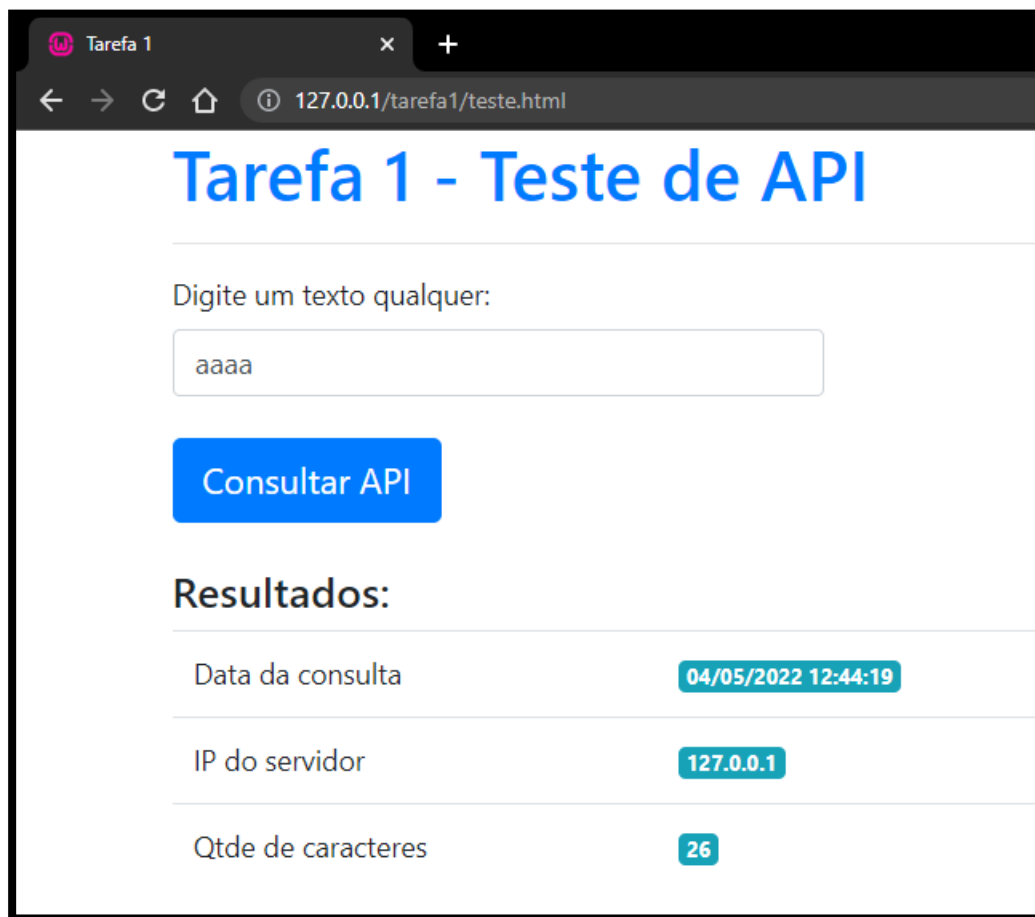
```

Comentários:

- (1) Altere este código para carregar a versão “full” do jquery. A código anterior era uma versão “slim” e não continha as funções para se comunicar com scripts externos (AJAX, POST, GET,...).
- (2) Código JQuery para armazenar o conteúdo da frase digitada cujo ID é “#um_texto_qualquer” na variável “frase_digitada”.
- (3) Função “POST” do JQuery, resumidamente essa função envia para o script “api.php” o conteúdo da variável “frase_digitada” atribuindo o nome “frase_a_enviar” e a resposta é armazenada na variável “result”. A “resposta” é o que aparecia na tela quando você rodava api.php.
- (4) Console.log() exibe na tela do console algum conteúdo com o propósito apenas para monitorar.
- (5) Converte o conteúdo de uma string (a resposta JSON lá da nossa API) em um objeto, assim podemos recuperar valores independentes.
- (6) Apenas para monitorar (verificar no console) cada atributo do objeto “obj” convertido acima.
- (7) Coloca cada valor recebido a uma tag DIV referente a cada ID correspondente. Exibindo assim os resultados.

No navegador, execute o index.html (127.0.0.1/tarefa1/index.html). Digite alguma frase e clique no botão Consultar API.

A seguinte tela aparecerá:



Ainda precisamos alterar a `api.php` para poder receber a frase digitada que foi enviada pelo comando JQuery abaixo:

```
$.post("api.php",{frase_a_enviar:frase_digitada},function(result){...
```

Ao clicar no botão "Consultar API", o JQuery executará o comando acima e o script `api.php` receberá a variável "frase_a_enviar" e respectivo valor.

Para receber esta variável, altere esta linha:

```
$frase = "Esta é uma frase de teste";
```

Para

```
$frase = $_POST["frase_a_enviar"];
```

Com a função `$_POST` do php podemos receber o que foi enviado pela função `$.POST` do JQuery (Javascript).

Experimente agora digitar algumas frases.

De uma forma muito simples e ainda sem considerar a segurança dos dados e acesso, este é um exemplo básico de como podemos implementar uma API em PHP e consultar ela por meio de Javascript.