



Roteiro de Laboratório: Cliente PHP para Consumo de WebService Spring Boot

Objetivo

Criar um cliente em PHP que consuma os endpoints da API REST de calculadora desenvolvida em Spring Boot no laboratório anterior.

Pré-requisitos




- Serviço Spring Boot da calculadora em execução (endereço: <http://localhost:8080/>)
- PHP instalado (versão 8.0 ou superior)
- Acesso a um servidor web (Apache, Nginx) ou PHP embutido (php -S localhost:8000)

Passo a Passo

1. Criar a Estrutura do Projeto

Crie uma pasta chamada cliente-php e dentro dela:

cliente-php/

- ├──  index.php (Frontend HTML + PHP)
- ├──  calcular.php (Lógica de consumo da API)
- └──  resultado.php (Exibição do resultado)

2. Criar o Frontend (index.php)

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Cliente PHP - Calculadora</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 20px; }
    .container { max-width: 400px; margin: auto; }
    input, select, button { margin: 5px 0; padding: 8px; width: 100%; }
    button { background-color: #4CAF50; color: white; border: none; cursor: pointer; }
    button:hover { background-color: #45a049; }
  </style>
</head>
<body>
  <div class="container">
    <h2>Calculadora via API Spring Boot</h2>
    <form action="calcular.php" method="GET">
      <input type="number" name="num1" placeholder="Número 1" required>
      <input type="number" name="num2" placeholder="Número 2" required>
      <select name="operacao">
        <option value="somar">Somar</option>
        <option value="subtrair">Subtrair</option>
        <option value="multiplicar">Multiplicar</option>
      </select>
    </form>
  </div>
</body>
</html>
```

```

        <option value="dividir">Dividir</option>
    </select>
    <button type="submit">Calcular</button>
</form>
</div>
</body>
</html>

```

3. Criar o Backend (calcular.php)

```

<?php
$num1 = $_GET['num1'] ?? 0;
$num2 = $_GET['num2'] ?? 0;
$operacao = $_GET['operacao'] ?? 'somar';

$url = "http://localhost:8080/$operacao/$num1/$num2";

// Inicializa o cURL
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

// Executa a requisição
$resposta = curl_exec($ch);

// Verifica erros
if (curl_errno($ch)) {
    die("Erro na requisição: " . curl_error($ch));
}
curl_close($ch);

// Redireciona para a página de resultado
header("Location: resultado.php?resposta=" . urlencode($resposta));
?>

```

4. Exibir o Resultado (resultado.php)

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>Resultado</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 20px; }
        .container { max-width: 400px; margin: auto; }
        a { color: #4CAF50; text-decoration: none; }
    </style>
</head>
<body>
    <div class="container">
        <h2>Resultado da Operação</h2>
        <p><?php echo htmlspecialchars($_GET['resposta'] ?? 'Nenhum resultado encontrado.');

```

</html>

5. Executando o Cliente PHP

Inicie o servidor PHP embutido (caso não use Apache/Nginx):

```
php -S localhost:8000 -t cliente-php/
```

Acesse no navegador:

```
http://localhost:8000/
```

Teste a calculadora:

Insira dois números.

Selecione a operação.

Veja o resultado retornado pela API Spring Boot.

Resultado Esperado

- O cliente PHP enviará requisições para a API Spring Boot.
- A API processará e retornará o resultado.
- O PHP exibirá o resultado na página.

Possíveis Erros e Soluções

- API não encontrada: Verifique se o Spring Boot está rodando em `http://localhost:8080/`.
- Erro no cURL: Instale a extensão cURL no PHP (`sudo apt install php-curl` no Linux).
- Problemas com números decimais: Use `floatval()` no PHP para garantir números decimais.