

## Que tal enviar dados para o servidor?

Durante o treinamento, lidamos apenas com requisições do tipo GET, porque eu pressupunha que você já conhecia o XMLHttpRequest . Contudo, para dar um brilho ainda maior em sua experiência, vamos praticar um pouquinho mais. Desta vez, realizaremos uma requisição do tipo POST para enviar dados para o servidor. Só fique atento com a volatilidade dos dados cadastrados, isto é, se você reiniciar o servidor, perderá os dados enviados. Aliás, não há problema nenhum nisso, porque a ideia aqui é focar o código no lado do cliente, ou seja, código JavaScript e não código do back-end.

Para conseguir realizar uma requisição do tipo POST você precisa:

- HTTP POST para /negociacoes ;
- Usar o cabeçalho Content-type sendo application/json ;
- No body um JSON da negociação.

Já preparamos um HTML bem básico para começar o cadastro! Para usá-lo no projeto **crie uma nova página** post.html , dentro do diretório aluraframe/client :

```
<!-- aluraframe/client/post.html -->
<html>
<head>
  <meta charset="UTF-8">
  <title>Negociações</title>
  <link rel="stylesheet" href="css/bootstrap.css">
  <link rel="stylesheet" href="css/bootstrap-theme.css">
</head>

<body class="container">
  <form class="form">

    <div class="form-group">
      <label for="data">Data</label>
      <input type="date" id="data" class="form-control" required autofocus/>
    </div>

    <div class="form-group">
      <label for="quantidade">Quantidade</label>
      <input type="number" min="1" step="1" id="quantidade" class="form-control" value="1">
    </div>

    <div class="form-group">
      <label for="valor">Valor</label>
      <input id="valor" type="number" class="form-control" min="0.01" step="0.01" value='

    <button class="btn btn-primary" type="submit" onclick="sendPost(event)">Enviar dados pa
  </form>

  <script>

    function sendPost(event) {
```

```
event.preventDefault();
console.log("Enviando post");

//aqui você deve ler os dados do formulário
//construir o json
//enviar o XMLHttpRequest
}
</script>
</body>
</html>
```

Se você achou que eu daria a solução logo de cara, se enganou! Contudo, depois de quebrar a cabeça, veja a solução do problema na minha resposta :)

Obs: Não esqueça de subir o servidor com `npm start`. Para acessar a página `post.html`, basta digitar `localhost:3000/post.html`