

Utilizando JavaScript com Spring Boot e Thymeleaf



**Certified
Developer**
The Ultimate Tech Degree

DigitalHouse >



Introdução

Agora que sabemos como construir APIs, vamos ver uma opção para consumí-las a partir da view utilizando javascript.

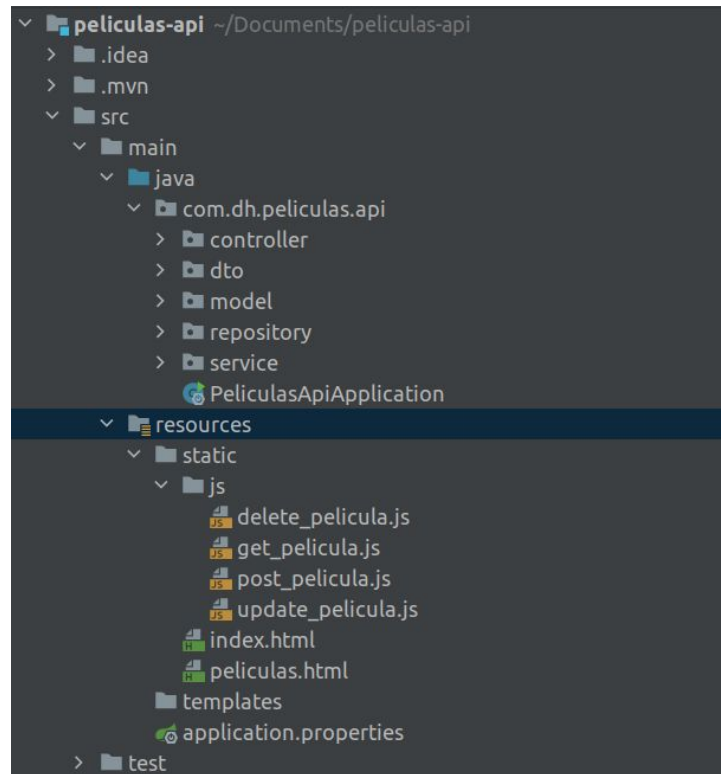
Para explicar como podemos fazer isso, vamos adicionar uma view ao projeto da API de filmes que desenvolvemos anteriormente.





Estrutura do projeto

Dentro da pasta static teremos as páginas HTML e o código javascript, que neste caso, agrupamos dentro da pasta js, onde estão localizados todos os arquivos .js.





Incluir um filme

Neste caso, vamos construir um formulário HTML para poder registrar (cadastrar) um filme.

```
<div class="row">
  <div class="col-sm-7" style="background-color:#e6fffa; padding:10px;
    border-radius:3px">
    <h3>Incluir filme</h3>
    <form id="add_new_filme">
      <div class="form-group">
        <label class="control-label" for="titulo">Título:</label>
        <input type="text" class="form-control" id="titulo"
          placeholder="Informe o título" name="titulo" required></input>
      </div>
    </form>
  </div>
</div>
```



```
<div class="form-group">
  <label class="control-label" for="categoria">Categoria:</label>
  <input type="text" class="form-control" id="categoria"
    placeholder="Informe a categoria" name="categoria" required></input>
</div>
<div class="form-group">
  <label class="control-label" for="premios">Prêmios:</label>
  <input type="number" class="form-control" id="premios"
    placeholder="Informe a quantidade de prêmios" name="premios"></input>
</div>
<button type="submit" class="btn btn-danger" id="btn-add-new-pelicula">Incluir</button>
</form>
<div id="response" style="display:none; margin-top:10px">
  </div>
</div></div>
```



Incluir um filme

Criamos uma função em **JavaScript** que se encarregará de ler os campos do formulário e montar o objeto **formData**.

```
const formulario = document.querySelector('#add_new_filme');  
formulario.addEventListener('submit', function (event) {  
  event.preventDefault();  
  const formData = {  
    titulo: document.querySelector('#titulo').value,  
    categoria: document.querySelector('#categoria').value,  
    premios: document.querySelector('#premios').value,  
  };  
});
```



Com o **fetch** enviamos a solicitação indicando a url, o método (POST) e o objeto filme (formData) em formato json utilizando o método **JSON.stringify()**.

```
const url = '/filmes';

const settings = {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
  },
  body: JSON.stringify(formData)
}

fetch(url, settings)
```



```
.then(response => response.json())
.then(data => {
    let successAlert = '<div class="alert alert-success alert-dismissible">' +
        '<button type="button" class="close" data-dismiss="alert">&times;</button>' +
        '<strong></strong> Filme incluído </div>'
    $("#response").append(successAlert);
    $("#response").css({"display": "block"});
    resetUploadForm();
})
.catch(error => { let errorAlert = '<div class="alert alert-danger alert-dismissible">' +
    '<button type="button" class="close" data-dismiss="alert">&times;</button>' +
    '<strong> Erro, tente novamente</strong> </div>'
    $("#response").append(errorAlert);
    $("#response").css({"display": "block"});
    resetUploadForm();});
});
```




Listar filmes

Neste caso vamos montar uma **tabela em HTML** para poder **obter a lista de filmes**.

```
<div id="div_filme_table">
  <h1>Filmes</h1>
  <div class="row col-md-6 table-responsive">
    <table id="filmeTable" class="table table-striped">
      <thead><tr><th>Id</th><th>Título</th><th>Categoria</th><th></th></tr></thead>
      <tbody>
      </tbody>
    </table>
  </div>
</div>
```



Listar filmes

Em seguida, enviamos uma solicitação do tipo **GET**, e definimos a url.

```
(function(){  
  event.preventDefault();  
  const url = '/filmes';  
  const settings = {  
    method: 'GET'  
  }  
  fetch(url, settings)
```



No caso de uma resposta correta, criamos uma função que injetará o HTML na tabela para criar a lista com a resposta da API (**response.json()**).

```
.then(response => response.json())
.then(data => {
  $.each(data, (i, filme) => {

    let deleteButton = '<button ' +
                      'id=' +
                      '\"' + 'btn_delete_' + filme.id + '\"'+
                      ' type="button" class="btn btn-danger btn_delete"
data-toggle="modal" ' +
                      'data-target="#delete-modal" ' +
                      '>&times</button>';
```

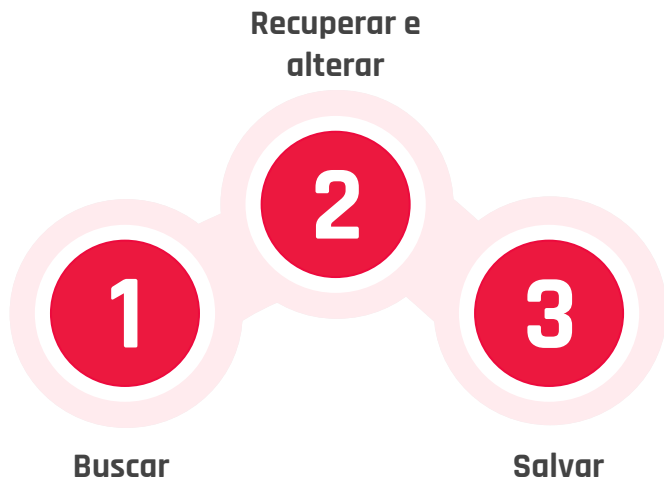


```
let get_More_Info_Btn = '<button' +  
    ' id=' + '\"' + 'btn_id_' + filme.id + '\"' +  
    ' type="button" class="btn btn-info btn_id">' +  
    filme.id +  
    '</button>';  
  
let tr_id = 'tr_' + filme.id;  
let filmeRow = '<tr id=\"' + tr_id + '\">' +  
    '<td>' + get_More_Info_Btn + '</td>' +  
    '<td class=\"td_titulo\">' + filme.titulo.toUpperCase() + '</td>' +  
    '<td class=\"td_categoria\">' + filme.categoria + '</td>' +  
    '<td>' + deleteButton + '</td>' +  
    '</tr>';  
$('#filmeTable tbody').append(filmeRow);  
});  
  
})  
})
```



Alterar um filme

Para modificar um filme, primeiro devemos **buscá-lo** pelo seu id e **recuperar** todos os dados do filme para colocá-lo em um formulário e fazer as modificações no(s) campo(s) desejado(s) para depois **salvar** essas alterações.





Formulário

```
<div class="row">
  <div class="col-sm-6" style="display: none; background-color:#e6fffa;
    padding:10px; border-radius:3px"
    id="div_filme_updating">
    <form id="update_filme_form">
      <div class="form-group">
        <label for="filme_id">Id:</label>
        <input type="text" class="form-control" id="filme_id" readonly>
      </div>
      <div class="form-group">
        <label for="filme_titulo">Titulo:</label>
        <input type="text" class="form-control" placeholder="Título"
          id="titulo">
      </div>
    </form>
  </div>
</div>
```



```
<div class="form-group">
    <label for="filme_categoria">Categoria:</label>
    <input type="text" class="form-control"
        placeholder="Categoria" id="categoria">
</div>
<div class="form-group">
    <label for="filme_premios">Prêmios:</label>
    <input type="number" class="form-control"
        placeholder="Premios" id="premios">
</div>
<button type="submit" class="btn-primary">Alterar</button>
```



```
    </form>  
    <div id="response" style="display:none; margin:10px">  
    </div>  
  </div>  
</div>
```





Buscar um filme

Neste caso, enviamos um pedido **GET** para url/filmes/{id}. Essa função será executada com o clique do usuário em um filme da lista.

```
function find(id) {  
  
  const url = '/filmes/' + id;  
  const settings = {  
    method: 'GET'  
  }  
  fetch(url, settings)
```



Recuperar um filme

A função que processa a resposta da API injetará os dados nos campos do formulário.

```
.then(response => response.json())
.then(data => {
    let filme = data;
    $("#filme_id").val(filme.id);
    $("#titulo").val(filme.titulo);
    $("#categoria").val(filme.categoria);
    $("#premios").val(filme.premios);
    $("#div_filme_updating").css({"display": "block"});
}).catch(error => {
    console.log(error);
    alert("Error -> " + error);
})
}
```



Salvar o filme

Similar a função de registrar (cadastrar) um filme, lemos os campos do formulário HTML, criamos o objeto e o enviamos para o controller, neste caso utilizando um método do tipo **PUT**.

```
window.addEventListener('load', function () {  
  const formulario = document.querySelector('#update_filme_form');  
  formulario.addEventListener('submit', function (event) {  
    let filmeId = document.querySelector('#filme_id').value;  
    const formData = {  
      id: document.querySelector('#filme_id').value,  
      titulo: document.querySelector('#titulo').value,  
      categoria: document.querySelector('#categoria').value,  
      premios: document.querySelector('#premios').value,  
    };  
  });  
});
```



```
const url = '/filmes';  
const settings = {  
  method: 'PUT',  
  headers: {  
    'Content-Type': 'application/json',  
  },  
  body: JSON.stringify(formData)  
}  
  
fetch(url, settings)  
  .then(response => response.json())  
  .then(data => {
```



```
let filme = data;

let successAlert = '<div class="alert alert-success alert-dismissible">'
    '<button type="button" class="close" ' +
    'data-dismiss="alert">&times;</button>' +
    '<strong> pelicula = </strong> ' + data.toString(); +
    '</div>'

$("#tr_" + filmeId + "
    td.td_titulo").text(filme.titulo.toUpperCase());
$("#tr_" + filmeId + "
    td.td_categoria").text(filme.categoria.toUpperCase());
$("#response").empty();
$("#response").append(successAlert);
$("#response").css({"display": "block"});
```



```
    }).catch(error => {  
      let errorAlert = '<div class="alert alert-danger alert-dismissible">' +  
        '<button type="button" class="close" data-dismiss="alert">&times;</button>' +  
        '<strong> Error </strong></div>';  
  
      $("#response").empty();  
      $("#response").append(errorAlert);  
      $("#response").css({"display": "block"});  
    })  
  
  })  
}
```



DigitalHouse>