

Refatoração - API

No decorrer do curso, percebemos a importância de refatorar o nosso código. Como podemos aplicar o que aprendemos no trecho de código abaixo?

```
const ApiService = {
  ListaAutores : () =>{
    return fetch('http://localhost:8000/api/autor')
    .then(res => ApiService.TrataErros(res))
    .then(res => res.json());
  },
  CriaAutor : autor => {
    return fetch('http://localhost:8000/api/autor', {method: 'POST', headers: {'cc
    .then(res => ApiService.TrataErros(res))
    .then(res => res.json());;
  }
}
```

Selecione uma alternativa

A

```
const ApiUrl = 'http://localhost:8000/api/autor'

const ApiService = {
  ListaAutores : () =>{
    return fetch(ApiUrl)
    .then(res => ApiService.TrataErros(res))
    .then(res => res.json());
  },
  CriaAutor : autor => {
    return fetch(ApiUrl, {method: 'POST', headers: {'content-type': 'applicat
    .then(res => ApiService.TrataErros(res))
    .then(res => res.json());
  }
}
```

B

```
const consumeApi = (url, header, body) =>
  fetch(url, header = '', body = '')
    .then(res => ApiService.TrataErros(res))
    .then(res => res.json())

const ApiService = {
  ListaAutores: consumeApi('http://localhost:8000/api/autor'),
  CriaAutor: autor => consumeApi('http://localhost:8000/api/autor', {method: 'PC
}
```

C

```
const ApiUrl = 'http://localhost:8000/api/autor'
```

```
const consumeApi = (url, header, body) =>
  fetch(url, header = '', body = '')
    .then(res => ApiService.TrataErros(res))
    .then(res => res.json())

const ApiService = {
  ListaAutores : consumeApi(ApiUrl),
  CriaAutor : autor => consumeApi(ApiUrl, {method: 'POST', headers: {'content-t
}
```