

Implementando a função Log

Transcrição

Para que fosse possível escrutinarmos o valor retornado por uma função `then` antes que fosse recebido por outra encadeamos em nossa Promise uma chamada ao `then`. Reparem que é o mesmo código que utilizamos logo após o filtro da lista:

```
.then(itens => {  
  console.log(itens);  
  return itens;  
})
```

Podemos isolar essa funcionalidade em uma função no módulo `promise-helpers.js`:

```
// app/utils/promise-helpers.js  
  
export const handleStatus = res =>  
  res.ok ? res.json() : Promise.reject(res.statusText);  
  
export const log = param => {  
  console.log(param);  
  return param;  
};
```

Agora, em `app.js`, vamos importar a função e utilizá-la:

```
// app/app.js  
import { handleStatus, log } from './utils/promise-helpers.js';  
  
document  
  .querySelector('#myButton')  
  .onclick = () =>  
    fetch('http://localhost:3000/notas')  
      .then(handleStatus)  
      .then(notas => notas.reduce((array, nota) => array.concat(nota.itens), []))  
      .then(log)  
      .then(itens => itens.filter(item => item.codigo == '2143'))  
      .then(log)  
      .then(itens => itens.reduce((total, item) => total + item.valor, 0))  
      .then(console.log)  
      .catch(console.log);
```

Excelente! Menos código duplicado. Agora podemos partir para a próxima melhoria.