

01

Alto acoplamento

Transcrição

Começando deste ponto? Você pode fazer o [DOWNLOAD \(https://s3.amazonaws.com/caelum-online-public/772-js-padroao-funcional/stages/06-project.zip\)](https://s3.amazonaws.com/caelum-online-public/772-js-padroao-funcional/stages/06-project.zip) completo do projeto anterior e continuar seus estudos a partir daqui.

Nossa aplicação está exibindo a soma dos itens das notas no console, nada extraordinário. Mas se além de fazermos isso quisermos exibir um `alert` com total calculado? Teremos que alterar a `app.js` e adicionar toda a lógica interessada na soma dos itens na instrução `then`:

```
// app/app.js
// código anterior omitido
const action = operations(() =>
  retry(3, 3000, () => timeoutPromise(200, service.sumItems('2143')))
  .then(soma => {
    console.log(soma); // um interessado
    alert(soma); // outro interessado
  })
  .catch(console.log)
);
// código posterior omitido
```

Podemos ter mais de um interessado no total das notas, por exemplo, para renderizar a resposta do usuário e assim por diante. Para cada novo interessado, teremos que alterar o módulo `app.js` correndo o risco de quebrá-lo. Em suma, o módulo `app.js` terá mais de um motivo para ser alterado. Uma solução para o forte acoplamento aqui citado é o emprego do pattern **Publish/Subscribe**.