01

Uma conexão ou várias?

Transcrição

Começando deste ponto? Você pode fazer o DOWNLOAD (https://github.com/alura-cursos/javascript-avancado-iii/archive/aula1.zip) completo do projeto anterior e continuar seus estudos a partir daqui.

Anteriormente, aprendemos a interagir com o `indexedDB`, nós também incluímos e listamos negociações em uma Object Store. Mas temos que pensar na manutenção e legitimidade do nosso código, por isso, começaremos a organizá-lo. A seguir, atacaremos a criação da conexão, chamando duas vezes o `ConnectionFactory`:

```
ConnectionFactory
    .getConnection()
    .then(connection => {
    });

// faz outras coisas e pede novamente a conexão

ConnectionFactory
    .getConnection()
    .then(connection => {
    });
```

A) O método `getConnection()` será um método estático, ou seja, invocado diretamente na classe.

B) O retorno de `getConnection` será uma promise, pois a abertura de uma conexão é um processo assíncrono.

C) Não importa quantas vezes seja chamado o método `getConnection()`, a conexão retornada deve ser a mesma.

D) Toda conexão possui o método `close()`, mas o programador não pode chamá-lo, porque a conexão é a mesma para a aplicação inteira. Só o próprio `ConnectionFactory` pode fechar a conexão.

Para entender o que queremos dizer quando falamos que a conexão retornada pelo método `getConnection()` deve ser a mesma, usaremos um exemplo compreensível para aqueles que trabalham com aplicações Web ou Back end. Existe um pool de conexões, no qual várias conexões são compartilhadas com os usuários. Quando trabalhamos com o IndexedDB, é comum termos uma única conexão que será usada pela aplicação. Ao criarmos a chamada para o `getConnection`, ele nos dará a conexão e se fizermos a mesma solicitação novamente, o retorno deverá ser o mesmo. A conexão será compartilhada com toda a aplicação e, por isso, o método `close()` não poderá ser chamado novamente. Lembrando que só o `ConnectionFactory` terá o poder de fechar a conexão.

Levando em consideração estas regras, faremos o design da classe `ConnectionFactory`.