

Atividade de Programação Guiada

PG2 - Web Services

Andy Ruiz Garramones
Gabriel Stepien
Cassiano Stradolini
Guilherme

A aplicação foi feita seguindo o passo a passo da especificação que se encontra no Moodle. No entanto, algumas modificações extras foram feitas para ficar mais flexível testar algumas coisas e para poder inferir alguns detalhes.

O cliente executa e pergunta para o usuário, via input, se deseja somar ou multiplicar. Logo após, é esperada a entrada dos 2 inteiros a serem operados.

Outro ponto modificado foi a adição de um *sleep* no método de *somarInt*.

Leia o README.txt para saber como rodar a aplicação de forma mais simples e rápida.

A seguir, screenshots da aplicação executando:

```
→ Jersey-RESTful-Webservice git:(main) make run-server
java -cp ".:jersey-bundle-1.19.jar" webservice.Servidor
http://localhost:9000/application.wadl
http://127.0.0.1:9000/application.wadl
out 18, 2020 2:46:31 PM com.sun.jersey.api.core.PackagesResourceConfig init
INFORMAÇÕES: Scanning for root resource and provider classes in the packages:
webservice
out 18, 2020 2:46:31 PM com.sun.jersey.api.core.ScanningResourceConfig logClasses
INFORMAÇÕES: Root resource classes found:
class webservice.CalculadoraRest
out 18, 2020 2:46:31 PM com.sun.jersey.api.core.ScanningResourceConfig init
INFORMAÇÕES: No provider classes found.
out 18, 2020 2:46:31 PM com.sun.jersey.server.impl.application.WebApplicationImpl _initiate
INFORMAÇÕES: Initiating Jersey application, version 'Jersey: 1.19 02/11/2015 05:39 AM'
```

Server executando

```
→ Jersey-RESTful-Webservice git:(main) make run-client
java -cp ".:jersey-bundle-1.19.jar" webservice.Cliente
Digite 1 para somar ou 2 para multiplicar
1
Digite primeiro operando
4
Digite segundo operando
6
Resposta da adicao: {"erro":"","operador":"+","operando1":"4","operando2":"6","resultado":"10"}
```

Cliente após executar request de soma

```
→ Jersey-RESTful-Webservice git:(main) make run-client
java -cp ".:jersey-bundle-1.19.jar" webservice.Cliente
Digite 1 para somar ou 2 para multiplicar
2
Digite primeiro operando
4
Digite segundo operando
6
Resposta da multiplicacao: {"erro":"","operador":"*","operando1":"4","operando2":"6","resultado":"24"}
```

Cliente após executar request de multiplicação

O motivo pelo qual se adicionou o sleep no método de somarInt foi para verificar se servidor executa simultaneamente invocações concorrente feitas por múltiplos clientes. O que foi confirmado como verdadeiro: a cada chamada de um cliente se abre uma execução no servidor.

Isso foi possível de confirmar pois foram disparados 2 clientes, 1 somando primeiro e logo após 1 multiplicando.

O cliente que pediu multiplicação recebeu a resposta em seguida, já o da soma recebeu após o tempo do sleep. Isso nos indicou que cada invocação rest abre uma execução no web server.

```
Jersey-RESTful-Webservice git:(main) make run-client
java -cp ".:jersey-bundle-1.19.jar" webservice.Cliente
Digite 1 para somar ou 2 para multiplicar
1
Digite primeiro operando
10
Digite segundo operando
10
[ ]
henios: Ajuda A última edição foi há alguns segundos
```

Cliente soma está com cursor piscando enquanto aguarda

```
Jersey-RESTful-Webservice git:(main) make run-client
java -cp ".:jersey-bundle-1.19.jar" webservice.Cliente
Digite 1 para somar ou 2 para multiplicar
1
Digite primeiro operando
10
Digite segundo operando
10
[ ]
henios: Ajuda A última edição foi há alguns segundos
```

```
Jersey-RESTful-Webservice git:(main) make run-client
java -cp ".:jersey-bundle-1.19.jar" webservice.Cliente
Digite 1 para somar ou 2 para multiplicar
2
Digite primeiro operando
10
Digite segundo operando
10
Resposta da multiplicacao: {"erro":"","operador":"*","operando1":"10","operando2":"10","resultado":"100"}
Jersey-RESTful-Webservice git:(main) [ ]
```

Cliente multiplicação pede pela operação e recebe resposta enquanto cliente soma ainda aguarda

```
Jersey-RESTful-Webservice git:(main) make run-client
java -cp ".:jersey-bundle-1.19.jar" webservice.Cliente
Digite 1 para somar ou 2 para multiplicar
1
Digite primeiro operando
10
Digite segundo operando
10
Resposta da adicao: {"erro":"","operador":"+","operando1":"10","operando2":"10","resultado":"20"}
Jersey-RESTful-Webservice git:(main) [ ]
```

```
Jersey-RESTful-Webservice git:(main) make run-client
java -cp ".:jersey-bundle-1.19.jar" webservice.Cliente
Digite 1 para somar ou 2 para multiplicar
2
Digite primeiro operando
10
Digite segundo operando
10
Resposta da multiplicacao: {"erro":"","operador":"*","operando1":"10","operando2":"10","resultado":"100"}
Jersey-RESTful-Webservice git:(main) [ ]
```

Finalmente Cliente soma recebe resposta pedida antes do Cliente soma

Com isso percebemos que um fluxo fez sleep e outro processou a mult.