

Teste para vaga de programador

Fase 2

Douglas Alves dos Reis

Data 01/11/2016

1. Proposta de Avaliação

Star Trek Rescue

A Frota Estelar recebeu um pedido de resgate de uma tripulação perdida em um planeta desabitado.

Porém, devido a nave ter sido muito danificada na queda, a tripulação de 3 pesquisadores se separou e somente conseguiu enviar parte da informação sobre a sua localização.

Com base nessas informações, a Frota Estelar conseguiu identificar que os tripulantes estavam separados aleatoriamente em 3 pontos de uma planície de 10x10 ao Norte do planeta.

A USS Enterprise tem que marcar a posição dos usuários lançando sinalizadores nas possíveis posições onde os tripulantes estejam, para que as equipes de busca os encontrem.

Você, como parte da equipe de engenharia da USS Enterprise, tem que desenvolver esse sistema que, mapeando a planície em formato de malha, dispare sinalizadores para pontos da planície (usando como referência pontos dos eixos x, y), um sinalizador indicando a posição. Os sinalizadores devem ser lançados um de cada vez, aguardando o retorno do anterior lançado.

O sinalizador ao chegar no ponto informado, deve informar se localizou algum tripulante (Usando o X).

Caso não tenha encontrado, ele rastreará as áreas adjacentes ao ponto que ele chegou, o alcance é de uma área, informando se em alguma delas existe um tripulante (marcando com um !).

Observe que há uma imprecisão no sinalizador, então ele marca as áreas adjacentes em comum aos dois pontos, sinalizador e tripulante.

Se não encontrar nada naquele ponto e nas respectivas áreas adjacentes marque com * para mapear a malha com mais precisão.

O sistema deverá prover uma API REST que permita ser acessada de qualquer sede da Frota Estelar via URLs. Dessa forma, as equipes de resgate conseguirão acionar o serviço de qualquer lugar e terão maiores recursos, uma vez que usaremos toda a infraestrutura do QG da Frota Estelar.

2. Desenvolvimento

Para o desenvolvimento da proposta, foi utilizado Java 7 com abordagem TDD e as seguintes tecnologias:

- **JUnit** para os testes
- **Maven** para gerenciamento de dependências
- **Log4j** para log do serviço Rest
- **Swagger** para descrição do serviço

3. O Serviço

O serviço disponibilizará os seguintes métodos:

POST /sinalizador

Consumes

application/json; charset=UTF-8

Exemplo:

```
{
  "x":3,
  "y":3
}
```

Produces

application/json; charset=UTF-8

Descrição

Dadas as coordenadas de um sinalizador lançado na matriz da planície, o serviço deve atualizar a mesma com o estado decorrente da proximidade (ou não) de um tripulante do ponto atingido.

Resposta

- Mensagem de sucesso
"Sinalizador lançado com sucesso"
- Mensagem de falha
"Coordenadas de lançamento são inválidas"

GET /planicie

Produces

application/json; charset=UTF-8

Descrição

Disponibiliza a planície em formato texto

Resposta

A própria planície em formato texto



swagger.yaml

4. Testes do Serviço

Para os testes do serviço, foi utilizado o SoapUI conforme imagens a seguir:

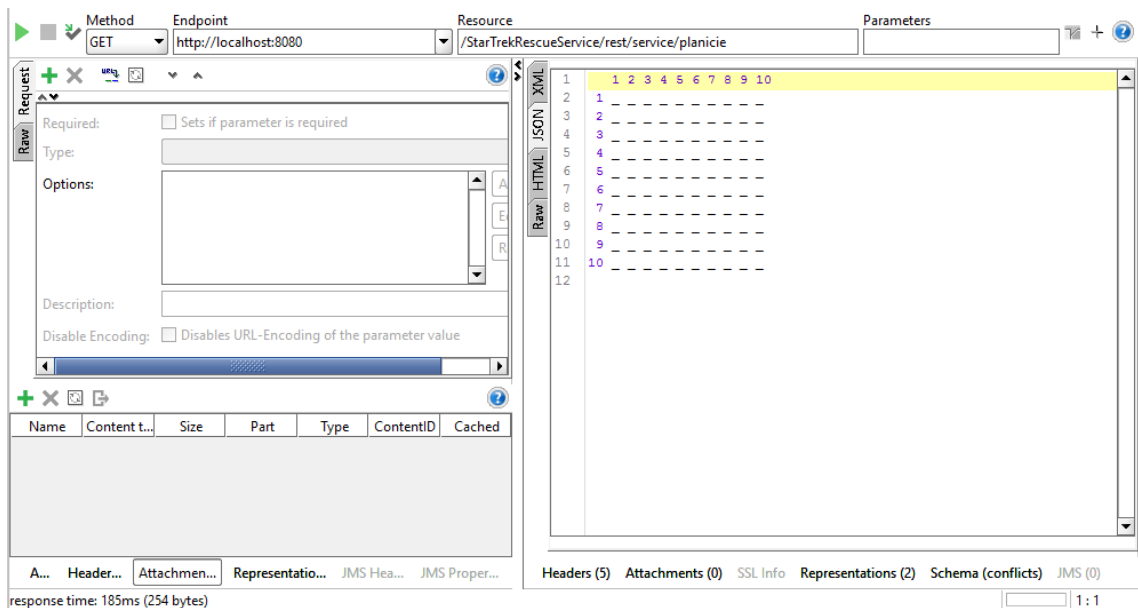


Figura 1: Planície inicial

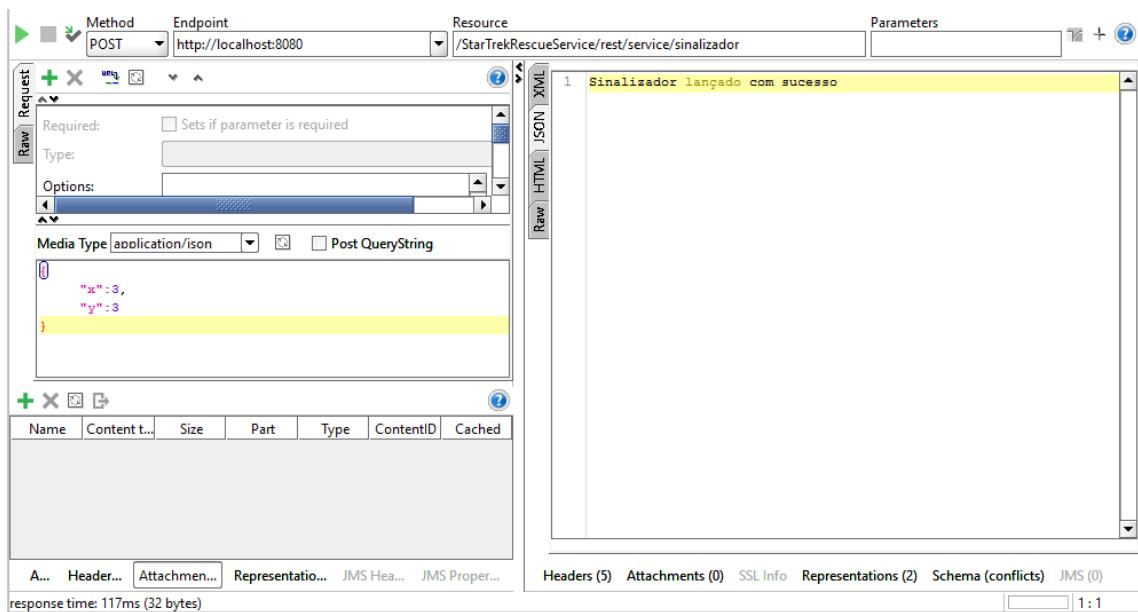


Figura 2: Sinalizador lançado na posição (3;3)

Method

GET

Endpoint

http://localhost:8080

Resource

/StarTrekRescueService/rest/service/planicie

Parameters

Raw

Request

Required:

☐ Sets if parameter is required

Type:

Options:

Description:

Disable Encoding:

☐ Disables URL-Encoding of the parameter value

+

×

📄

Name	Content t...	Size	Part	Type	ContentID	Cached

Raw

JSON

XML

HTML

1

2

3

4

5

6

7

8

9

10

11

12

1

2

3

4

5

6

7

8

9

10

A...

Header...

Attachmen...

Representatio...

JMS Hea...

JMS Proper...

response time: 7ms (254 bytes)

1:1

Figura 3: Resultado do lançamento