

Candidato

Rafael Pablo Sanzana Batista

Data

06/04/2017

Apresentação:

Na MAIS desenvolvemos nossas aplicações usando o framework Spring, sua documentação pode ser encontrada no site <https://spring.io/docs>. Com base no Spring criamos essa tarefa de programação para medir e avaliar nossos candidatos.

Nessa tarefa será medida a capacidade do candidato de seguir um roteiro e suas capacidades de desenvolvimento, de transpor barreiras para atingir um objetivo e sua pró-atividade. Serão avaliadas as habilidades do candidato de codificação e uso de algumas ferramentas.

O projeto Spring possui inúmeras bibliotecas disponíveis, mas isso não é um limitante, caso ache necessário é permitido o uso de qualquer biblioteca. Encorajamos o uso de comentário, mas isso não é um requisito da tarefa.

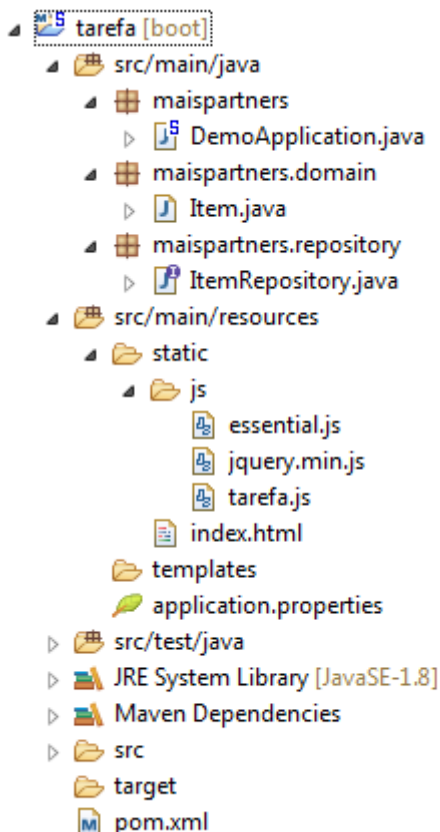
Esse tutorial possui três partes conclua até onde conseguir no prazo estipulado. Caso tenha alguma dificuldade entre em contato conosco. Boa sorte e bom trabalho, esperamos ter você em nossa equipe.

Parte 1: Contextualização.

Dentro do arquivo zip que enviamos existe uma pasta “Tarefa” e esse arquivo pdf contendo as orientações. A pasta “Tarefa” tem um tamanho por volta de 116KB. Caso esteja muito diferente deste valor entre em contato conosco.

Para podermos continuar com a tarefa é necessário que o candidato possua instalado em seu computador java 8, git, maven usaremos abaixo o Spring Tool Suite como IDE, que pode ser encontrada no link <https://spring.io/tools>, aconselhamos o seu uso ou uso do eclipse, mas o candidato tem a liberdade para usar qualquer IDE que deseje ou pode optar por não usar nenhuma.

Primeiro passo dessa parte será abrir o projeto no STS, o projeto deverá ter uma estrutura parecida com a seguinte, podendo variar com a forma de visualização do projeto escolhida.



É aconselhável olhar o código para entender um pouco o que está sendo feito. Agora o próximo passo é rodar a aplicação. Ao rodar você será capaz de ver a seguinte tela:

Item

Nome:

Salvar

Tabela de Itens

Listar

Id	Name	
----	------	--

Objeto

É uma tela bem simples e precisamos torná-la funcional, isso ocorrerá na próxima etapa. Agora é preciso que você crie um repositório público no github e coloque o projeto lá. Quando fizer isso nos passe o link para o projeto por email.

Parte 2: Nosso código precisa de algumas mudanças. (Lembre de comitar cada passo feito!)

Precisamos fazer o botão salvar item e listar item funcionar no index.html.

- 1- Edite o index.html para que os botões chamem suas respectivas funções encontradas no arquivo tarefa.js.
- 2- Edite o arquivo tarefa.js para que ele faça o que esperamos.

Dica: O Spring Data Rest cria as urls para você saiba usá-las.

2.1- Função `saveltem()` – para salvar o item é necessário fazer um POST na url correta com o json equivalente ao modelo que temos. Crie dentro da função um ajax para que isso ocorra.

2.2- Função `listItem()` – para pegarmos a lista de itens é necessário fazer um GET na url correta. Crie dentro da função um ajax para que isso ocorra. Além de simplesmente pegarmos queremos preencher a tabela com esses dados, para isso passe a lista de itens para função `renderList()`, que já implementamos.

Exemplo de uso:

```
...  
renderList(lista a ser renderizada);  
...
```

Dica: Use a função `console.log()` no que é retornado pelo rest para enviar a lista corretamente.

Exemplo da lista a ser passada para função:

```
▼ Array[3] ⓘ  
  ▼ 0: Object  
    ► _links: Object  
      name: "teste"  
    ► __proto__: Object  
  ▼ 1: Object  
    ► _links: Object  
      name: "teste2"  
    ► __proto__: Object  
  ▼ 2: Object  
    ► _links: Object  
      name: "teste3"  
    ► __proto__: Object  
    length: 3  
    ► __proto__: Array[0]
```

- 3- Agora falta a função `deleteItem` – ela recebe o id do item. Para fazê-la funcionar basta fazer uma requisição do tipo DELETE na url correta.
- 4- Agora adicione na entidade `Item` um campo numérico com o nome “nmt” e adicione um input no html e javascript para salvá-lo.
- 5- Faça com que a função para listar os itens seja chamada assim que um item novo for criado.
- 6- Faça com que a função para listar os itens seja chamada assim que a página carregar.

Bonus: Edite o `essential.js` e `index.html` para listar também a coluna “nmt”.

Parte 3: Entidade Novoltem

1- Crie uma entidade chamada Novoltem.

Necessariamente ela precisa conter:

Um campo texto chamado “name”

Dois campos de texto qualquer

Um campo de inteiro que só aceite valores de 1 a 10

Um campo chamado “createdAt” com a data de criação

Um campo double

Faça ter um relacionamento de um para um com a entidade Item

2- Crie um rest find by name e find by id. **Dica:** *Spring Data Rest facilita muito isso.*

3- Crie um controlador rest, no pacote maispartners.controllers, com endereço de acesso “/novorest” com 2 métodos GET e POST; **Dica:** Use anotação `@RestController` para facilitar.

3.1- Metodo GET - retorna a lista de Novoltem existente;

3.2- Método POST - recebe o id de uma entidade Novoltem criada e um número inteiro p (qualquer), encontra o Novoltem pelo id, altera seu campo inteiro com a seguinte conta:

Se $(p + \text{inteiro da entidade}) \leq 10$ e ≥ 1 mudar o inteiro para o resultado da soma, se não salvar inteiro como 3;

Retorna um json no formato: (Use um objeto json para isso.) **Dica:** Use a biblioteca `org.json`

```
{
  "item": "id do item",
  "inteiro": "inteiro que foi salvo",
  "data": "hora do retorno"
}
```

4- Crie no index.html botões que acessem esses métodos e printem no console do browser.

***Dica:** Se baseie na parte 1 para fazer.*

Bonus: Crie um crud para essa entidade no index.html.