AVALIAÇÃO 2 DE ELETIVA WEB – PARTE 2

Não se esqueça de colocar o seu RA e nome na pasta do projeto Netbeans resultante da avaliação.

Uma abordagem muito comum em sistemas modernos é o retorno de respostas de requisições a um servidor no formato JavaScript Object Notation (JSON). Suponha, por exemplo, que se deseja retornar um objeto JSON referente a um cliente. No sistema em questão, um Cliente herda de PessoaFisica que, por sua vez, está associada a um ou mais RGs. Cada RG possui o seu estadoExpeditor.



A seguir, é apresentado o objeto de Cliente e sua respectiva representação em formato JSON:

Estado estadoDoRG1 = new Estado();

estadoDoRG1.setSigla("SP");

estadoDoRG1.setNome("Sao Paulo");

RG rgDoCliente1 = new RG();

rgDoCliente1.setNumero("123.456.789");

rgDoCliente1.setOrgaoExpeditor("SSP");

rgDoCliente1.setEstadoExpeditor(estadoDoRG1);

Estado estadoDoRG2 = new Estado();

estadoDoRG2.setSigla("MG");

estadoDoRG2.setNome("Minas Gerais");

RG rgDoCliente2 = new RG();

rgDoCliente2.setNumero("123.456.780");

rgDoCliente2.setOrgaoExpeditor("SSP");

rgDoCliente2.setEstadoExpeditor(estadoDoRG2);

Cliente cli = new Cliente();

cli.setNome("Joao da Silva");

cli.setNumeroCliente("12345");

cli.setClienteDesde(new Date());

cli.setRg(new RG[] {rgDoCliente1, rgDoCliente2});

{

"nome": "Joao da Silva",

"clienteDesde": "29/11/2018",

"numeroCliente": "12345",

"rg": [{

"numero": "123.456.789",

"orgaoExpeditor": "SSP",

"estadoExpeditor": {

"sigla": "SP",

"nome": "Sao Paulo"

}

},

{

"numero": "123.456.780",

"orgaoExpeditor": "SSP",

"estadoExpeditor": {

"sigla": "MG",

"nome": "Minas Gerais"

}

}

]

}

Os elementos mais importantes desta notação são:

* Um objeto é envolvido por chaves {}
* Uma array/lista/conjunto/... é envolvida por colchetes []
* Os nomes dos atributos geralmente aparecem entre aspas
* Os valores dos atributos aparecem entre aspas para textos, fora de aspas para números e booleanos (true, false). Ainda, o valor de um atributo pode ser null (sem aspas)

Embora um desenvolvedor Java possa escrever uma Servlet ou serviço qualquer que imprima uma String JSON como saída (ver exemplo em projeto NetBeans anexo), o ideal é que isso seja feito por um *framework*. Desta forma, seu papel é criar um framework, usando metaprogramação e anotações, que converta um objeto qualquer em uma representação JSON. Não se preocupe com a formatação de caracteres especiais. A seguir, você verá o uso esperado do *framework*.

Jeyzon jeyzon = new Jeyzon();

jeyzon.toString(cli); // Este método deve retornar o JSON mencionado no início da prova para cliente.

O framework deve ter duas anotações: uma que permite especificar que um atributo não deve fazer parte do resultado JSON: @IgnoreJeyzon; e outra que permite especificar o formato de um campo de data: @Formate(padrao = “dd/MM/yyyy”).

Como exemplo de uso destas anotações, caso as classes Cliente e Pessoa tivessem sido definidas da forma descrita a seguir, o JSON também apresentado a seguir seria gerado.

// ...

public class Pessoa {

private String nome;

@IgnoreJeyzon

private RG[] rg;

// ...

}

public class Cliente extends Pessoa {

private String numeroCliente;

@Formate(padrao = “yyyy-MM”)

private Date clienteDesde;

// ...

}

{

"nome": "Joao da Silva",

"clienteDesde": "2018-11",

"numeroCliente": "12345"

}