



Instituto Superior de Engenharia de Lisboa

ENGENHARIA INFORMÁTICA E DE COMPUTADORES

Ambientes Virtuais de Execução

Série 1

Autores:

35392, João Rodrigues, LI41N

35466, Nick Laert, LI41N

LISBOA, 31 DE MARÇO DE 2016

ANO LECTIVO 2015/2016

Desenvolvimento

Nesta série de exercícios foi-nos proposto desenvolver uma biblioteca capaz de criar a representação JSON de um objecto passado por parâmetro através de reflexão. De modo a simplificar o código optamos por criar um método *Route* que define qual o *Type* do objecto recebido e chama o método auxiliar responsável por criar o JSON para o mesmo. Este método é também onde está presente, embora não de forma óbvia, a recursividade da solução desenvolvida, visto que os métodos auxiliares chamados, chamam novamente o método *Route* caso necessário.

Outro dos objectivos da série de exercícios era que permitisse escolher o critério de seleção dos membros que seriam utilizados para gerar a representação JSON, sendo possível escolher, campos, propriedades ou métodos *non-void* e sem parâmetros. Para tal, acrescentamos um parâmetro que recebe um *Enum* criado na classe *Jsonfier* para identificar o critério escolhido.

Os métodos auxiliares falados anteriormente são:

- *GetPrimitiveValues* – responsável por devolver a representação apenas do valor do objecto, sendo que é chamado para tipos primitivos ou *string*.
- *GetArrayValues* – responsável por devolver a representação de um *array*, chamando para isso o método *Route* de modo a que seja possível representar um *array* de tipos primitivos ou outros tipos.
- *GetEnumNames* – responsável por devolver a representação em JSON de um *Enum*.
- *GetPropertyValues* – responsável por devolver a representação das propriedades do objecto recebido, com o nome da propriedade e o valor, sendo que no caso do valor chama novamente o método *Route*.
- *GetFieldsValues* – muito similar ao método anterior, mas ao invés de devolver a representação das propriedades, devolve a representação para os campos.