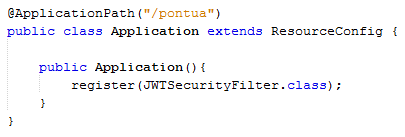
**SERVIDOR PONTUA**

**Classe Application**

****

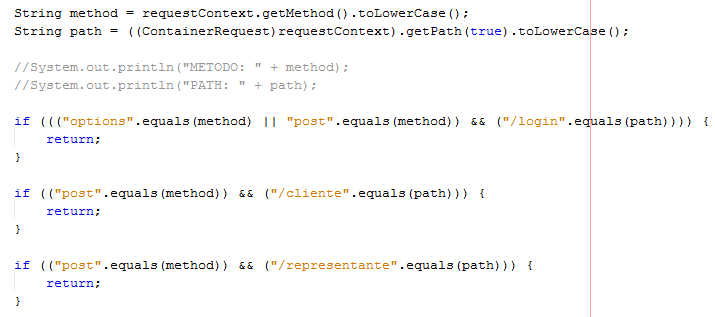
-Registra a classe “JWTSecurityFilter”;

-Todas as classes REST estendem da classe Application usando “/pontua” como path base;

**Classe JWTSecurityFilter**

-Todas as classes do REST passam por essa classe antes;

-Funciona como um filtro;



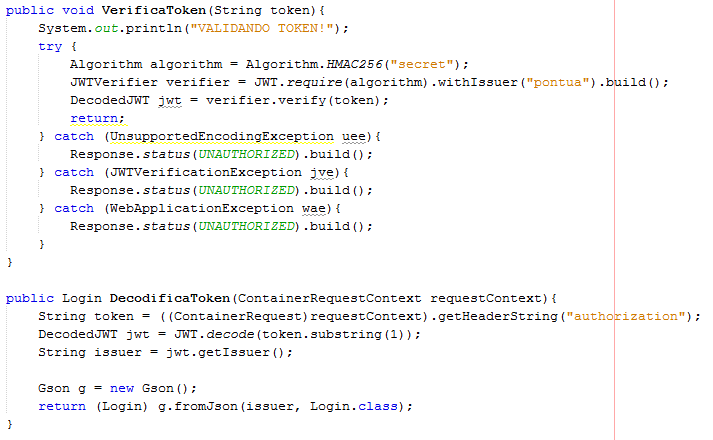
No código acima pega o método e o path chamado, e abaixo faz as verificações.

Quando é dado o “return”, significa que está autorizando a continuação da requisição.



No código acima, no primeiro if, faz a verificação se o token esperado é nulo. No segundo if verifica se o token é de um representante, para que possa acessar a uri (/representante/pontos). No terceiro if verifica se o token é de um cliente, para que possa acessar a uri (/cliente/pontos).

Caso não entre em nenhuma condição ele chama diretamente a função “VerificaToken(strToken)”. Lembrando que essa função também é chama das duas condições que antecedem o “else”.



Na função “VerificaToken”, o método recebe o token como parâmetro e verifica se o token foi gerado pelo servidor e se está autorizado.

Na função “DecodificaToken”, o método pega o “authorization” referente a quem solicitou a chamada do método e decodifica o token em um Json, inserindo-o em uma classe de atributos.

**Classe LoginWS**

-Efetua o login de Representantes e de Clientes

-Path (/pontua/login)

-Verbo (POST)



-Recebe um Json com “email” e “senha”;

-Faz uma consulta do email e senha na base de dados;

-Caso seja encontrado retorna um token gerado pelo servidor;

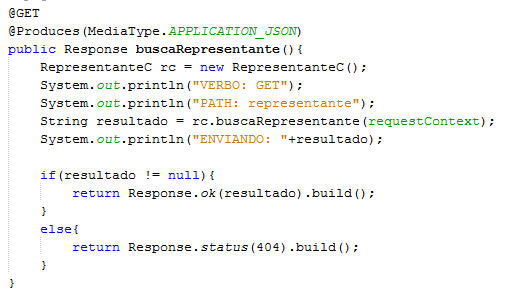
-Caso não seja encontrado retorna o status 401.

**Classe RepresentanteWS**

-Efetua as operações de Representante

-Path (/pontua/representante)

-Verbo (GET)

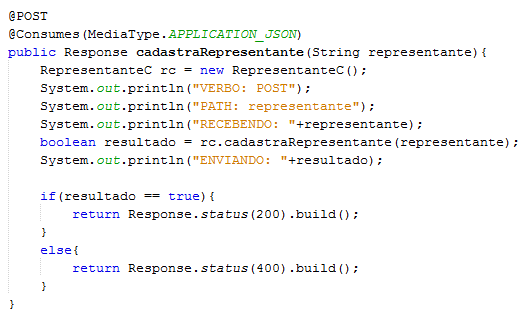


-Busca as informações do representante solicitante baseado no seu token;

-Se o representante for encontrado retorna suas informações;

-Caso ocorra algum erro retorna o código 400.

-VERBO (POST)

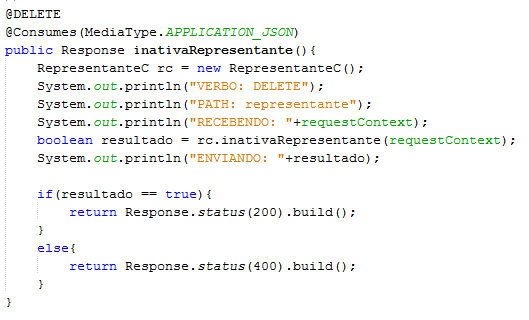


-Cadastra um novo representante através do Json recebido com os dados a serem cadastrados;

-Caso seja cadastrado com sucesso, retorna o código 200;

-Caso ocorra algum erro, retorna o código 400.

-VERBO (DELETE)

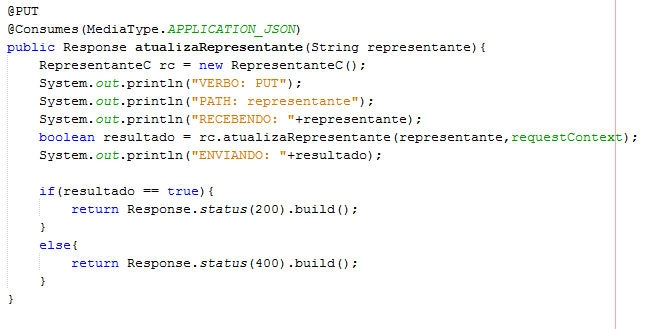


-O representante auto inativa-se, baseado nas informações do seu token;

-Caso a inativação seja realizada com sucesso retorna o código 200;

-Caso ocorra algum erro retorna o código 400.

-VERBO (PUT)



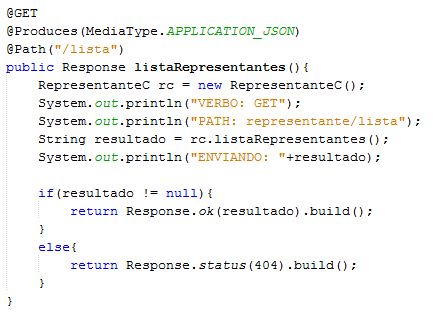
-Recebe os novos dados do representante a ser alterado;

-Se a alteração dos dados ocorrer com sucesso retorna 200;

-Caso ocorra algum erro retorna o código 400.

-Path (/pontua/representante/lista)

-VERBO (GET)

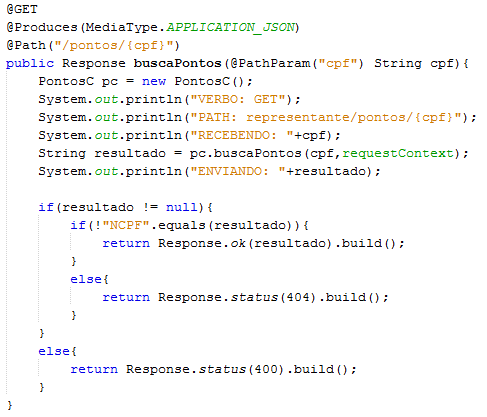


-Retorna todos representantes ativos;

-Caso o Representante não seja encontrado retorna o código 404.

-Path (/pontua/representante/pontos)

-VERBO (GET)



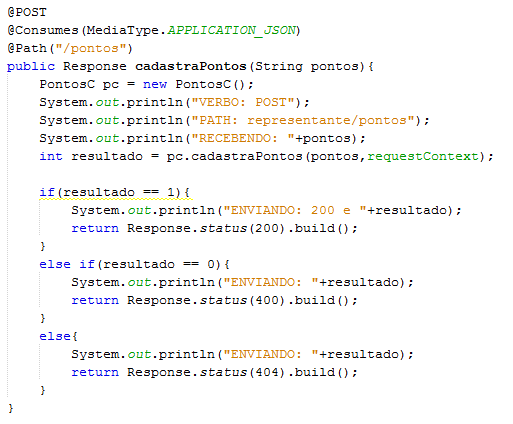
-Recebe o cpf como parâmetro;

-Retorna os pontos do cliente associados ao representante que solicita, baseado no cpf recebido por parâmetro e no token do representante que solicita;

-Caso o cpf não seja encontrado retorna o status 404;

-Caso ocorra outro erro retorna o status 400.

-VERBO (POST)



-Recebe como parâmetro um Json contendo cpf e id\_promocao;

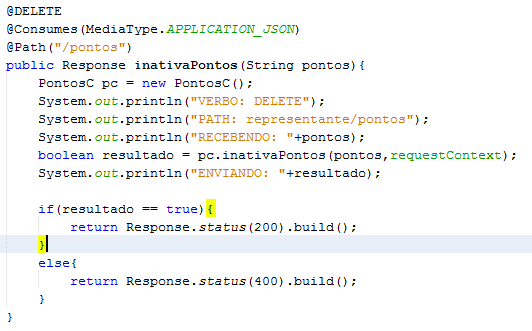
-Cadastra pontos da promoção referente a id\_promocao, para o cliente referente ao cpf;

-Caso seja cadastrado com sucesso retorna o status 200;

-Caso a promoção não seja encontrada retorna o status 404;

-Caso ocorra outro erro retorna o status 400.

-VERBO (DELETE)



-Recebe como parâmetro um Json contendo cpf;

-Inativa a última atribuição de pontos para o cliente baseado no cpf e no representante que solicita, através do seu token;

-Caso a operação seja realizada com sucesso retorna 200;

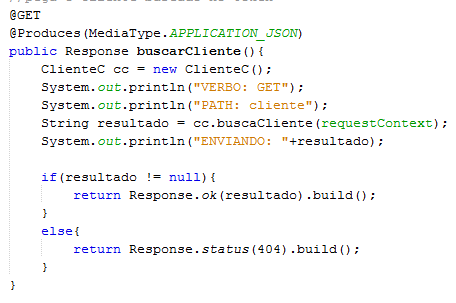
-Caso ocorra algum erro retorna 400.

**Classe ClienteWS**

-Efetua as operações de Cliente

-Path (/pontua/cliente)

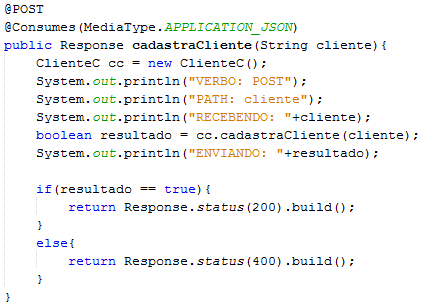
-VERBO (GET)



-Retorna dados do cliente associado ao token de quem requisita;

-Caso não seja encontrado retorna o statu 404.

-VERBO (POST)

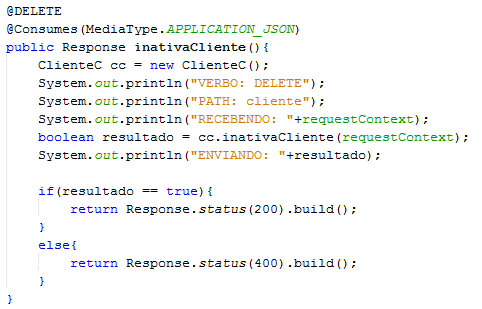


-Cadastra um novo cliente através do Json recebido com os dados a serem cadastrados;

-Caso seja cadastrado com sucesso, retorna o código 200;

-Caso ocorra algum erro, retorna o código 400.

-VERBO (DELETE)

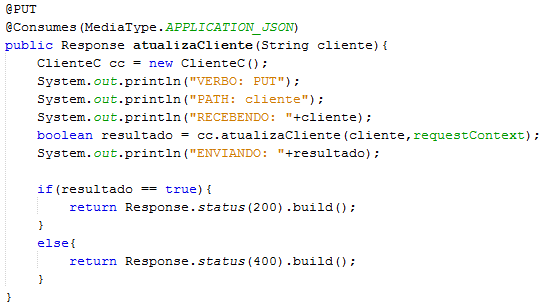


-O cliente auto inativa-se, baseado nas informações do seu token;

-Caso a inativação seja realizada com sucesso retorna o código 200;

-Caso ocorra algum erro retorna o código 400.

-VERBO (PUT)



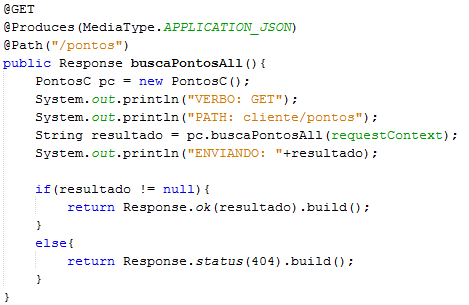
-Recebe os novos dados do representante a ser alterado (Json);

-Se a alteração dos dados ocorrer com sucesso retorna 200;

-Caso ocorra algum erro retorna o código 400.

-Path (/pontua/cliente/pontos)

-VERBO (GET)

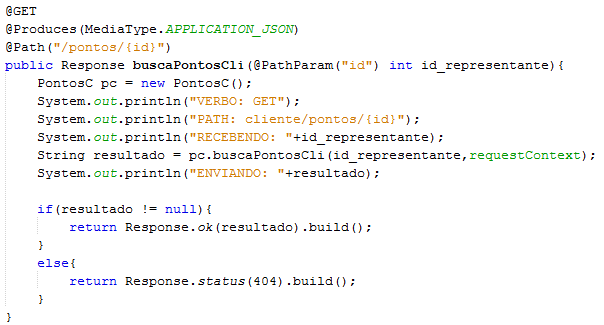


-Retorna todos os pontos associados ao cliente que requisita, através do seu token;

-Caso o cliente não seja encontrado retorna o status 404.

-Path (/pontua/cliente/pontos/{id})

-VERBO (GET)



-Recebe o id do representante como parâmetro;

-Retorna os pontos do cliente que solicita através do seu token, associados ao id do representante recebido por parâmetro;

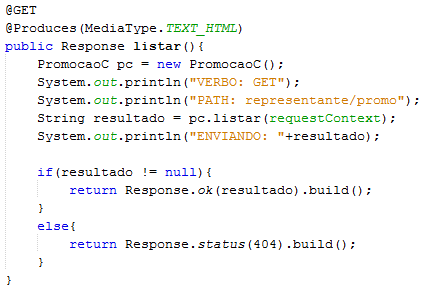
-Caso o id não seja encontrado retorna o status 404;

**Classe PromocaoWS**

-Efetua as operações de Representante

-Path (/pontua/representante/promo)

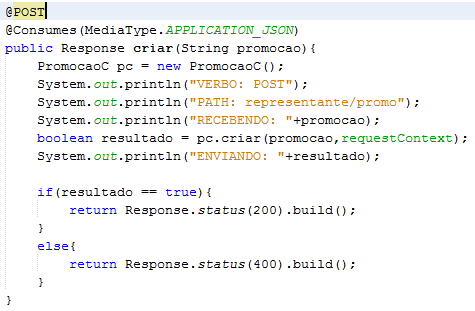
-VERBO (GET)



-Retorna todas as promoções associadas ao representante que solicita através do seu token

-Caso não encontre nenhuma promoção retorna o status 404.

-VERBO (POST)



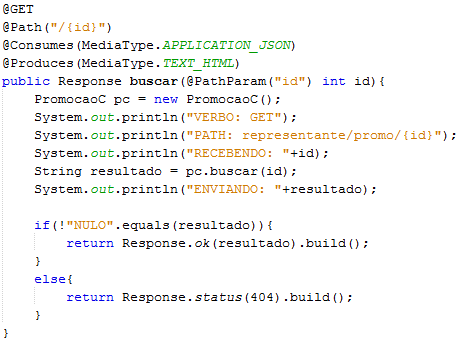
-Recebe um Json com os dados da promoção a serem cadastrados;

-Caso a promoção seja cadastrada com sucesso retorna o status 200;

-Caso ocorra algum erro retorna o status 400.

-Path (/pontua/representante/promo/{id})

-VERBO (GET)

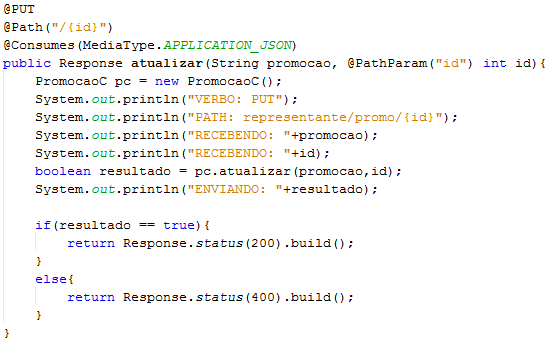


-Recebe o id da promoção como parâmetro;

-Retorna os dados da promoção referentes ao id recebido como parâmetro;

-Caso a promoção referente ao id não seja encontrada retorna o status 404.

-VERBO (PUT)

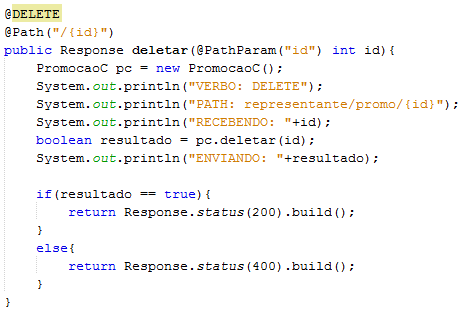


-Recebe o id da promoção a ser alterada, e os novos dados da promoção no formato Json;

-Caso a alteração seja realizada com sucesso retorna o status 200;

-Caso a promoção referente ao id não seja encontrada retorna o status 404.

-VERBO (DELETE)



-Recebe como parâmetro o id da promoção a ser inativada;

-Caso a promoção seja inativada com sucesso retorna o status 200;

-Caso ocorra algum erro retorna o status 400.