```
function servicetask10(attempt, message) {

    incluiTituloProtheusOrdem();

}

function incluiTituloProtheusOrdem() {

    log.info("@@@@@@@@@@@@@@@ INTEGRAÇÃO CONTRATO DE CAMBIO
@@@@@@@@@@@@@@@@4");

    if(hAPI.getCardValue("tipoCliente") == "1"){
        var E1_FLUXO  = "N";
    }else if(hAPI.getCardValue("tipoCliente") == "2"){
        var E1_FLUXO  = "S";
    }

    if(hAPI.getCardValue("rdTipoInclusao") == "ordem"){
        var E1_NUM  = hAPI.getCardValue("referencia");
        var E1_XOPER = "";
    }else if(hAPI.getCardValue("rdTipoInclusao") ==
"fechamentoParcial"){
        var E1_NUM  = hAPI.getCardValue("zoomReferencia");
        var E1_XOPER = hAPI.getCardValue("cdOperadora");
    }

    try {
        var clientService =
fluigAPI.getAuthorizeClientService();

        var data = {
            companyId : '1',
            serviceCode : "rest-protheus",
            endpoint : '/TITREC',
            method : 'POST',
            timeoutService : '100',
            headers : {
                "Accept-Charset" : "UTF-8",
                "Content-Type" : "application/json",
            },
            params : {
                "EMPRESA" : "" +
hAPI.getCardValue("cdEmpresaProtheus"),
                "FILIAL" : "" +
hAPI.getCardValue("cdFilialProtheus"),
                "E1_PREFIXO" : "CAM",
                "E1_NUM" : E1_NUM,
```

```
                    "E1_PARCELA" : "A",
                    "E1_TIPO" : "BOL",
                    "E1_NATUREZ" : "" +
hAPI.getCardValue("txt_cdNatureza"),
                    "E1_CLIENTE" :
hAPI.getCardValue("a1cod"),
                    "E1_LOJA" : hAPI.getCardValue("a1loja"),
                    "E1_EMISSAO" :
hAPI.getCardValue("creationDate"),
                    "E1_VENCTO" :
dataFormatada(hAPI.getCardValue("dataCredito")),
                    "E1_VALOR" :
parseFloat(hAPI.getCardValue("moedaEstrangeira").replaceAll("
\\.","").replaceAll(",", "."),
                    "E1_HIST" :
hAPI.getCardValue("observacoes"),
                    "E1_MOEDA" : new
java.math.BigInteger(hAPI.getCardValue("cdMoeda")),
                    //"E1_MOEDA" : 1,
                    "E1_XNOMSOL" : "",
                    "E1_XEMAILS" : "",
                    "E1_XNOME" : "",
                    "E1_XCPF" : new
java.math.BigInteger(hAPI.getCardValue("zoomcpfCnpj")),
                    "E1_XTPOPER" : ""+
hAPI.getCardValue("tipoOperacao"),
                    "E1_XDTFCH" : "",
                    "E1_XMOEDA" : ""+new
java.math.BigInteger(hAPI.getCardValue("cdMoeda")),
                    //"E1_XMOEDA" : "1",
                    "E1_XMOEDEX" : "",
                    "E1_XCONTRA" : "",
                    "E1_XTXCLI" : parseFloat(0),
                    "E1_TXMOEDA" : parseFloat(1),
                    "E1_XTXSPOT" :parseFloat(0),
                    "E1_XDESPBC" : parseFloat(0),
                    "E1_XIOF" : parseFloat(0),
                    "E1_XVLRTOT" : parseFloat(0),
                    "E1_XSPREAD" : parseFloat(0),
                    "E1_XVLRSPE" : parseInt(0),
                    "E1_XCOMIS" : parseFloat(0),
                    "E1_XOBSERV" : "" +
hAPI.getCardValue("observacoes"),
                    "E1_XOPER" : E1_XOPER
                }
            };
```

```
          //log.dir(data);
          log.info("JSON de integração " +
JSONUtil.toJSON(data));

          var retornoApi =
clientService.invoke(JSONUtil.toJSON(data));
          log.info("retornoApi.getResult() " +
retornoApi.getResult());

          var retornoApiGetResult = "" +
retornoApi.getResult();
          retornoApiGetResult = retornoApiGetResult.replace(
                    /[\u0000-\u001F\u007F-\u009F]/g, '')

          var retornojson = JSON.parse(retornoApiGetResult);

          var statusHttp = retornoApi.getHttpStatusResult();

          // VERIFICA SE O RETORNO ESTA VAZIO
          if (retornoApi.getResult() == null ||
retornoApi.getResult().isEmpty())
               throw java.lang.Exception("Ocorreu uma falha
no retorno da API!");

          if (statusHttp >= "400") {
               throw retornoApi.getResult();
          }

          //hAPI.setCardValue("retornoProtheus___" + indice,
retornojson.Mensagem);

/*        if (retornojson.Mensagem == "OK") {
               hAPI.setCardValue("codigoTitulo___" + indice,
codigoTitulo);
          }*/

          if (retornojson.Mensagem != "OK") {
               throw retornojson["Mensagem Detalhada"];
          }

     } catch (err) {
          //throw err +" - "+ retornoApi.getResult()+" - "+
JSON.stringify(data);
          log.error(err)
          throw err + " - Linha: " + err.lineNumber;
     }
```

```
}

function preencherZerosEsquerda(numero, tamanhoTotal) {
     var numeroString = String(numero);
     while (numeroString.length < tamanhoTotal) {
          numeroString = '0' + numeroString;
     }
     return numeroString;
}

function retornaData() {
     // Buscar a data e hora atual
     var data = new Date();
     var dia = data.getDate();
     var mes = data.getMonth() + 1;
     var ano = data.getFullYear();
     var hora = data.getHours();
     var min = data.getMinutes();
     var seg = data.getSeconds();

     if (dia < 10) {
          dia = "0" + dia;
     }
     if (mes < 10) {
          mes = "0" + mes;
     }
     if (hora < 10) {
          hora = "0" + hora;
     }
     if (min < 10) {
          min = "0" + min;
     }
     if (seg < 10) {
          seg = "0" + seg;
     }

     return dataInclusaoCompleta = dia + '/' + mes + '/' +
ano;
}

function dataFormatada(data) {
     // Expressão regular para verificar o formato dd/mm/yyyy
     var regexDDMMYYYY = /^\d{2}\/\d{2}\/\d{4}$/;

     // Expressão regular para verificar o formato yyyy-mm-dd
     var regexYYYYMMDD = /^\d{4}-\d{2}-\d{2}$/;
```

```
    if (regexYYYYMMDD.test(data)) {
        // Se a data estiver no formato yyyy-mm-dd, fazemos
a conversão para yyyy-mm-dd
        var partes = data.split('-');
        var dia = partes[2];
        var mes = partes[1];
        var ano = partes[0];
        return dia + '/' + mes + '/' + ano;
    } else if (regexDDMMYYYY.test(data)) {
        // Se a data estiver no formato dd/mm/yyyy, não é
necessário converter
        return data;
    } else {
        // Se a data não estiver em nenhum dos formatos
suportados, retorna null ou a própria data
        // Isso pode ser ajustado de acordo com a
necessidade do seu caso.
        return null;
    }
}
```