# SAP Gateway – GET_STREAM marotão que cospe uma DANFE

*20/03/2017 10:00*

FALA, BANDO DE MORTO VIVO! Todo mundo morto aí? Ou vivo? Quem sabe, faz tanto tempo que eu não escrevo aqui...

Eis um *post*zinho de dica pra dar uma movimentada no *blog*, que, com exceção dos *podcasts* (que estão demorando pra sair também, diga-se de passagem), ficou pelas plantinhas secas rodando no deserto...

Sim, são plantas secas e não feno. *Tumbleweeds*. Porém, divirjo. Vamos ao tema.

Depois de ter passado por alguns projetos de UI5/Gateway aqui em nossa pátria amada, é bem possível que você, assim como eu, já tenha encarado o tema ~~de terror~~ que mais abala o mundo dos consultores SAP: NF-e. E quando falamos de NF-e, normalmente o usuário não quer nem saber de chegar perto do XML, e sim de ver a sua representação gráfica, a infame **DANFE**. Como faremos, então, para o nosso serviço Gateway cuspir aquele PDF bonito, bem diagramado, minimalista, com todas as informações corretas, em um tempo aceitável para o usuário?

A única coisa que você precisa no serviço pra iniciar a impressão, neste caso, é o número da nota no SAP (J_1BNFDOC-DOCNUM). Há diversas maneiras de chegar nele, então veja você mesmo(a) como buscá-lo, dependendo do ambiente.

Uma vez tendo o DOCNUM, a ideia é submeter um job do relatório RSNAST00, que realiza a impressão do DANFE. O caso normal, nesse tipo de aplicação, é gerar a DANFE em modo de reimpressão; não tenho certeza se funciona quando a DANFE ainda não foi impressa pela primeira vez, mas isso é algo a ser testado. O *spool* gerado é a própria DANFE. Uma vez tendo o número do *spool*, podemos gerar um PDF usando a função CONVERT_OTFSPOOLJOB_2_PDF.

A pegadinha aqui é que o *spool* não é gerado imediatamente quando submetemos o *job*; logo, temos que ficar procurando o *id* do *spool* na tabela TBTCP. Mas essa busca não pode durar para sempre, porque estamos falando de um serviço Gateway. O serviço pode dar *timeout* e o usuário pode ficar na mão se a geração do *spool* der algum problema, ou mesmo se demorar um pouco mais que seja do que o parâmetro de *timeout* do servidor. Então, marcamos o tempo do início da busca e vamos comparando com o tempo atual. Se passar do tempo de *timeout* definido, retornamos um erro.

Na hora de voltar o PDF pelo método /IWBEP/IF_MGW_APPL_SRV_RUNTIME~GET_STREAM, basta colocar no *header* da resposta o parâmetro content-disposition com o valor outline; filename="{ nome_do_arquivo }.pdf" pra fazer o arquivo abrir como *download* numa aba nova.

E só lembrando, é claro, que pra usar o GET_STREAM a entidade precisa estar marcada com o *flag* **Media** lá na SEGW.

É hora do código:

```
METHOD /iwbep/if_mgw_appl_srv_runtime~get_stream.

  DATA: lt_source_keys TYPE /iwbep/t_mgw_tech_pairs,
        lv_docnum      TYPE j_1bdocnum,
        lv_pdf         TYPE xstring,
        ls_header      TYPE ihttpnvp,
        ls_stream      TYPE ty_s_media_resource.

  FIELD-SYMBOLS <key> TYPE /iwbep/s_mgw_tech_pair.

  CASE io_tech_request_context->get_entity_set_name( ).
```

```
*   Aqui a entidade que devolve a DANFE chama "DANFE" mesmo, com a entity set
*   chamando "DANFEs"
    WHEN 'DANFEs'.

*     Neste caso estou assumindo que temos como chave da entidade o próprio
*     DOCNUM
      lt_source_keys = io_tech_request_context->get_source_keys( ).
      READ TABLE lt_source_keys ASSIGNING <key> WITH KEY name = 'DOCNUM'.
      IF sy-subrc = 0.
        lv_docnum = <key>-value.
      ENDIF.

      TRY.

*         Um pouco de modularização, pra não encher o GET_STREAM de lógica
          lv_pdf = me->get_danfe( lv_docnum ).

*         Header bonitão que faz a resposta ser interpretada como arquivo PDF,
*         abrindo numa aba nova:
          ls_header-name = 'content-disposition'.
          ls_header-value = 'outline; filename="' && lv_docnum && '.pdf"'.
          set_header( ls_header ).

          ls_stream-value = lv_pdf.
          ls_stream-mime_type = 'application/pdf'.
          copy_data_to_ref(
            EXPORTING
              is_data = ls_stream
            CHANGING
              cr_data = er_stream
          ).

        CATCH.

          RAISE EXCEPTION TYPE /iwbep/cx_mgw_tech_exception
            EXPORTING
*             textid = alguma mensagem de erro aqui
              method = gcs_methods-get_stream.

      ENDTRY.

  ENDCASE.

ENDMETHOD.

**********************************************************************

METHOD get_danfe.

  DATA: lv_jobname            TYPE btcjob,
        lv_jobcount           TYPE btcjobcnt,
        ls_spool_parameters   TYPE pri_params,
        lv_keep_alive_timeout TYPE pfepvalue,
        lv_spool_gen_time     TYPE tzntstmpl,
        lv_timeout            TYPE tzntstmpl,
        lv_begin_time         TYPE tzntstmpl,
        lv_current_time       TYPE tzntstmpl,
        lv_listident          TYPE tbtcp-listident,
        lv_src_spoolid        TYPE rspoid,
        lv_spoolid            TYPE rqident.

* A primeira coisa a se fazer aqui é abrir o job:
  lv_jobname = 'DANFE_STREAM_' && iv_docnum.

  CALL FUNCTION 'JOB_OPEN'
```

```
    EXPORTING
      jobname          = lv_jobname
    IMPORTING
      jobcount         = lv_jobcount
    EXCEPTIONS
      cant_create_job  = 1
      invalid_job_data = 2
      jobname_missing  = 3
      OTHERS           = 4.

  IF sy-subrc <> 0.
*   Retornar uma exception se der erro aqui
*   RAISE EXCEPTION TYPE (...)
  ENDIF.

* Agora, damos um SUBMIT no RSNAST00 para fazer uma reimpressão do DANFE:
  SUBMIT rsnast00 WITH s_kappl = 'NF'
                  WITH s_objky = lv_docnum
                  WITH s_nacha = '1'
                  WITH p_again = abap_true
                  TO SAP-SPOOL
                  SPOOL PARAMETERS ls_spool_parameters
                  WITHOUT SPOOL DYNPRO
                  VIA JOB lv_jobname NUMBER lv_jobcount
                  AND RETURN.

  CALL FUNCTION 'JOB_CLOSE'
    EXPORTING
      jobcount  = lv_jobcount
      jobname   = lv_jobname
      strtimmed = abap_true.

* Com o job submetido para execução, começamos o processamento do loop de
* timeout.

* Primeiro, lemos o parâmetro de timeout padrão do servidor:
  CALL FUNCTION 'RSAN_SYSTEM_PARAMETER_READ'
    EXPORTING
      i_name     = 'icm/keep_alive_timeout'
    IMPORTING
      e_value    = lv_keep_alive_timeout
    EXCEPTIONS
      read_error = 1
      OTHERS     = 2.

  IF sy-subrc = 0.
    lv_timeout = lv_keep_alive_timeout.
  ELSE.
    lv_timeout = 30. "Valor padrão do icm/keep_alive_timeout
  ENDIF.

  GET TIME STAMP FIELD lv_begin_time.

  DO.

    CLEAR: lv_listident, lv_current_time.

*   Já temos o número de spool do job?
    SELECT SINGLE listident
      FROM tbtcp BYPASSING BUFFER
      INTO lv_listident
      WHERE jobname  = lv_jobname
        AND jobcount = lv_jobcount.

    IF lv_listident <> 0.
```

```
*     Sim; hora de sair do loop
      EXIT.

    ELSE.

      GET TIME STAMP FIELD lv_current_time.

*     Fazemos a subtração para ver se já bateu o limite do timeout:
      TRY .

          lv_spool_gen_time = cl_abap_tstmp=>subtract(
              tstmp1 = lv_current_time
              tstmp2 = lv_begin_time
          ).

        CATCH cx_parameter_invalid_range cx_parameter_invalid_type.

*         Deu algum problema na subtração dos tempos, logo é melhor já assumir
*         o tempo de timeout e sair do método
          lv_spool_gen_time = lv_timeout.

      ENDTRY.

      IF lv_spool_gen_time >= lv_timeout.
*       O limite do timeout foi batido e nada do spool gerado...
*       Logo, retornamos uma exception aqui
*       RAISE EXCEPTION TYPE (...)
      ENDIF.

    ENDIF.

  ENDDO.

* Com o número do spool encontrado, basta gerar o PDF:
  lv_src_spoolid = lv_listident.

  CALL FUNCTION 'CONVERT_OTFSPOOLJOB_2_PDF'
    EXPORTING
      src_spoolid              = lv_src_spoolid
      no_dialog                = abap_true
      pdf_destination          = 'X'
    IMPORTING
      bin_file                 = rv_pdf
    EXCEPTIONS
      err_no_otf_spooljob      = 1
      err_no_spooljob          = 2
      err_no_permission        = 3
      err_conv_not_possible    = 4
      err_bad_dstdevice        = 5
      user_cancelled           = 6
      err_spoolerror           = 7
      err_temseerror           = 8
      err_btcjob_open_failed   = 9
      err_btcjob_submit_failed = 10
      err_btcjob_close_failed  = 11
      OTHERS                   = 12.

  IF sy-subrc <> 0.
*   Outra exception se der erro aqui
*   RAISE EXCEPTION TYPE (...)
  ENDIF.

* Como limpeza final, podemos opcionalmente deletar o job e o spool:
  CALL FUNCTION 'BP_JOB_DELETE'
    EXPORTING
      jobcount                 = lv_jobcount
```

```
      jobname                = lv_jobname
      forcedmode             = abap_true
    EXCEPTIONS
      job_is_already_running = 1
      OTHERS                 = 2.

  lv_spoolid = lv_listident.

  CALL FUNCTION 'RSPO_R_RDELETE_SPOOLREQ'
    EXPORTING
      spoolid = lv_spoolid.

ENDMETHOD.
```

Agora, é só virar pro seu(ua) amiguinho(a) desenvolvedor(a) UI5 e dizer “pronto, jamanta, pode chamar essa URL bonita aqui que o teu PDF da DANFE vai vir”. Aí é só fazer um sap.m.URLHelper.redirect e pronto.

**PLOT TWIST:** sap.m.URLHelper.redirect não funciona no Safari. A criança chora e a mãe não vê.

Abraços e até a próxima!

PS.: Obrigado ao grande Custódio, onipresente nos *posts*, pelo *peer review*.

---

## Comentários

> **Custodio** — *20/03/2017 20:24*
>
> Boa Leo, tava com saudades dos posts de voces, achei ate que tinham esquecido como se escreve.
>
> 2 coisinhas rapidas:
>
> 1 – faltou o F na tabela j_1bnFdoc
>
> 2 – Evite usar parametros obsoletos no Gateway, como iv_entity_set_name e it_key_tab. Veja essa post https://blogs.sap.com/2017/02/01/avoid-using-deprecated-sap-gateway-apis-in-your-odata-service-implementation/
>
> Ia tb dar uma cornetada no codigo, mas depois da minha barrigada sobre o anagrama fiquei meio timido, rs
>
> Abraco!
>
> Custodio
>
> PS: aguardando pacientemente o zombiecast mensal 😕

> > **Leo Schmidt** — *21/03/2017 10:48*
> >
> > Olha o Custódio aí!
> >
> > Bom, só posso falar por mim… as coisas estavam bem corridas ultimamente, só agora consegui um tempo pra escrever de novo aqui. Mas realmente, já tá juntando teia de aranha.
> >
> > 1 – Bem observado, corrigi.
> > 2 – Olha aí, nem sabia dos obsoletos. *Duly noted*.
> >
> > E tua solução do problema dos anagramas, no final, foi mais clara do que a minha…

> Agora, com relação ao *cast*, o jeito é lotar o @mauriciooocruz de DM mesmo, pq agora ele é o diferentão do fuso horário.
>
> []s