

□ Documento incompleto, estamos desarrollándolo □

Descarga y Visualización de Documentos/Informes PDF

Es habitual que necesitemos hacerle llegar al usuario de nuestro CGI Néfele un documento PDF, tanto realizando una descarga directa como haciendo una visualización previa y que el usuario después decida lo que quiere hacer con el.

La forma mas fácil es poner un link de descarga del documento en la página generada con nuestro CGI y al hacer click el usuario en dicho enlace proceder ha realizar la descarga.

El problema que le vemos a esta opción es que el documento ha de estar en una carpeta publica de nuestro Dominio, lo cual implica que cualquiera podría acceder a los documentos si sabe como se llaman, o por fuerza bruta.

Nosotros recomendamos no utilizar esta técnica, preferimos que si tenemos almacenados los documentos en nuestro Dominio estén en una carpeta no accesible, por ejemplo en una subcarpeta dentro de cgi-bin o en cualquier otra carpeta a la que pueda acceder nuestro CGI pero no así el navegador directamente.

Si el documento no esta previamente almacenado sino que lo creamos al vuelo cuando el usuario lo solicita, recomendamos igual que en el caso de los almacenados, crearlo en una carpeta no accesible por el navegador.

Para lograr este nivel de seguridad Néfele dispone de varias herramientas que nos facilitan el envío de los documentos desde la carpeta segura al navegador.

- [oCgi:SendFile](#) - Para el envío propiamente de un fichero.
- [nfl_EmbedFile](#) - Para la creación del HTML necesario para presentar un visualizador con el documento
- [nfl_EmbedPDFjs](#) - Para la creación del HTML necesarios para presentar nuestro documento en el visualizador [PDF.js](#)
- [oCgi:SendPage](#) - Para el envío del código HTML generado en las dos funciones "Embed"

Creando al vuelo Documentos/Informes PDF con...

FastReport

Cuando queremos crear ficheros PDF utilizando FastReport tenemos que tener en cuenta que se va ha ejecutar en el servidor, por lo que no podemos permitir que nos pase la ejecución de nuestro CGI por un error o por que abra un dialogo pidiendo algún parámetro o avisándonos de algun mensaje.

Por eso debemos asignarle unas cuantas propiedades a FastReport que lo evite.

```
:lAbortOnErrors := .F. // Evitamos que cancele el proceso por un error
:oEngineOptions:SetSilentMode( .T. ) // No queremos que de ningún mensaje
:SetProperty("PDFExport", "ShowDialog", .F.) // No necesitamos que pregunte nada para exportar
```

La primera propiedad hemos de asignarla **antes** del :Create, las otras dos despues.

```
WITH OBJECT TFastReport():New()
  :lAbortOnErrors := .F. // Evitamos que cancele el proceso por un error
  :Create()
  :oEngineOptions:SetSilentMode( .T. ) // No queremos que de ningún
mensaje
  // Parámetros del PDF
  :SetProperty("PDFExport", "ShowDialog", .F.) // No necesitamos que
pregunte nada para exportar
  :SetProperty("PDFExport", "FileName", "document.pdf")
  :SetProperty("PDFExport", "DefaultPath", cReportPath) // carpeta donde
se guardará el PDF generado
  :SetProperty("PDFExport", "EmbeddedFonts", .t.)
  :SetProperty("PDFExport", "PrintOptimized", .T.)
  // Propiedades del PDF
  :SetProperty("PDFExport", "Title", "Informe de Eliminados por Bookings
" + :oEd_Nombre:Value)
  :SetProperty("PDFExport", "Author", "Néfele Project")
  :SetProperty("PDFExport", "Producer", "Néfele Project")
  :SetProperty("PDFExport", "Creator", "Néfele Project")
  :SetProperty("PDFExport", "Keywords", "Néfele, xBase, Xailer")
  /*
  .
  .
  .
  */
  :DoExport("PDFExport")
  IF :nLastError <> 0
    oCGI:Console( :cLastError )
  ENDIF
  :Free()
END
```

wkhtmltopdf

wPDFReport

Con la Clase [wPDFReport](#) podemos crear un documento PDF elemento a elemento, en el servidor no se crea el PDF sino que se envía al navegador las instrucciones para que lo cree en el lado del cliente por lo que reducimos el tiempo de ejecución de nuestro CGI.

From:
<https://www.nefele.dev/wiki/> - **Nefele Project**

Permanent link:
<https://www.nefele.dev/wiki/tecnicas/pdfviewer?rev=1625089791>

Last update: **30/06/2021 21:49**

