csg-pdf

Author: Patrick Pliessnig
contact/support: google groups -> simple groupware

Introduction

Printing data to pdf is not a standard function in sgs. Csg-fpdf provides an easy way to do exactly this. It is based on the csg framework and fpdf (http://www.fpdf.org/). another source of inspiration is the 'table design with fpdf' from von der Born (http://www.vonderborn.com/extended-tables-with-fpdf.php). the design of csg-fpdf is adaptable to different needs by extending the base class csgPDFForm.

Template structure

The idea of von der Born is to construct a php array of text and formating elements – a multi cell – and print these multicells into the pdf document. Csg-pdf does this but organisizes itself differently. It moves the properties of the multi cells and the pdf document properties into the sgs-database with two different modules. Printing data to a pdf document is then basically merging the multi cells with other data taken from the database (or data generated by a php procedure) and writing the merged information into a fpdf document.

Elements

- sgs module csgFPDFTemplates.xml
  this module contains the property of the pdf document like orientation, paper size, etc
- sgs module csgFPDFMuliCells.xml
  this module contains the formating elements of a multicell plus a default text.
- php class csgPDFForm.php
  this class is the pdf engine. It merges the multicells from module csgFPDFMultiCells with other data and writes them to the pdf document.
- php script export_to_pdf.php
  a simple script to launch the pdf engine with a url.

Installation

1. Install the modules (xml files) like normal sgs modules
2. Copy the class csgPDFForm.php into your csg framework directory
3. Copy fpdf from fpdf.org into a directory according to the fpdf installation instructions.
   I suggest copying it into a subdirectory of <sgs-installation-directory>/ext/ext/ .
4. Copy export_to_pdf.php script to your own csg script directory.

5. adjust the fpdf directory pathnames in the export_to_pdf.php script (variables $fpdf and $fpdf_font) according to your path names.

Usage

To create a pdf template you need to do this:

1. create a new pdf template asset in a csgFPDFTemplates folder
   please consult fpdf for more info on available settings.
   The name of the template must correspond to your own customized pdf class.
   For example your pdf class can be named myPDFForm in a file myPDFForm.php (see below).
2. create your multicell definitions in a csgFPDFMultiCells folder
   please consult fpdf for more info on available settings.
   The multicells are organised in rows, each row is subdivied into columns.
   So for each multicell you have to specify a row and a column.
   If you group the multicells by rows in the sgs user interface, it is easier to visualize them.
3. In the export_to_pdf.php script adjust the folder id for your pdf templates folder and your pdf multicells folder in the variables $pdf_template_schema and $pdf_multi_cell_schema.

   Note: *the script uses a view called 'csg' in the module definitions. The csg view shows all needed fields and has the nosqlfolder=”true” property set. This way you can use one folder for all pdf templates and one folder for all multicells and access all this information from within these two folders.*

Run the pdf engine

Use: `http://<your-server>/bin/index.php?csg=export_to_pdf&template=myPDFForm`.

The export_to_pdf.php script will create a new object of class myPDFForm, customize it, if necessary, and outputs the content in form of a new pdf document.

This url runs the pdf form without additional data. If you want to add data you can do so by adding &folder=a_folder_id&view=a_view_name&id=an_id (or whatever other parameters you want to pass to your pdf template). In this case you have to process the additional parameters in your class myPDFForm with the customize() function.

The default csgPDFForm just prints the multi cells as is without additional data.

Of course you could write your complex and customized myPDFForm to include all possible kind of data. But it is probably easier to write one customized pdf class for one type of data. For example your write a customized class PDFContactForm to print sgs contacts into a pdf document. This way you keep your customized classes small and simple.

Customize the pdf template

To customize a pdf template you create your own pdf class in php like this:

```
class myPDFFORM extends csgPDFForm

{

     public function customize()

     {

          do customized adjustments here;

          for example get a csgAsset object from a csgSchema and
          push the data to the multicells. The csgSchema could be
          obtained by …
          &folder=a_folder_id&view=a_view_name&id=an_id

     }

}
```

The corresponding template asset in the csgFPDFTemplates module must have the same name than the customized class. This pdf template could then be called like this:

```
http://<your-server>/bin/index.php?
csg=export_to_pdf&template=myPDFForm&folder=80401&view=display&id=101
```

you can access the fields of the csgFPDFTemplate asset with:

`$tmplate_name = $this -> form -> name;` or

`$doc_size = $this -> form -> size;` etc

if you want to modify the content or style of a multi cell you can achieve it with this kind of statements:

`$this -> multi_cells[a_row, a_col] -> text = “foo”;`

This statement would set the text of the multi cell located at row `a_row` and column `a_col` to “foo”, where `a_row` and `a _col` are the integers you used in the module to place the multi cell within the template.

Example: PDFContactForm

Say you want to print the firstname and the lastname of a sgs contact asset in a pdf document.

You want to print the pdf document with a url like this: `http://<your-server>/bin/index.php?csg=export_to_pdf&template=PDFContactForm&folder=80401&view=display&id=101`

The template should have the form:

Firstname	**Lastname**
John	Doe

1. Create a new asset in csgPDFTemplates with name = PDFContactForm
2. create four multicells in csgPDFMultiCells
    - multicell[1,1] : specify a bold font and text = 'Firstname'
    - multicell[1,2] : specify a bold fond and text = 'Lastname'
    - multicell[2,1] : the text will be specified in your customized php class
    - multicell[2,2] : the text will be specified in your customized php class
3. create your customized class PDFContactForm

```
class PDFContactFORM extends csgPDFForm
{
      public function customize()
      {
            $folder = csgScript::get_argument("folder") -> value;
            $view = csgScript::get_argument("view") -> value;
            $id = csgScript::get_argument("id") -> value;
            $schema = new csgSchema($folder,$view);
            $contact = csgAsset::get_from_db($schema,$id);
            $this->multi_cells[2][1]->text = $contact->firstname;
            $this->multi_cells[2][2]->text = $contact->lastname;
      }
}
```

   copy this class definition in a file with name `PDFContactForm.php` in your custom classes directory. Please consult csg framework documentation for more information.

Status of the pdf engine

It is a work in progress.

Use at your own risk!

If you want to help testing for your needs, you are welcome

Version

Version 0.1

Do not consider the interface to be stable.

Always check the latest version before using a new feature.

To-Do:

- header & footer
- markup language for dynamic data in multi cells without php programming

Reference

Class csgPDFForm:

Is the base class to print a pdf form. It basically iterates over an array of MultiCell-csgAssets to write its content to a fpdf document. For own forms this class should be extended with your own class.

Create

- New csgPDFForm (csgAsset $form, csgSchema $multi_cells)
  // $form is based on the module csgFPDFTemplates.xml
  // $multi_cells is an associative array of csgAssets based on module csgFPDFMultiCells.xml
  // the keys of the array are the row number and column number to access a multi cell you can thus write $this -> multi_cells[row number, col number]
- customize()
  // used by export_to_pdf.php script to process customisation
- protected property $multi_cells
  // array of multi cells, each multi cell is a csgAsset of module csgFPDFMultiCells.xml
- protected property $form
  // csgAsset of module csgFPDFTemplates.xml
- output($name = null, $destination = null)
  // used by export_to_pdf.php script to output multicells to the pdf document.
  // for more info on the parameters consult fpdf.org