

# **Documentation application Comptage de pages couleurs**

(Faute de meilleur nom:-) )

*Application développée par Mathieu Kerhervé (06.77.70.83.65 – mathieukerherve14@gmail.com)*

## **I – But général de l'application**

Pouvant s'implémenter sur une page de commande d'impression, le but serait, pour le client, d'uploader son document, et d'avoir un retour sur le nombre de pages couleurs et noir et blanc que comporte son document, et pour l'entreprise, d'obtenir également ces informations, afin, dans le cas de documents à la taille important (thèses), de ne pas avoir à recompter le nombre de pages d'une part, et d'empêcher le client d'entrer des informations erronées de l'autre.

## **II – Fonctionnement de l'application**

### **A) FRONT**

L'application se compose d'une page front, en html, avec un appel vers des pages externes Bootstrap et jQuery. Cette page contient également du code JQuery, qui récupère le fichier de l'utilisateur, et l'envoie, en Ajax, vers la page PHP de traitement. Évidemment, cette page, d'une laideur incommensurable, et n'incluant rien des autres informations nécessaires à la commande (taille et type de papier, couverture, reliures, etc...) est le plus simple possible, et peut être modifiée à l'envie (en faisant attention de conserver les id, tel que spécifié dans le code, sous peine de ne plus pouvoir communiquer avec le JQuery et le PHP, ainsi qu'en laissant les inputs en disabled, les clients n'étant pas censés pouvoir modifier ces valeurs).

### **B) BACK – PHP**

La page de traitement PHP récupère le fichier envoyé par le client, et effectue 2 tests dessus : poids et type. Si le fichier n'est pas un PDF, ou pèse plus de 7MB (le poids peut évidemment être changé), un message s'affiche sur la page, indiquant le problème.

Sinon, le fichier sera copié sur le serveur, et la page appelle un programme externe, appelé GhostScript.

### **C) BACK – GhostScript**

Ce programme possède un device, inkcov, qui prend en paramètre un fichier PDF, et renvoie la couverture d'encre sur une page, par couleurs. Chaque couleur (Cyan, Magenta, Jaune et Noir) possède un chiffre par page, indiquant le pourcentage d'encre qu'il représente sur chaque page (par exemple, un chiffre de 0,02821 signifie que cette couleur représente 2,821% de couverture d'encre sur la page.

### **D) BACK – PHP**

La page récupère les résultats de Ghostscript, se débarrasse du superflu, stocke le tout dans un tableau de tableau ( [[p1], [p2] ... [pn]] ). Un test est effectué sur chaque page : si le taux de C, de M ou de J est > à 0, le numéro de la page en couleurs vient s'ajouter à un tableau.

Enfin, un objet json est renvoyé au front, comprenant : le nombre de pages total, le nombre de pages en couleurs, en noir et blanc, et (non utilisé pour le moment, mais pouvant l'être dans le futur) la liste des pages couleurs.

### **E) FRONT**

La page affiche une phrase récap des nombres de pages, et peuple les <inputs disabled> avec ces mêmes numéros.

### **III – Problèmes rencontrés**

Si le document envoyé est en RGB plutôt qu'en CMYB, ghostscript renvoie des taux de CMY supérieurs à 0, mais égaux entre eux. La page de traitement comporte du code commenté, spécifiant un test supplémentaire pouvant palier à ça, vérifiant l'égalité entre chaque couleur. Un problème se pose néanmoins dans ce cas : je suis tombé sur un PDF (shah.pdf) où une des pages, en couleur contient exactement la même distribution de CMYB, et n'est donc pas comptabilisé par le programme comme une page en couleur, si le test d'égalité est mis en place. Ce cas est, je pense, très rare, mais peut entraîner une sous facturation (moindre dans le cas d'une seule page, mais quand même). Les sites de concurrents ayant développé une application similaire (Printadom.com et copees.com) ont apparemment également ce problème avec ce document, comptant 139 pages couleurs là où il y'en a 140 !

### **IV – Mise en place de l'application**

Dans ce dossier, se trouvent :

- un dossier de pdfs pouvant être utilisé pour tester l'application
- un dossier public, dans lesquels se trouvent les scripts Bootstrap (si besoin), les script jQuery et le gif de chargement
- les binaires Ghostscript, une version Windows, et une version Linux
- la page FRONT formulairePDF.php
- la page BACK Traitement\_Donnees\_Pages.php

Le binaire Linux est déjà installé dans le dossier /home/dsprint/public\_html/Gestion\_client/Ghostscript du serveur o2switch (le fichier s'appelle gs-950). Il peut être copié et déplacé, mais attention à modifier les chemins d'accès sur la ligne 22 de la page BACK, de même que le chemin d'accès vers le dossier uploads. Un test peut être fait grâce au terminal fourni par o2switch.