

API de génération de fichiers PDF

1. Solution pdf24.org

L'API pdf24 proposée sur le site <http://fr.pdf24.org/>) est une solution relativement basique. Elle se révèle largement suffisante pour la création de documents PDF de faible complexité.

L'API (bibliothèque à installer sur votre site) est disponible en téléchargement libre et est décrite à l'adresse <http://fr.pdf24.org/JavaScript-pdf-api.html>.

Étudions dans le détail un script JavaScript d'exemple. La plus grande partie du code utile a été placée dans la section HTML <body>. Le code est intégralement reproduit ci-après. Un commentaire détaillé vous en est fourni.

Section HTML <head>

Il est nécessaire d'installer sur votre site la librairie évoquée ci-avant (`pdf24.js`). Dans la section HTML <head>, cette librairie est appelée comme suit :

```
<!-- Intégration de la librairie JavaScript pdf24.js -->
<script type="text/Javascript" src="pdf24.js"></script>
```

Section HTML <body>

```
/*
Création du fichier PDF et paramétrages de base
*/
/* Instanciation d'un nouveau document */
var fichierPDF = new PDF24Doc();
/* Définition du jeu de caractères */
fichierPDF.setCharset("UTF-8");
/* Nom du fichier (suffixé par pdf par l'API) */
fichierPDF.setFilename("test");
/* Définition du format de la page (A4 en format portrait) */
fichierPDF.setPageSize(210, 297);
```

Le code débute par l'instanciation d'un nouveau document PDF nommé `fichierPDF` par l'utilisation du constructeur `PDF24Doc`.

Le jeu de caractères UTF-8 est ensuite assigné.

La méthode `setFilename` de l'objet `fichierPDF` permet de lui affecter un nom. Il est à noter que l'extension `pdf` sera ajoutée automatiquement. La méthode `setPageSize` sert à indiquer le format de la page (format A4 en mode portrait dans notre cas).

Passons à la définition du contenu du document lui-même ensuite :

```
/*
Création du contenu du document PDF
```

```


*/
var contenuPDF = new PDF24Element();
contenuPDF.setTitle("Ceci est un document généré par pdf24.js en PDF");
contenuPDF.setUrl("http://www.pdf24.org");
contenuPDF.setAuthor("Christian VIGOUROUX");
contenuPDF.setDateTime("2010-04-15 8:00");
var texte = "";
texte = "Le contenu du document supporte des mises en forme HTML avec les
balises habituelles, comme ici du <b>gras</b> !!!<br /><br />";
texte += "<table border=1>";
texte += "<tr><td><font color=blue>Code</font></td><td><font
color=blue>Libellé</font></td><td><font color=blue>Vitesse
maxi</font></td></tr>";
texte += "<tr><td>V001</td><td>Porsche 930 Turbo</td><td
align=right>290</td></tr>";
texte += "<tr><td>V002</td><td>Porsche 964 Turbo</td><td
align=right>300</td></tr>";
texte += "<tr><td>V003</td><td>Ferrari 430</td><td
align=right>320</td></tr>";
contenuPDF.setBody(texte);

```

Par l'instanciation d'un objet PDF24Element, le contenu du fichier, nommé contenuPDF est créé.

Plusieurs méthodes sont à votre disposition pour affecter du contenu à cette variable contenuPDF, notamment :

- setTitle pour définir un titre au document,
- setURL pour que l'on puisse cliquer sur le titre du document et accéder à l'URL passée en paramètre,
- setAuthor pour signer le document,
- setDateTime pour dater le document,
- setBody pour définir le contenu du document.


 Vous l'avez remarqué dans l'exemple, il est possible de réaliser des mises en forme sophistiquées en intégrant dans le texte des balises HTML.

Il reste ensuite à affecter l'élément (c'est-à-dire la variable contenuPDF) à l'objet fichierPDF comme suit :

```

/*
Ajout du contenu dans le document PDF
*/
fichierPDF.addElement(contenuPDF);

```

 Il est bien entendu possible d'affecter à un même objet de la classe PDF24Doc différents contenus consécutivement.

Ensuite, mais cela n'a aucun caractère obligatoire, il est possible de programmer l'envoi du fichier PDF (test.pdf dans notre cas) via messagerie électronique à un destinataire signalé. Ceci a été prévu dans notre script :

```

/*

```

```
Paramétrage de l'envoi du fichier PDF par mail
*/
fichierPDF.setEmailTo("christian.vigouroux@online.fr");
fichierPDF.setEmailFrom("christian.vigouroux@univ-rennes1.fr");
fichierPDF.setEmailSubject("Fichier créé par pdf24");
fichierPDF.setEmailBody("Le fichier PDF est joint à ce mail. Merci à
www.pdf24.org!");
fichierPDF.setEmailBodyType("text");
```

Comme vous pouvez le remarquer, cette séquence de code ne présente aucune difficulté particulière, les noms des méthodes retenus par le développeur de cette API étant très explicites.

Dans notre cas le destinataire christian.vigouroux@online.fr recevra un mail signalé comme émis par l'émetteur christian.vigouroux@univ-rennes1.fr avec le texte "Fichier créé par pdf24" comme sujet, le texte "Le fichier PDF est joint à ce mail. Merci à www.pdf24.org!". Le fichier test.pdf se trouvera en pièce jointe dans ce mail.

L'envoi de mail est une séquence de code facultative. En son absence, l'API génère dans votre navigateur un nouvel onglet vous invitant :

- soit à télécharger le fichier PDF,
- soit à expédier le fichier PDF à un destinataire via un mail (l'adresse mail du destinataire vous sera demandée),
- soit à expédier le fichier PDF à un destinataire via un fax (le numéro de fax du destinataire vous sera demandé).

Le script se termine par la création effective du document par la méthode create appliquée à l'objet fichierPDF :

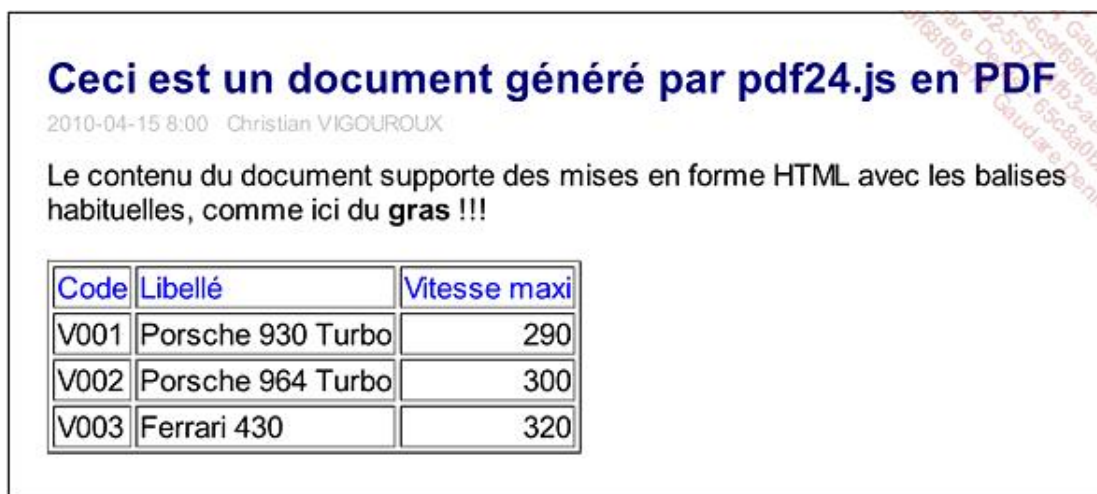
```
/*
Création du fichier PDF
*/
fichierPDF.create();
```

Exécution du script

Le script PDF_01.htm affiche ceci :



Le fichier test.pdf généré se présente comme suit :




2. Solutions alternatives

Il existe de nombreuses API générant du PDF à la volée. Leur étude dépasse bien entendu le cadre de ce livre. Signalons :

- l'API jsPDF (<http://parall.ax/products/jspdf>),
- l'API d'Adobe (<http://www.adobe.com/devnet/acrobat/javascript.html>).

L'API JavaScript d'Adobe Systems est bien sûr très complète.

 Rappelons que la société Adobe Systems est tout simplement à l'origine du format de fichier PDF (*Portable Document Format*).

Source Wikipedia : Le Portable Document Format, communément abrégé en PDF, est un langage de description de pages créé par la société Adobe Systems et dont la spécificité est de préserver la mise en forme d'un fichier - polices d'écritures, images, objets graphiques, etc. - telle qu'elle a été définie par son auteur, et cela quels que soient le logiciel, le système d'exploitation et l'ordinateur utilisés pour l'imprimer ou le visualiser.