reveal.js, faire des supports de présentation webeux

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 11 septembre 2013

https://www.bortzmeyer.org/reveal-js.html

Récemment, je me suis posé la question <https://www.bortzmeyer.org/logiciel-presentation-nouveau.html> de savoir s'il existait des alternatives rigolotes au logiciel de création de présentations que j'utilise d'habitude, LaTeX/Beamer. J'ai découvert qu'il existait notamment tout un tas d'outils fondés sur les techniques Web et j'ai choisi de commencer à utiliser reveal.js <http://lab.hakim.se/reveal-js/>. Je documente ici quelques points techniques découverts à cette occasion.

Ces outils Web ont en commun d'utiliser des techniques qui sont maîtrisées par beaucoup de gens (mais pas par moi) comme HTML, CSS ou JavaScript. Si on utilise un des thèmes pré-définis avec reveal.js <http://lab.hakim.se/reveal-js/>, l'utilisation de ce logiciel est très simple. Il existe plusieurs tutoriels, voici celui que je recommande <http://htmlcheats.com/reveal-js/reveal-js-tutorial-r>.

Mais si on utilise ces outils Web, c'est souvent pour modifier et mettre son propre thème. Pour des modifications simples et limitées, ajouter son code CSS est suffisant. Pour des modifications plus fondamentales, il faut aller plus loin. La procédure d'écriture de son propre thème est documentée <https://github.com/hakimel/reveal.js/blob/master/css/theme/README.md> mais pas vraiment triviale. D'abord, il faut installer quelques outils, notamment node.js et le gestionnaire de tâches Grunt <http://gruntjs.com/>. Sur une Debian stable (sur la version de développement, surnommée « sid », c'est un peu plus simple, sur une Ubuntu, ça marche pareil), voici comment j'ai fait (en partant des instructions officielles de reveal.js <https://github.com/hakimel/reveal.js#installation>). D'abord, installer node.js. Cela peut se faire de plusieurs façons, mais je tenais à avoir un paquetage Debian installé, pour faciliter les mises à jour ultérieures. (Et merci à l'auteur d'un bon article sur cette procédure <https://github.com/joyent/node/wiki/Installing-Node.js-via-package-manager>.)

```
% wget -N http://nodejs.org/dist/node-latest.tar.gz
# aptitude install python g++ make checkinstall
% tar xzvf node-latest.tar.gz
% cd node-v*
% ./configure
# checkinstall
```

Deux avertissements importants pour la dernière étape : checkinstall doit tourner sous root (à moins qu'il existe une méthode plus astucieuse) **et** surtout, il faut penser à changer le numéro de version dont la syntaxe par défaut ne plait pas à Debian. Lorsque checkinstall demande :

```
This package will be built according to these values:

0 -  Maintainer: [ stephane@tyrion ]
1 -  Summary: [ Node.js ]
2 -  Name:    [ nodejs ]
3 -  Version: [ v0.10.18 ]
4 -  Release: [ 1 ]
5 -  License: [ GPL ]
6 -  Group:   [ checkinstall ]
7 -  Architecture: [ i386 ]
8 -  Source location: [ node-v0.10.18 ]
9 -  Alternate source location: [  ]
10 - Requires: [  ]
11 - Provides: [ node ]
12 - Conflicts: [  ]
13 - Replaces: [  ]
```

Il faut changer la version :

```
Enter a number to change any of them or press ENTER to continue: 3
Enter new version:
>> 0.10.18
```

Ensuite, on se retrouve avec un paquetage node.js installé :

```
ii  nodejs    0.10.18-1 i386         Node.js
```

Maintenant, Grunt. Il n'est pas dans Debian pour un problème de licence <http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=673727>. Mais le gestionnaire de paquetages de JavaScript npm l'installe sans problème :

```
# npm install -g grunt-cli
```

Ensuite, plus qu'à récupérer reveal.js et à installer les dépendances :

```
% cd reveal.js
% npm install
```

Pour créer ou modifier un thèmes, il faut utiliser le langage SASS, le CSS étant trop primitif pour les thèmes compliqués de reveal.js. On installe le programme qui va traduire le SASS en CSS :

```
# aptitude install ruby-sass
```

Et on peut alors s'attaquer à la réalisation (ou la modification en partant d'un des thèmes existants) d'un thème en éditant un fichier d'extension `.scss`. La documentation de reveal.js <https://github.com/hakimel/reveal.js/blob/master/css/theme/README.md> guide un peu (mais il faut vraiment lire les commentaires dans le fichier `template/settings.scss`). Voici une partie de mon thème :

```
$mainColor: #000000;
$headingColor: #000000;
$headingTextShadow: none;
$headingTextTransform: none;
$backgroundColor: #ffffff;
$mainFont: 'arial', sans-serif;
$headingFont: 'arial', sans-serif;

// Custom CSS rules

.reveal em {
   font-style: normal;
   font-weight: bold;
}

.reveal pre {
  font-size: 110%;
}

// Add this element *before* the slides section

.reveal  #decoheader {
   display: block;
   position: absolute;
   top: 0%;
   left: 0%;
   margin: 0px;
   height: 12%;
   width: 100%;
   background-color: #000099;
   z-index: 50;
}
```

Vous voyez que le SASS comprend une bonne partie de CSS normal, comme les trois règles à la fin. Malheureusement, il n'existe pas de guide de création d'un nouveau thème, expliquant les pièges et les problèmes (par exemple, je pense que l'utilisation de `position: absolute` dans mon code ci-dessus est responsable de certains problèmes avec mon thème mais je ne peux pas en être sûr).

Une fois qu'on a terminé son SASS, on le compile en CSS :

```
% grunt sass

% find . -name 'bortzmeyer*'
./css/theme/source/bortzmeyer.scss
./css/theme/source/bortzmeyer.scss~
./css/theme/bortzmeyer.css
```

À chaque modification du `.scss`, on refera ce `grunt sass`. Vous pouvez voir le résultat sur mon exposé DNS à la Cantine <https://www.bortzmeyer.org/dns-cantine.html> (et vous rendrez compte ainsi que je ne connais pas grand'chose à CSS).

Autre manipulation technique qu'on peut avoir à faire avec reveal.js, l'impression en PDF. Elle est documentée <https://github.com/hakimel/reveal.js#pdf-export>, n'est pas extrêmement pratique mais marche comme indiquée. (Attention, cela produit un PDF d'une taille ridiculement élevée.)

Enfin, d'autres extensions nécessiteront JavaScript. Par exemple, reveal.js ne permet pas, par défaut, de mettre un compteur affichant la page courante. Ce besoin a déjà été exprimé <https://github.com/hakimel/reveal.js/pull/56>. Pour moi, la solution de mohikaner marche : on ajoute son code JavaScript à la page HTML (il doit y avoir une solution plus propre, surtout pour partager ce code entre plusieurs exposés, mais je ne me suis pas penché sur la question), et on ajoute un élément de la classe correspondante dans le source HTML :

```
<div id="decofooter">
    <p><a href="http://www.bortzmeyer.org/">Stéphane Bortzmeyer</a> -
       <span class="slidenumber"/></p>
</div>
```