

[BASH] Utiliser l'API fontello.com

générer et mettre à jour une police d'icônes webfont en quelques secondes

mercredi 14 octobre 2015, par [placido](#)

Allons directement dans le vif du sujet en créant un fichier **Makefile** [1] quelque part en amont de votre projet :

```
1. # Makefile
2.
3. FONT_DIR    ?= ./css/icons
4. PROJECT_NAME ?= projet_foo
5. FONTELLO_HOST ?= http://fontello.com
6.
7. fontopen:
8.     @if test ! `which curl` ; then \
9.         echo 'Install curl first.' >&2 ; \
10.        exit 128 ; \
11.        fi
12.    curl --silent --show-error --fail --output .fontello \
13.        --form "config=@${FONT_DIR}/config.json" \
14.        ${FONTELLO_HOST}
15.    x-www-browser ${FONTELLO_HOST}/`cat .fontello`
16.
17.
18. fontsave:
19.     @if test ! `which unzip` ; then \
20.         echo 'Install unzip first.' >&2 ; \
21.         exit 128 ; \
22.         fi
23.     @if test ! -e .fontello ; then \
24.         echo 'Run `make fontopen` first.' >&2 ; \
25.         exit 128 ; \
26.         fi
27.     rm -rf .fontello.src .fontello.zip
28.     curl --silent --show-error --fail --output .fontello.zip \
29.         ${FONTELLO_HOST}/`cat .fontello`/get
30.     unzip .fontello.zip -d .fontello.src
31.     rm -rf ${FONT_DIR}/*
32.     mv `find ./fontello.src -maxdepth 3 \
33.         \( -name "config.json" -o -name "*.woff" -o -name
34.         "*.ttf" -o -name "*.svg" -o -name "*.eot" \) ` ${FONT_DIR}
35.     mv `find ./fontello.src -maxdepth 3 -name "${PROJECT_NAME}-codes.css" `
36.     ${FONT_DIR}/codes.scss
37.     rm -rf .fontello.src .fontello.zip
```

[Télécharger](#)

Passons rapidement le script en revue

Tout d'abord, on doit éditer le script pour renseigner 2 informations :

- **FONT_DIR**, qui est l'emplacement du dossier qui va recevoir les éléments issus de fontetto (.ttf, .woff, .css, .css, ...), relativement au fichier Makefile. Personnellement j'ai choisi un dossier "icons" dédié à cela, et distinct des polices (webfonts) complètes qui elles sont dans un autre dossier "fonts".
- **PROJECT_NAME** qui désigne le nom du projet.

A présent, direction fontello.com où vous pouvez commencer à composer votre webfont personnalisée **projet_foo**. Enregistrez (bouton en haut à droite), téléchargez le .zip et récupérez le fichier config.json (qui est le seul qui nous intéresse) et copiez le dans le dossier FONT_DIR.

À l'usage

À partir de maintenant, l'ajout ou modification d'icônes va se faire en un clin d'oeil via 2 commandes successives, issues du fichier *Makefile* :

1. **make** fontopen

inspecte le fichier *config.json* et ouvre le navigateur avec les paramètres d'édition sauvegardés.

1. **make** fontsave

récupère la version mise à jour du projet_foo en écrasant les anciens fichiers.

Pour ma part, c'est surtout le fichier *projet_foo-codes.css* avec les codes qui m'importe (voir et modifier au besoin la ligne 32) ; c'est pourquoi je renomme en *codes.scss* pour l'appeler ultérieurement dans la compilation SASS.

P.-S.

Ultime remarque, si vous constatez un problème d'affichage des icônes après une mise à jour, essayer de changer le paramètre en fin d'url dans la déclaration css font-face.

1. **@font-face** {
2. **font-family**: 'projet_foo';
3. **src**: url('icons/projet_foo.eot?248676');
4. /*
5. ...
6. */
7. }

Télécharger

devenant par exemple :

1. **@font-face** {
2. **font-family**: 'projet_foo';
3. **src**: url('icons/projet_foo.eot?1039286');
4. /*
5. ...
6. */
7. }

[Télécharger](#)

C'est un désagrément dû aux navigateurs, qui allouent un cache spécifique pour les webfonts.

Notes

[1] `chmod +x Makefile` pour le rendre exécutable