Gallery

Gallery est un module Ocsigen pour « La Vie Est Un Jeu ».

Table des matières

Chapitre I

Le projet

I.1 Présentation

Gallery est un module Ocsigen permettant, comme son nom l'indique, d'afficher de manière simple et intéractive une galerie d'images.

Le concept est de pouvoir appeler une des fonctions du module en spécifiant un simple chemin et celle-ci renvoiera un élément d'AST valide Ocsigen de type div contenant un affichage des images contenues dans le dossier, ainsi que les sous-dossier. Il sera possible de parcourir toute l'arborescence et de voir les images s'y trouvant.

I.2 Destinataires

Gallery est un module destiné à être utilisé par :

- « La Vie Est Un Jeu ». (voir détails plus loin) ;
- La communauté des utilisateurs Ocsigen.

Il devrait donc à terme être mis en ligne sur le site Ocsigen afin de servir d'exemple d'utilisation d'Ocsigen. Il pourra aussi être récupéré tel quel pour être intégré dans un site web Ocsigen ayant besoin d'afficher une galerie d'images.

I.3 Fonctionnalités requises

La liste suivante présente les fonctionnalités minimales requises pour que ce module soit terminé.

- Gallery doit être un module
- Il doit être complétement indépendant du site web sur lequel il se trouve
- Il ne doit donc pas avoir besoin de connaître les services existants
- Il ne doit pas créer ses propres URL
- Il doit utiliser Eliom

- Il doit y avoir une fonction prenant en paramètre un chemin et renvoyant un élément de type div contenant la galerie
- L'affichage doit contenir des miniatures des images
- Il doit être possible d'afficher l'image dans sa taille originale

- Il doit être possible de parcourir les dossiers et les sous-dossiers
- La distinction entre des fichiers et dossiers doit être clair
- La récupération de la liste des fichiers des sous-dossiers devra être faite uniquement lorsque c'est nécessaire (Pour cela, le client appelera une fonction du serveur au moment du click sur l'icône du dossier)
- Il devra être possible de revenir au dossier parent (sauf à la racine)

- L'affichage des images ne doit pas provoquer de changement de page
- L'affichage des images devra être sur la même page
- L'affichage des images devra se faire en plein centre de la page
- Il devra être possible de "fermer" l'affichage d'une image pour revenir à la liste des images et dossiers

I.4 Fonctionnalités bonus

La liste suivante présente des fonctionnalités dites “bonus” proposées pour être ajouté au module mais pas obligatoires.

- Les chemins peuvent être envoyés de plusieurs façons (liste, string..)

- Un fichier CSS pourra être inseré afin de proposer un style graphique par défaut
- Le fichier CSS pourra être inseré grâce à une fonction du module

- Il pourra être possible d’afficher le nom des images
- Il pourra être possible d’afficher l’extension des images

- Le module pourra avoir une interface (mli ou eliomi) décrivant les fonctions utiles du modules uniquement

- Il pourra être possible de générer automatiquement les miniatures grâce au binding d’**ImageMagick** pour OCaml
- Cette génération pourra se faire grâce à une fonction du module
- Cette génération pourra se faire uniquement lorsque les images miniaturisées n’existent pas déjà afin d’alléger le processus
- Cette génération pourra se faire au choix : jamais, au lancement du serveur, au lancement du site web par les utilisateurs

- Il pourra y avoir un affichage des images d’une taille intermédiaire pour éviter de charger des images trop lourdes et ces images seront générées de la même manière que les miniatures

- L’utilisateur pourra à tout moment savoir à quel chemin il se trouve
- L’utilisateur pourra cliquer sur les dossiers constituant le chemin afin d’y retourner de manière rapide o

- Il pourra être possible de naviguer à travers les images en cliquant sur des boutons
- Il pourra être possible de naviguer à travers les images en utilisant les flèches

Chapitre II

Technique

II.1 Technologies utilisées

Ce module utilisera **Ocsigen** et en particulier **Eliom**.

Plus de détails sur Ocsigen et ses différents services :

http://ocsigen.org/

Plus de détails sur le choix de l'utilisation de cette technologie sur le cahier des charges du projet « La Vie Est Un Jeu » à l'URL suivante :

http://eip.epitech.eu/2014/

II.2 Hébergement

Le module est hebergé et versionnés sur un dépôt GitHub :

https://github.com/db0company/Gallery

Le repot est sur le compte de l'auteur et est public.

Ainsi, il sera possible pour les utilisateurs du module de parcourir les sources facilement mais aussi d'y contribuer en créant un *fork* du dépôt et en y effectuant des *pull-request*. Le système de tickets de l'interface sera aussi très utile.

II.3 Arborescence

II.3.1 Fichiers nécéssaires à l'utilisation de Gallery

`gallery.eliom` est le module complet Gallery.

`gallery.eliomi` est éventuellement l'interface du module Gallery.

`pathname.ml` et `pathname.mli` est un module indépendant de Gallery et d'Ocsigen. Il propose un type et des fonctions pour gérer les chemins de fichiers et dossiers.

`split.eliom` est un module Eliom permettant de découper une chaîne de caractère en utilisant un caractère séparateur. Son implémentation est différente côté client et côté serveur, dû à l'absence du module `Str` côté client.

II.3.2 Fichiers du dépôt

`static/css/gallery.css` est un fichier de style proposé par défaut pour la galerie.

`example.eliom` est un fichier d'exemple avec un service simple utilisant Gallery.

`example.conf` est le fichier de configuration de l'exemple.

`static/images` est un dossier contenant des images pour illustrer l'exemple.

`Makefile` est un Makefile d'exemple.

`static/css/style.css` est un fichier de style proposé pour l'exemple.

`doc/CDC_Gallery.pdf` est le cahier des charges du module Gallery.

`doc/CDC_Gallery.tex` est la source de la documentation du module Gallery.

II.4 Conventions de code et de style

- Coder proprement et séparer le code en de nombreuses fonctions et sous-fonctions.
- Le projet utilise des modules.
- Les types sont souvent abstraits.
- Le projet comporte un ou plusieurs tests utilisant le module.
- N'est exporté dans l'interface du module que ce qui est nécessaire.
- Toutes les valeurs exportées sont justifiables.
- Si un type est complexe, un module contenant ce type avec des fonctions outils
- permettront de l'utiliser.

- Le coding style suivant est utilisé :
  - Header obligatoire contenant au minumum : Auteur (pseudo ou nom complet, pourquoi pas un mail ou un site), Nom du module et description.
  - Les lignes des fichiers ne doivent pas dépasser 80 colonnes.
  - Il doit y avoir un commentaire au dessus de chaque valeur du module pour expliquer son rôle.
  - Au dessus de ce commentaire doit se trouver la signature de la valeur en commentaire également, sauf si c'est trivial.
  - Une fonction ne doit pas contenir plus d'un filtrage. Appeler des sous-fonctions.
  - Les commentaires doivent être alignés à gauche et faire toujours la même longueur : 79 ou 80 colonnes.
  - L'interface associée au module doit contenir les mêmes commentaires pour les valeurs sauf bien sûr la signature puisqu'elle est dans le code directement.
  - Les sauts de lignes nombreux permettent d'aérer le code.

Chapitre III

Planning

III.1 Planning prévu

Jour 1 Réflexion sur le Module, Affichage simple d'images.
Jour 2 CSS, Module de chemins.
Jour 3 Ajout des dossiers et sous-dossiers.
Jour 4 Affichage d'images en taille réelle, parcours d'images.
Jour 5 Génération automatique de miniatures.

III.2 Difficultés rencontrées

- **Installation d'Ocsigen** (16 Mai 2012)
  Tâche en cours : Compilation classique du bundle Ocsigen sur un Ubuntu LTS 12.04 en 64 bits.
  Problème survenu : /usr/bin/ld : cannot find -lgdbm_compat
  Résolution : Installation supplémentaire requise de `libgdbm-dev`.

- **O'Closure pour Graffiti** (30 Mai 2012)
  Tâche en cours : Réalisation du tutoriel des applications Eliom côté client et côté serveur.
  Problème survenu : Il manque O'Closure.
  Résolution : Réinstallation complète du bundle, à partir du dépôt Darcs et en indiquant l'option d'installation d'OClosure.

- **Intégration de fichier statique** (5 Juin 2012)
  Tâche en cours : Utilisation d'un fichier CSS pour styliser la galerie.
  Problème survenu : `ocsigenserver: main: Fatal - Error in configuration file: Unexpected tag <static> inside <site dir=">`
  Résolution : La balise <static> appartient à l'extension staticmod. Il faut donc le charger avant avec `<extension findlib-package="ocsigenserver.ext.staticmod"/>`

- **Intégration de fichier CSS** (11 Juin 2012)
  Tâche en cours : Utilisation d'un fichier CSS pour styliser la galerie.
  Problème survenu : Le CSS ne se charge pas du tout.
  Résolution : Je suis passé à autre chose et ça s'est mis à marcher magiquement.

- **Mise à jour d'Ocsigen** (11 Juin 2012)
  Problème survenu : Sortie de la nouvelle version d'Ocsigen avec les nouveaux noms de module. Mise à jour complète puis modification de tout les fichiers.

- **Utilisation d'Eliom et des interfaces de module** (19 Juin 2012)
  Tâche en cours : Passage à Eliom pour utiliser `js_of_ocaml`. Modification des `.ml` en `.eliom` et ajout de compilation côté client.
  Problème survenu : `Error: Could not find the .cmi file for interface gallery.mli`.
  Résolution : Il n'est pour l'instant pas possible d'utiliser des interfaces de modules avec Eliom. On espère que ce sera bientôt possible d'utiliser des `.eliomi`.

- **Remplacement d'une div par une autre** (27 Juin 2012)
  Tâche en cours : Parcours de sous-dossiers, remplacement du dossier actuel par le contenu du nouveau dossier sur lequel on vient de cliquer.
  Problème survenu : Comment accéder à un élément dans lequel on se trouve lorsque l'on utilise la syntaxe `div ~a:[a\_onclick ...` ?
  Résolution : Utilisation des arrow, de fonctions externes et de `getElementById`

- **Récupération d'élément par son id** (2 Juillet 2012)
  Tâche en cours : Parcours de sous-dossiers, remplacement du dossier actuel par le contenu du nouveau dossier sur lequel on vient de cliquer.
  Problème survenu : Aucune erreur, ni pendant la compilation, ni pendant le lancement, ni dans la console Javacript. Et pourtant, il ne se passe rien.
  Résolution : Il ne trouve pas l'élément parce que la page n'est pas complétement chargée. Il faut ajouter un `Eliom_service.onload`.

- **Utilisation des arrow** (3-9 Juillet 2012)
  Tâche en cours : Parcours de sous-dossiers.
  Problème survenu : Il y a peu d'explications sur les arrow sur le site puisque la fonctionnalité est encore au stade expérimental.
  Résolution : Explication des arrow avec `try.ocaml`.

- **Utilisation du module CamlImages** (9 Juillet 2012)
  Tâche en cours : Génération automatique de miniatures.
  Problème survenu : Le module CamlImages est difficile à compiler. Une fois celui-ci fonctionnel, on s'aperçoit que beaucoup de fonctions n'ont pas été implémentées pour beaucoup de format d'images.
  Résolution : Utilisation des bindings d'ImageMagick pour OCaml.

- **Str.split côté client** (11 Juillet 2012)
  Tâche en cours : Portage de l'affichage du contenu d'un dossier côté client.
  Problème survenu : Le module `Str` n'existe pas côté client.
  Résolution : Recode complet de la fonction `Str.split`.

- **Récupération de services côté client** (11 Juillet 2012)
  Tâche en cours : Parcours de fichiers et dossiers, lien vers un sous-sous-dossier. La génération de l'affichage a dû être entièrement re-pensé pour le côté client puisque les contraintes sont complétement différentes côté client et côté serveur.
  Problème survenu : Comment faire en sorte d'appeler la fonction côté serveur qui récupérera la liste des fichiers si l'on a pas accès au service ?

Résolution : En cours.

- **Recupération des fichiers dans le static_dir** (16 Juillet 2012)
  Problème survenu : Les fichiers doivent forcément être dans le static_dir. On doit parcourir le dossier avec opendir. Il faut donc pouvoir récupérer le path du static_dir.
  Résolution : En cours.

III.3 Planning final

Dead-line finale Galerie : 1 Aôut 2012.

Chapitre IV

À propos

IV.1 Auteur

- **Barbara Lepage**
- db0company@gmail.com
- http://db0.fr/

IV.2 La Vie Est Un Jeu

« La Vie Est Un Jeu » est un projet sur trois ans dans le cadre des « Epitech Innovative Projects » mené par un groupe de dix étudiants.

Ce projet, sous forme d'un site web et d'applications mobiles, propose à ses utilisateurs de pimenter leur quotidien. Pour cela, on leur propose de manière ludique de se fixer des objectifs, les réaliser, les collectionner et enfin les partager.
C'est donc à la fois un jeu et un réseau social, destiné à tous les âges !

Pus de détails sur le projet sont disponibles sur le site vitrine du projet :

http://eip.epitech.eu/2014/

IV.3 Le Groupe

Lepage Barbara lepage.barbara@gmail.com

Caradec Guillaume guillaume.caradec@gmail.com

Corsin Simon simoncorsin@gmail.com

Glorieux François fra.glorieux@gmail.com

Klarman Nicolas nickoas@gmail.com

Lassagne David david.lassagne@gmail.com

Louvigny Guillaume guillaume@louvigny.fr

El-Outmani Youssef youssef.eloutmani@gmail.com

Le-Cor Wilfried wilfried.lecor@gmail.com

Lenormand Frank lenormf@gmail.com

Le détail des rôles de chacun des membres du groupe est disponible sur le cahier des charges à l'URL suivante :

http://eip.epitech.eu/2014/