COFFEE TASK

Fiche de sécurité, développement et scénarios de tests

Anthony Réformatsky - Samir Lerognon - Vanessa Dettwiller

Fiche sécurité

Sécurité du site:

Les comptes utilisateurs seront protégés par un identifiant/mot de passe.

Protection des injections SQL : utilisation de PDO et requêtes préparées.

Backend :

- Mise en place de fail2ban, afin de palier aux attaque de type brute-force.
- Backups régulier des fichiers et bases de données conservés sur 30 jours.
- Dispositifs anti-DDOS au sein du data-center d'OVH.
- CHMOD particulier suivant les dossiers. Par exemple, 660 dans le dossier d'upload.
- Bloquer l'exécution de tous les scripts PHP sauf index.php (le point d'entrée)
- Lors de l'upload des fichiers seul l'id est marqué comme nom du fichier , ceci afin d'éviter des problèmes avec les nom des fichiers contenant par exemple des slashes ou des points.

Base de donnée :

- Sauvegardes aussi conservés sur 30 jours.
- Limit des droits d'accès de l'utilisateur courant (ne peut que SELECT / UPDATE), tout autre modification de la structure ou de suppression n'est pas autorisé afin d'éviter de possibles pertes de données.

Mot de passe :

Les mots de passes sont hashés en SHA-2 (afin de palier aux problèmes de SHA-1) et jamais stockés en clair.

Préconisations :

Utilisation de mots de passes forts, et non-publication des tâches à n'importe-qui.

Améliorations possibles :

- Dans le futur, il faudra probablement opter pour un nouvel algorithme de chiffrement des mots de passe avec un cost plus important, mais donc aussi plus long à générer.
- Protection CSRF à chaque requête Ajax.
- Mise en place de certificats SSL 4096 bits.

Développement & Scénario de tests

Liste des différentes librairies utilisées :

- SlimPHP (framework base) : MIT / https://github.com/rivetchip/Slim/tree/develop
- SlimSkeleton : MIT / https://github.com/rivetchip/slim-skeleton/tree/develop
- Moment.js : MIT / https://momentjs.com/
- ROME : MIT / https://bevacqua.github.io/rome/
- SimpleTemplating : MIT / https://github.com/rivetchip/
- SimpleAjax/PrettyDate : MIT / https://github.com/rivetchip/simple-web-utils
- Normalize.CSS : MIT / https://necolas.github.io/normalize.css/

Développement :

Les différentes libraires se trouvent dans le dossier /src/

Base de données :

/config.php la configuration général de l'app, avec les accès à la base.

La structure de la base se trouve dans /database/kanban.sql

PHP :

La totalité de toutes les fonctions et classes sont documentés suivant le format PHPDoc.

/app/xxxController.php : les controllers suivant les différentes pages

/app/ApiController.php : le controller de base regroupant toutes les actions pour les appels AJAX.

-> Pour ajouter d'autres pages / appels, il faut les rajouter dans /routes.php

/app/AbstractController.php : canAction() les différents droits d'accès suivant les différentes actions.

/uploads/ le dossier contenant les différents fichiers uploadés (avatar / images sur les tâches) n'est contenu que l'id de l'utilisateur ou l'id du fichier (qui fait référence à la table "files" de la base).

Javascripts :

/jscripts/core.js les fonctions de base incluses sur chaque page.

/jscripts/xxx.js : les noms des fichiers sont nommés en fonction du nom de la template et son automatiquement appelés si ce dernier existe.

Arborescence – Développeur

```
|.
|-app                  // application ( controllers & modèles BDD )
|---Interfaces  // interface des classes
|---Models             // models de l'application
|-database             // contient les scripts & design BDD
|-documentations// documentations développeur
|-img                  // dossier des images, relative au style
|-jscripts             // scripts javascripts
|---3rdparty           // modules externes javascripts
|-----moment           // Moment.js
|-----[...]
|-----rome             // ROME datepicker
|-----[...]
|-src                  // coeur
|---Database           // accès PDO
|---[...]
|---Slim               // framework PHP
|---PHPMailer          // framework mail
|---[...]
|-style                // styles de l'application
|---fonts              // polices intégrées au site
|-templates            // templates:
|---base               // blocs de base header/footer, etc
|---elements           // éléments base : snackbar,messages erreurs, etc
|---mails              // templates emails
|-uploads              // dossier d'uploads des fichiers
|---avatars            // avatars des utilisateurs
|---files              // photos des tâches
```

Scénario de tests :

Les tests ont été faits manuellement. Le temps manquait pour rédiger différents tests unitaires.

Liste des différents tests :

- Connexion / Inscription / déconnexion / modification du profil
- Liste des différents informations de l'utilisateur dans le menu

- Création / suppression d'un projet
- Ajout des différents utilisateurs et d'un modérateur au projet
- Liste les différents projets et différentes tâches suivant les droits d'accès de l'utilisateur.

- Ajout / modification / suppression d'une catégorie

- Ajout d'une tâche avec un utilisateur lié ou à soi-même
- Ajout de photos sur une tâche et validation de ces dernières via l'espace admin / modérateur
- Validation d'une tâche avec envoi d'email à l'utilisateur
- Expiration d'une tâche avec date
- Suppression d'une tâche
- Partage d'une tâche avec mini-popup de visualisation

- Filtrer les différentes tâches suivant des critères

- Mini-popup d'informations destiné aux utilisateurs.

- Test du responsive sur toutes les pages et accès aux mêmes fonctionnalités en mode touch
- Test général des différents droits d'accès

Documentation API Développeur

Les différents fonctions sont documentées suivant le standard PHPDoc.

La documentation développeur générée des différentes APIs se trouvent dans le dossier :
/documentations/dev-docs/index.html