Aix-Marseille Université

M3103 - Programmation Web côté serveur

Réalisation d'un site de partage et de création de dessins

Auteurs :
Lucien Aubert
Tom Henoch

Enseignants :
Brett Desbenoit
Laurent Carmignac

Table des matières

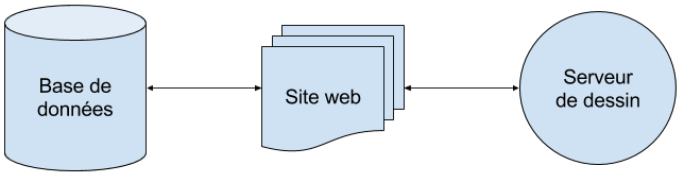
1 Introduction

L'utilisateur sera en mesure de dessiner avec des outils basiques (crayon, carré, cercle, lettrages) et de partager sa création. Les dessins ainsi créés sont modifiables par plusieurs utilisateurs.

Les utilisateurs pourront commenter les dessins et les marquer comme "J'aime".

2 Structure

L'application est composée de trois modules qui interagissent :
- un serveur web (site + base de données)
- un serveur spécifique à l'édition de dessin
- une application client servie par le serveur web



La base de données n'est accessible que par le site web, doté d'une API type REST permettant au serveur de dessin de lui passer des requêtes.

2.1 Serveur Web

Le site web est développé avec le framework Symfony 3 et utilise une base de données MariaDB. Il est hébergé sur un serveur Apache 2.4.

Il joue les rôles d'interface utilisateur et de gestionnaire de base de données.

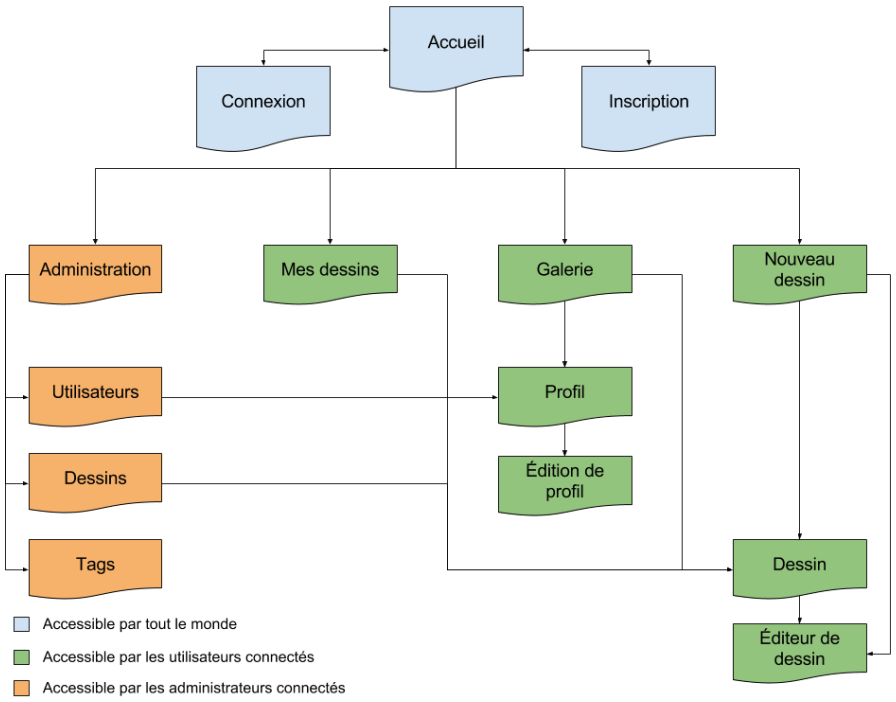
C'est lui qui donne la permission au serveur de dessin de démarrer une connexion avec un client.

Pour l'utilisateur, il permet de
- s'inscrire
- se connecter (authentification par nom et mot de passe)
- créer un nouveau dessin
- consulter la galerie
- rechercher un dessin par nom
- rechercher un dessin par tag
- modifier son profil
- modifier ses dessins (titre, auteurs, tags)

2.1.1 Ajax

Nous avons privilégié des interfaces avancées pour certains formulaires (champs d'édition de tags, d'auteurs, de titre). Pour se faire nous avons eu recours à l'utilisation de `XMLHttpRequest`, une object Javascript qui permet de faire envoyer des requêtes HTTP par le client. Le client envoie donc des requêtes `GET` vers le serveur pour signaler l'ajout ou la suppression d'un tag ou d'un auteur, le serveur envoie sa réponse sous forme d'un fichier JSON avec une message d'erreur ou de succès qui sera parsé par le client pour afficher une notification.

2.1.2 Plan du site



Les pages protégées rediriges vers la page de connexion si l'utilisateur n'est pas connecté et vers la page d'accueil quand il tente d'accéder à du contenu qui ne lui est pas accessible.

Symfony aidant, il nous est possible de vérifier si l'utilisateur nous a donné le cookie d'une session grâce à cette simple condition, `$this` étant notre controleur.

```
1 if(!$this->getUser())
2   return $this->redirectToRoute('login');
```

La chaîne `'login'` correspond ici à la route vers laquelle rediriger le client. Le système de routing du framework permet de définir des routes en leur attribuant un nom, des arguments et une méthode qui sera exécutée quand la route sera demandée par un client. Ces définitions sont faites sous forme de commentaires formatés comme suit.

```
/**
* @Route(
*   "/sketch/{sketchId}",
*   requirements={"sketchId": "\d+"},
*   name="sketch"
* )
*/
```

Les fichiers source sont ensuite traités par Symfony et les commentaires parsés pour produire un module de routage sur mesure.

3 Conclusion

La création de cette application nous a permis de découvrir le framework Symfony 3 ainsi que la technologie des WebSockets. L'utilisation du Javascript pour le développement complet de l'outil de dessin nous a appris beaucoup sur ce langage, tant côté serveur que côté client.

Le déploiement de l'application a été un défi que nous avons relevé en nous documentant sur le serveur web que nous utilisions, Apache 2.4, ce qui nous a permis d'approfondir les connaissances que nous avons acquis en cours de réseau.

Références