

Modèle MVC

Nous développons l'application web gsb avec PHP et Symfony. Comme de nombreux frameworks PHP, Symfony utilise le motif d'architecture logicielle MVC (pour Modèle-vue-contrôleur). Il permet de travailler les modèles, les vues et les contrôleurs séparément pour une meilleure maintenabilité et évolutivité du code.

Le pattern MVC permet de bien organiser son code source, son but est justement de séparer la logique du code en trois couches que l'on retrouve dans des dossiers distincts.

- **Modèle** : cette couche gère les données du site. Son rôle est d'aller récupérer les informations « brutes » dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur. Dans Symfony la notion de modèle se retrouve sous la forme (entre autre) d'une Entité. Une entité est une classe PHP, qui peut être connectée à une table de votre base de données via l'ORM. Lorsqu'une entité est liée à une table, via l'ORM, il y a en général un fichier "repository" associé. Un repository permet la génération de requêtes simples ou complexes et dont le développeur peut modifier à volonté. Un ORM (Object Relation Mapper) permet de gérer manipuler et de récupérer des tables de données de la même façon qu'un objet quelconque. Symfony utilise Doctrine comme ORM dans son système par défaut.
- **Vue** : cette couche se concentre sur l'affichage. Elle récupère des variables pour savoir ce qu'elle doit afficher. On y trouve essentiellement du code HTML mais aussi quelques boucles et conditions. Nos fichiers vues seront organisés dans un dossier nommé templates, dans Symfony ces fichiers sont créés avec Twig : un moteur de modèle flexible, rapide et sécurisé.
- **Contrôleur** : cette couche gère la logique du code qui prend des décisions. C'est en quelque sorte l'intermédiaire entre le modèle et la vue : le contrôleur va demander au modèle les données, les analyser, prendre des décisions et renvoyer le texte à afficher à la vue. Le contrôleur contient exclusivement du PHP. Dans Symfony nous avons les controllers et les services par exemple, dans les services il y aura l'intelligence du code et dans les controllers toute l'organisation logique.

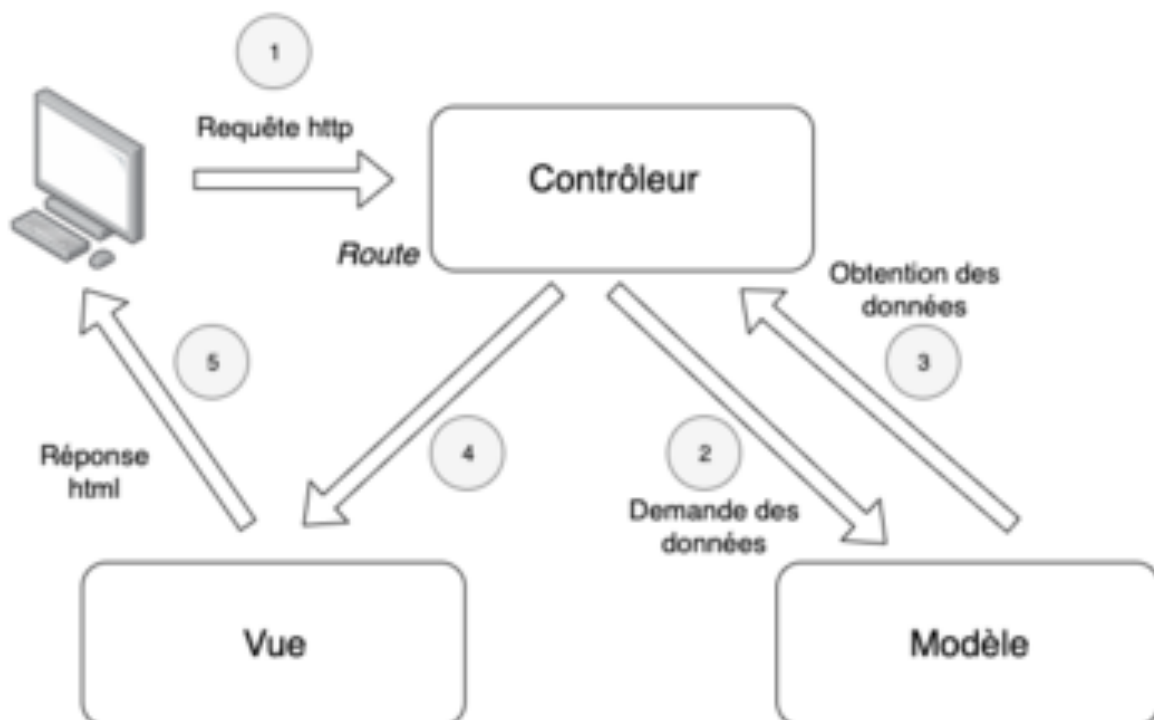


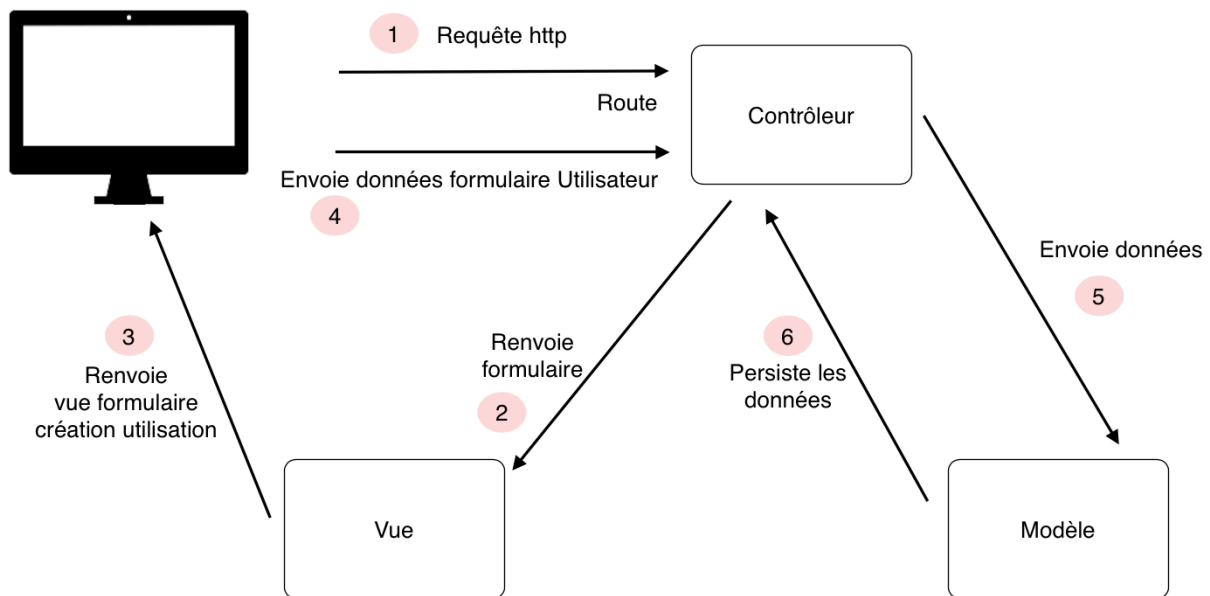
Schéma méthode

Utilisateur

Créer :

Route : /admin/user/new

Ce que nous avons besoin : `UserAdminController.php`, `admin/user/new.html.twig`, `App\Entity\User.php`, `App\Form\UserType.php`, `EntityManagerInterface.php`, `Doctrine`.

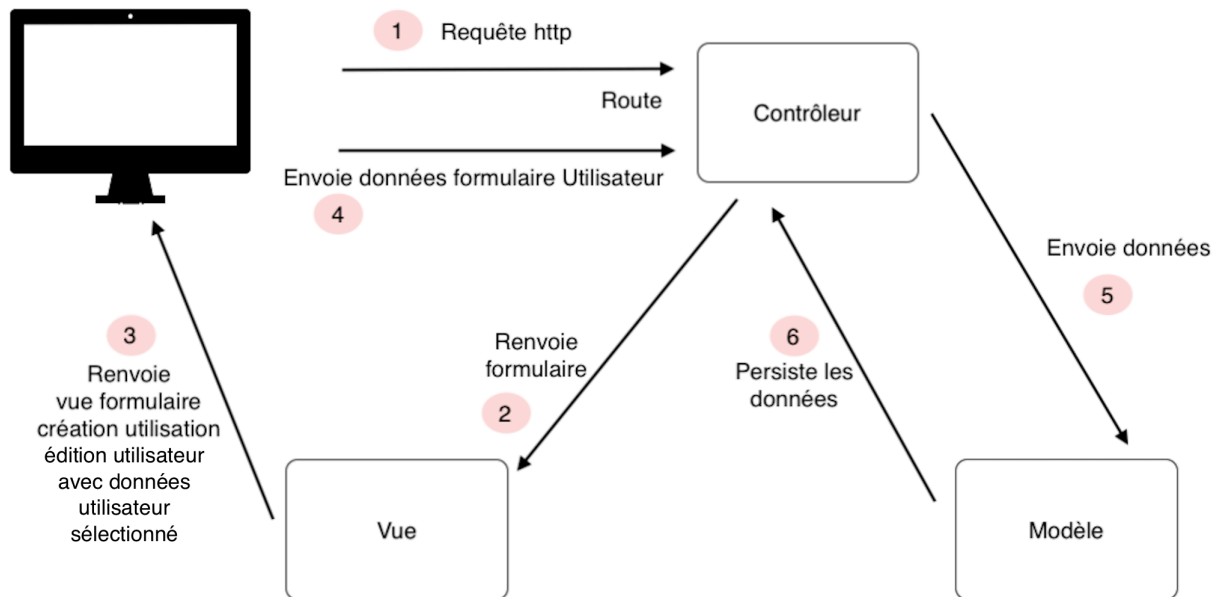


L'utilisateur entre la route dans son navigateur, une requête http sera envoyée au Contrôleur (`UserAdminController.php`), dans celui-ci est instancié l'entité utilisateur (`User.php`) et le formulaire utilisateur (`UserType.php`), le contrôleur renvoie ces données à la vue (`new.html.twig`), la vue est renvoyée à l'utilisateur. Ensuite, l'utilisateur est invité à insérer des données dans les différents formulaires, les données sont envoyées au contrôleur, le contrôleur vérifie les données puis instancie le modèle (`EntityManagerInterface.php`, `Doctrine`), il envoie ces données au modèle, celui-ci valide et persiste et flush les données.

Éditer :

Route : /admin/user/{id}/edit

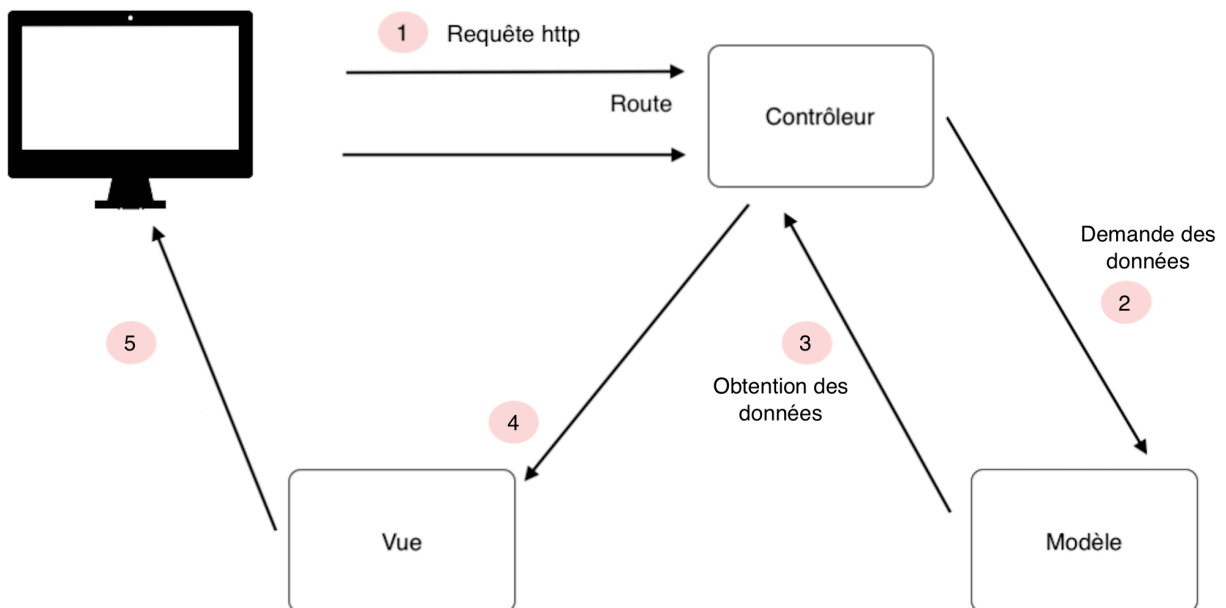
Ce que nous avons besoin : `UserAdminController.php`, `admin/user/edit.html.twig`, `App\Entity\User.php`, `App\Form\UserType.php`, `EntityManagerInterface.php`, `Doctrine`.



Lister :

Route : /admin/user/list

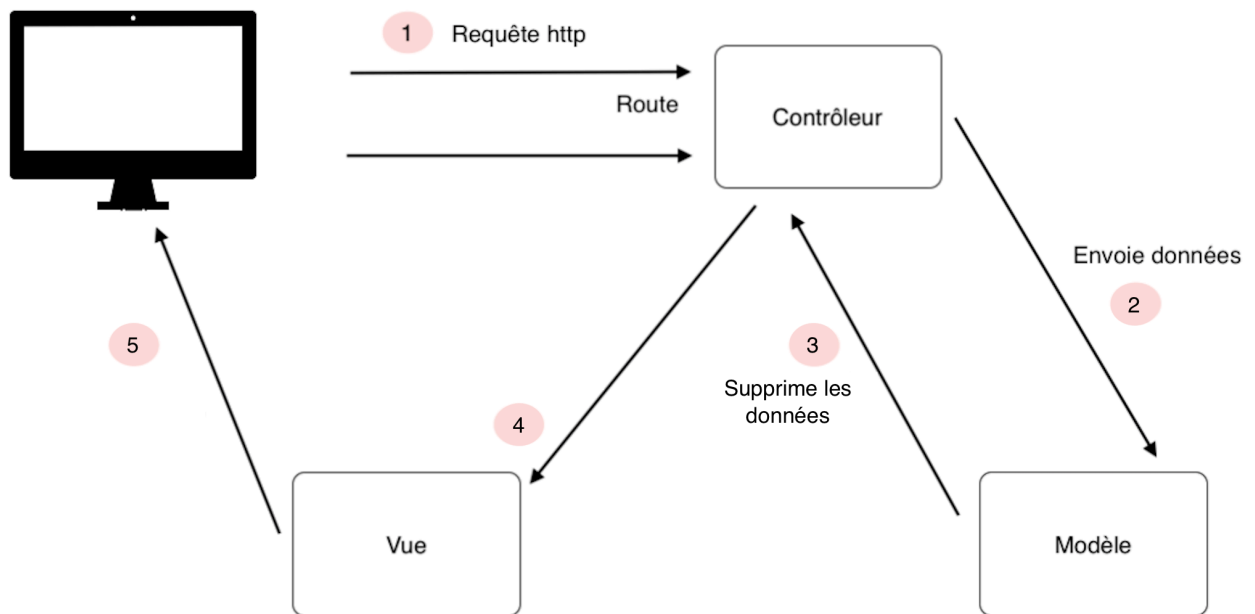
Ce que nous avons besoin : UserAdminController.php, admin/user/list.html.twig, App\Entity\User.php, App\Repository\UserRepository.php.



Supprimer :

Route : /admin/user/{id}/remove

Ce que nous avons besoin : UserAdminController.php, praticien/list.html.twig, App\Entity\User.php, EntityManagerInterface.php, Doctrine.

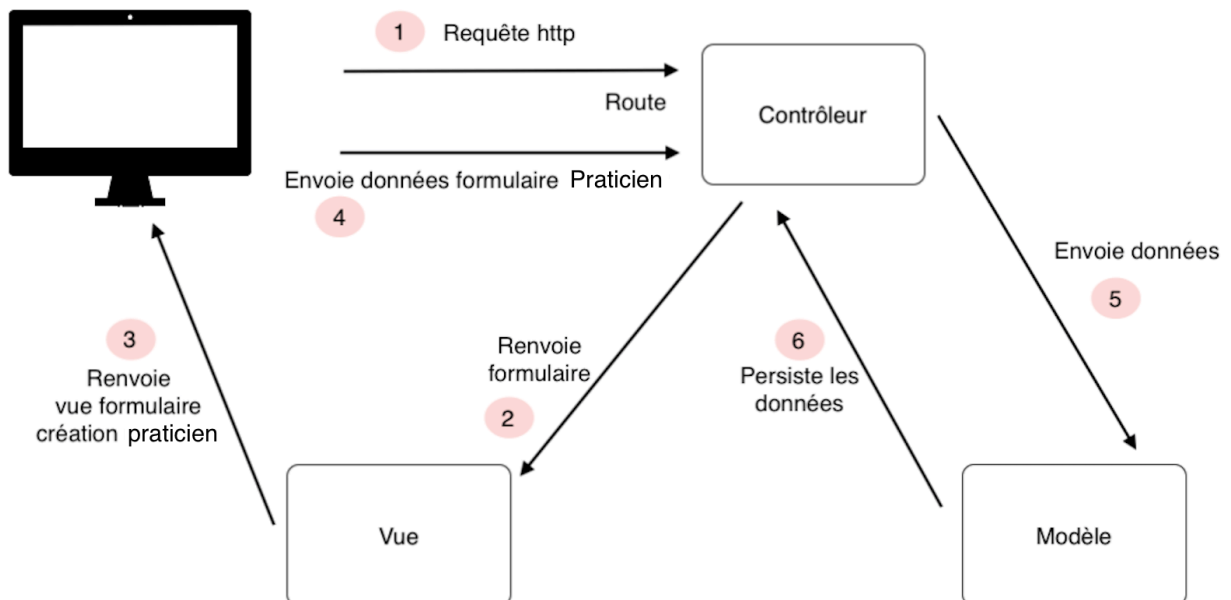


Praticien

Créer :

Route : /praticien/new

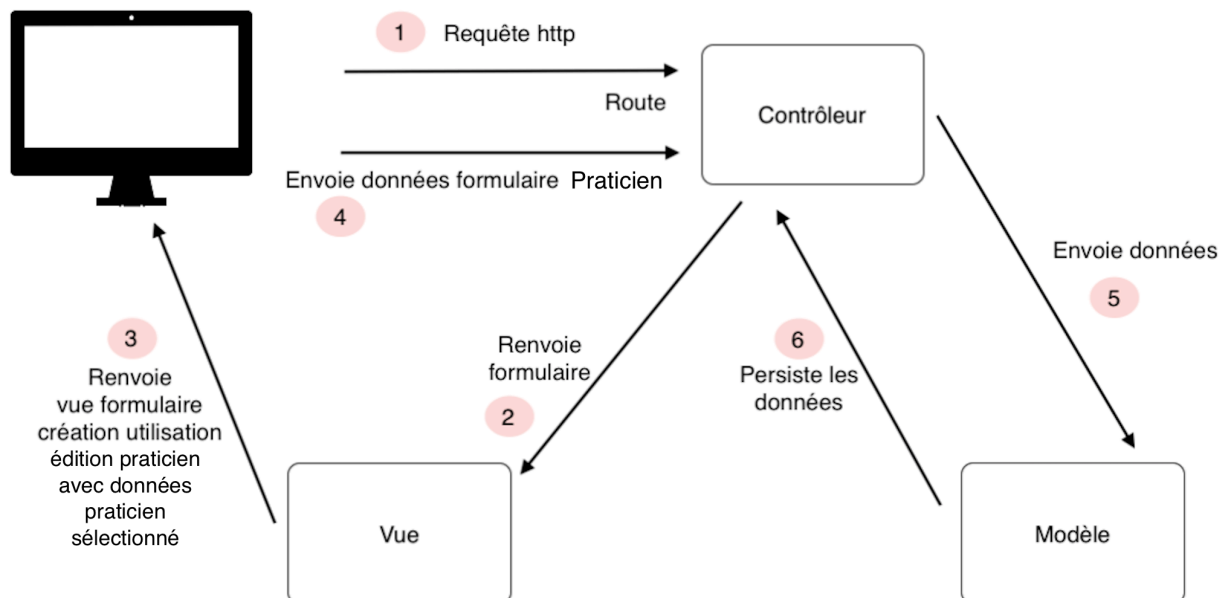
Ce que nous avons besoin : *PraticienController.php*, *praticien/new.html.twig*, *App\Entity\Praticien.php*, *App\Form\PraticienType.php*, *EntityManagerInterface.php*, *Doctrine*.



Éditer :

Route : /praticien/{id}/edit

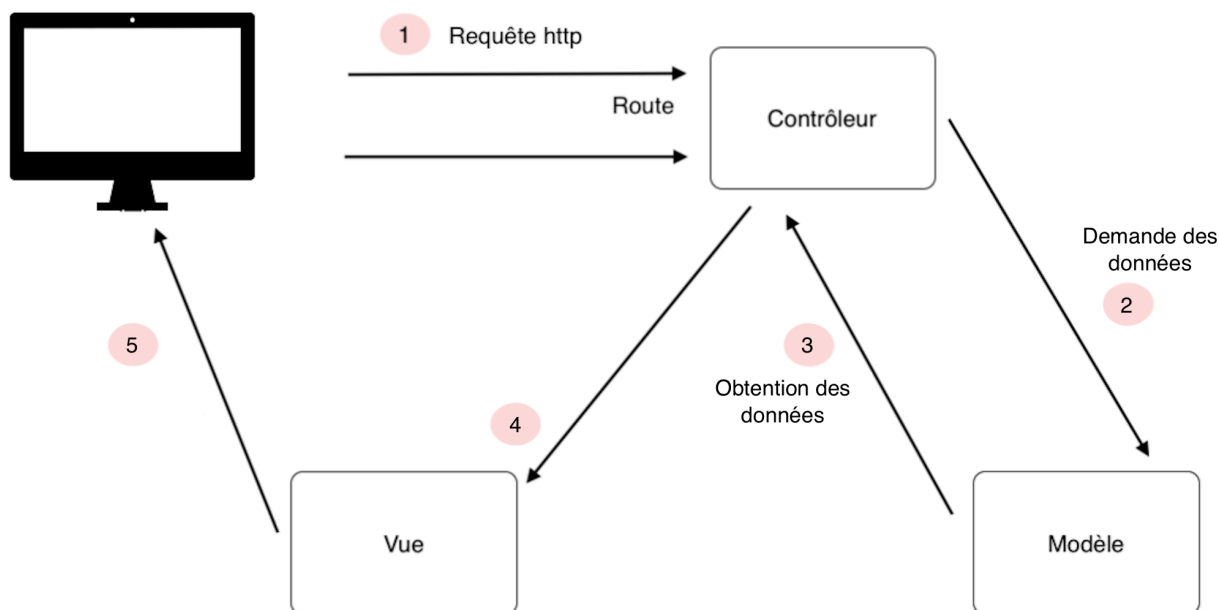
Ce que nous avons besoin : *PraticienController.php*, *praticien/edit.html.twig*, *App\Entity\Praticien.php*, *App\Form\PraticienType.php*, *EntityManagerInterface.php*, *Doctrine*.



Lister :

Route : /praticien

Ce que nous avons besoin : PraticienController.php, praticien/index.html.twig, App\Entity\Praticien.php, App\Repository\PraticienRepository.php.



Supprimer :

Route : /praticien/{id}

Ce que nous avons besoin : PraticienController.php, praticien/index.html.twig, App\Entity\Praticien.php, EntityManagerInterface.php, Doctrine.

