

TP3

Gestion de la couche d'authentification

Afin de sécuriser votre application, l'objectif est d'ajouter une couche d'authentification via formulaire.

Symfony vous met à disposition le nécessaire pour gérer toute votre couche d'authentification grâce au composant Security.

<https://symfony.com/doc/current/components/security/authentication.html>

Vous gèrerez 3 types d'utilisateur :

- ANONYME

Routes	Verbes
/login	GET, POST
/password-resetting	GET, POST
/advertisements	GET
/categories/{id}/advertisements	GET
/advertisements/{id}	GET

- UTILISATEUR AUTHENTIFIE

Routes	Verbes
/advertisements/create	GET, POST
/advertisements/{id}	DELETE, PUT
/logout	GET, POST

- ADMINISTRATEUR

Routes	Verbes
/categories/create	GET, POST
/categories/{id}	DELETE, PUT

Formulaire d'authentification

Utilisez le composant Security pour mettre en place le formulaire d'authentification.

Vous aurez besoin d'une entité User pour pouvoir gérer vos utilisateurs en base de données.

Pensez à lier votre entité User à votre entité Advertisement.

Récupération de mot de passe

Afin de permettre à l'utilisateur de réinitialiser son mot de passe en cas d'oubli, implémenter une fonctionnalité de réinitialisation de mot de passe par email. A vous de voir si Symfony vous permet de le faire ou si vous avez besoin d'un Bundle supplémentaire.

Pensez à configurer l'envoi d'emails pour que la récupération de mot de passe fonctionne.

Lors des développements, il peut être utile d'intercepter les envois d'emails. Vous pourrez vous servir de [MailCatcher](#) pour ce point.

Déconnexion

Pensez à permettre à vos utilisateurs connectés de pouvoir se déconnecter.

Testing

Puisqu'un projet ne fonctionne que s'il est testé, mettez en place des tests sur votre projet.

Afin de tester fonctionnellement votre application, mettez en place des smoking tests (test du code http de toutes les pages de votre application).

Aller plus loin

Si vous avez fait toutes les parties précédentes des TP, vous pouvez aller plus loin :

- Mettre en place des tests fonctionnels (CasperJS, Selenium...)
- Exposer la liste des annonces via une API REST
- Ajouter un compteur de vues sur chaque annonce
- Gérer du cache sur la liste des annonces
- Créer une commande qui désactive les annonces de plus de 1 mois.
- Afficher une carte basée sur le code postal sur les annonces