

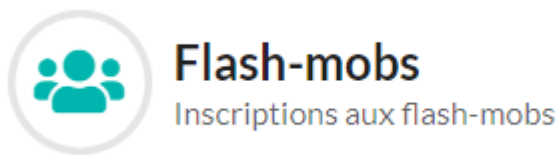
Initialisation

- Créer un projet nommé **tp4** dans votre dossier **web-dyna-td1**.
- Importer la base de données **flash-mobs** (script *flash-mobs.sql*) et générer les classes (models).

Notes :

Pour la génération des liens hypertextes, on utilisera impérativement le nom (name) donné aux routes.

Intégrer le header suivant sur toutes les vues :



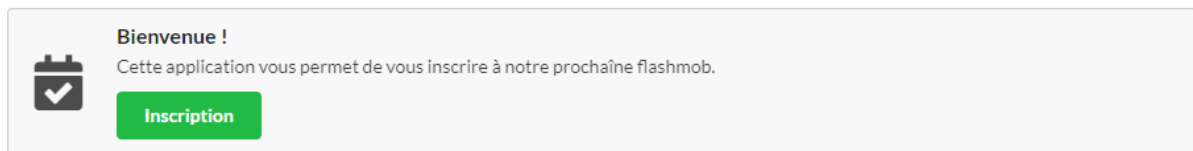
Indications : utiliser les classes ui container, ui segment et ui header de semantic-ui

Créer un contrôleur InscriptionsController

I. Page par défaut (index) :

Faire de l'action **index** du contrôleur **InscriptionsController** la page par défaut de l'application.

La page par défaut doit afficher le message suivant dans une vue :



Le bouton inscription est un lien vers </inscription>

Indications : utiliser les classes ui message et ui button de semantic-ui

II. Formulaire d'inscription :

A l'adresse </inscription>, afficher le formulaire suivant :

| | |
|--|--|
| Nom ou pseudo | Adresse mail |
| <input type="text" value="Votre nom"/> | <input type="text" value="Votre email"/> |
| Valider l'inscription | |

Le formulaire sera posté à l'adresse </do/inscription>

Tous les champs sont obligatoires.

Indications : utiliser les classes ui form et ui input de semantic-ui

III. Envoi de l'inscription

Les données récupérées du post du formulaire doivent permettre d'insérer une instance d'Inscription dans la base de données.

- En cas de succès :
 - Enregistrer dans le cookie de nom **inscription** l'adresse email saisie.
 - Retourner à la page d'accueil (/)
- En cas d'échec, afficher le message :







Inscription
Votre inscription avec l'adresse email `evanyou@vuejs.net` n'a pu être réalisée !

Ré-essayer

Le bouton ré-essayer renvoie vers la route </inscription>

IV. Liste des inscrits

A l'adresse </inscriptions>, afficher la liste des inscrits (issus de la table **inscriptions** de la base de données)

| Nom | Email | Date & heure |
|---------------|--|---------------------|
| Olaf |  olaf@gmail.com | 2020-05-10 12:38:22 |
| YOU Evan |  evanyou@vuejs.net | 2020-05-10 15:52:57 |
| Olive |  olive@local.fr | 2020-05-10 16:06:13 |
| Sherman Benji |  benjamin.sherman@test.fr | 2020-05-10 16:31:20 |


Retour à l'accueil

Indications : utiliser les classes [ui table](#) et [ui label](#) de semantic-ui

V. Prise en compte de l'inscription et déconnexion

En cas d'inscription déjà réalisée (Cookie présent), la page d'accueil doit maintenant afficher le message :

Done !
Vous êtes déjà inscrit à notre prochaine flashmob avec l'email `benjamin.sherman@test.fr`.

 [Consulter les inscriptions](#) [Se déconnecter](#)

Le bouton **Se déconnecter** renvoie à l'adresse </logout> qui permet de supprimer le cookie **inscription** et de retourner à la page d'accueil.

Indications

On pourra utiliser les classes et méthodes suivantes :

DAO

- `insert(object $instance): boolean`
Insère l'objet \$instance dans la base de donnée, retourne faux en cas d'échec

UCookie

- `exists(string $name): boolean`
Détermine si le cookie de nom \$name existe
- `delete(string $name):boolean`
Supprime le cookie de nom \$name
- `get(string $name):string`
retourne la valeur du cookie de nom \$name

URequest

- `setValuesToObject(object $o)`
Affecte membre à membre les valeurs du POST à l'objet \$o

UResponse

- `header('location', string $path)`
Effectue une redirection vers l'url \$path

Exemple :

// Retour à la page d'accueil

`UResponse::header("location" , "/");`

Routes

| Path | Methods | Action & parameters | Cache | Expired | Name |
|---|---------|------------------------|--------------------------|---------|----------------|
| ♥ controllers\InscriptionsController::class | | | | | |
| /_default/ | | index () | <input type="checkbox"/> | | index |
| /inscription/ | get | inscription () | <input type="checkbox"/> | | inscription |
| /inscriptions/ | get | inscriptionsList () | <input type="checkbox"/> | | inscriptions |
| /logout/ | get | logout () | <input type="checkbox"/> | | logout |
| /do/inscription/ | post | validateInscription () | <input type="checkbox"/> | | do.inscription |

Rendu

- Dossier **app** de votre projet **tp4** à zipper,
- à renommer selon le modèle :
TD{num}-{prenom}.{nom}.zip
- et à déposer sur **Ecampus** en fin de séance impérativement.
- 2^{ème} rendu facultatif avant minuit.

Commits

Commits à effectuer obligatoirement toutes les ½ heure

Ne pas faire de push avant le second rendu

La régularité des commits atteste que votre travail est original, et que vous n'avez pas recopié celui d'un autre.