

Rapport Projet Web

Application

Système d'évènements social géo localisés

BA Mamadou Lamine



2013

Sommaire

Objectifs.....	3
Fichiers et dossiers du projet.....	3
Fichier java	3
Fichiers JSP	4
Fonctionnalités implémentées.....	5
I. Ajout d'évènements.....	5
Etat.....	5
Traitement.....	5
II. Récupération des stadium de l'open data Nantes.....	5
Etat	5
Dataset utilise pour les stadium	5
Classe de traitement.....	5
Champs utilises du dataset	5
III. Envoi de mails.....	6
Etat.....	6
Apres ajout d'un événement : sendEmailNewEvent ()	6
Nouvelle souscription a un événement : sendEmailPormotorNewSubcribtion ()	6
Confirmation souscription a un événement : sendEmailSubcribtion ().....	6
Confirmation newsletter : sendEmailNewsletter ()	6

Objectifs

The project aims to build a geolocalised social event system. The use case is the following:

- A user post a public event: "we will play soccer tomorrow in public stadium near la beaujoire, we want 5 people more"
- The stadium is referenced in local open data,
- Registered members interested by soccer in this area receive a notification and can notify sender of its intention.

Fichiers et dossiers du projet

Fichier java

Package	Fichiers	Traitements	Remarques
fr.nantes.event.bean Package des beans	EventBean.java	Bean correspondant a un événement	
	UserBean.java	Bean pour utilisateur (membre)	
fr.nantes.event.controller Package des controllers (Traiteurs d'actions)	Controller.java	Controller de base	Les autres controllers héritent de celui la
	Event.java	Controller qui traitement les actions sur les événements (ajout, détails, souscription, désinscription)	
	Index.java	Controller qui correspond a la page d'accueil du site	Traite les actions du formulaire de recherche a gauche du site
	User.java	Traites les éventuelles actions sur les utilisateurs	
fr.nantes.event.dao Contient les classe @PersistenceCapable c'est-à-dire les POJOs	EventDao.java	Classe qui correspond a l'entité événement du datastore	Classe est @PersistenceCapable
	GuestDao.java	Correspond a la table des souscriptions d'évènements	
	NewsletterDao.java	Entité qui correspond a une newsletter	
	UserDao.java	Correspond a l'entité User du datastore	

fr.nantes.event.servlet Package des servlets	Dispatcher.java	Servlet de base qui reçoit tous les actions (« .do »)	Elle passe le traitement au controller approprié
fr.nantes.event.util Package des classes utiles	EventUtility.java	Classe qui implémente les méthodes utiles pour les événements (recherche, suppression, souscription, désinscription,...)	
	Initialization.java	Classe d'initialisation qui récupère l'url en cours et charge les données utiles (Stadium, sports, ...)	
	PMF.java	Classe d'accès au datastore pour les persistances, récupération, suppression sur le datastore	
	SendEmail.java	Classe qui implémente les différents envois de mails	
	UserUtility.java	Contient les méthodes utiles pour l'utilisateur (insertion si nouveau, récupération, ...)	
	Util.java	Méthodes de conversion utiles	
	Utility.java	Méthodes utiles pour les dates, et autres	
	XmlParser.java	Classe qui implémente les méthodes pour aller récupérer les stadium du dataset	

Fichiers JSP

Dossiers	Fichier	Traitements	
war/	addEvent.jsp	Affiche le formulaire pour ajouter un événement	L'action redirige vers /event/add.do
	banner.jsp	jsp charge d'afficher la bannière	
	contacts.jsp	affiche la page « Contacts »	
	detailsEvent.jsp	Détails d'un événement	Passe par l'action /event/details.do
	footer.jsp	Afficher le pied de page	
	header.jsp	Contient les balises « meta »	Vérifie à chaque chargement l'utilisateur
	index.jsp	Page d'accueil fait appel à la méthode pour afficher les événements à venir	

	left.jsp	Page de gauche	
	members.jsp	Affiche la liste des membres du système	
	menu-search.jsp	Affiche le menu	
	myEspace.jsp	Correspond a « My Espace »	
	right.jsp	Partie droite du site	

Fonctionnalités implémentées

I. Ajout d'évènements

Etat

Cette fonction marche correctement

Il faut être connecté pour ajouter un événement. Pour ce faire, il faut cliquer sur le bouton à gauche « Add new event ».

Formulaire : <http://localhost:8888/addEvent.jsp>

Action : <http://localhost:8888/event/add.do>

Traitement

Les urls terminées par « .do » passent tous par le : `fr.nantes.event.servlet.Dispatcher` qui, selon la valeur du « `getServletPath` », fait appel au contrôleur approprié.

II. Récupération des stadiums de l'open data Nantes

Etat

Cette fonctionnalité est implémentée et marche.

Dataset utilisé pour les stadiums

http://data.nantes.fr/api/publication/24440040400129_NM_NM_00024/LOC_EQUIPUB_SPORT_NM_STBL/content/?format=xml

Classe de traitement

La classe `fr.nantes.event.util.XmlParser.java` permet de faire le fetching des données reçues de l'URL.

Champs utilisés du dataset

Champs	Utilisation
name	Représente le nom affiché sur la liste déroulante

ADRESSE	l'adresse du stadium
_l	De la forme [47.1929657069178, -1.53691739665354]

III. Envoi de mails

Etat

Cette fonctionnalité marche

Tous les envois de mails se font par la classe **fr.nantes.event.util.SendEmail.java**

Après ajout d'un événement : **sendEmailNewEvent ()**

Un mail est envoyé a tous les utilisateurs du système et ceux qui sont souscrits à la newsletter.

Nouvelle souscription a un événement : **sendEmailPormotorNewSubcription ()**

Un mail est envoyé au propriétaire de l'événement pour lui signaler une nouvelle souscription.

Confirmation souscription a un événement : **sendEmailSubcription ()**

Mail envoyé à celui qui s'est souscrit a l'événement pour lui confirmer sa souscription.

Confirmation newsletter : **sendEmailNewsletter ()**

Mail envoyé après qu'une personne se soit inscrite a la newsletter a droite du site