

Bilel BAKINI
Hamza BEY
Paolo DE VATHAIRE
Stéven BERNARD

13/02/2013

ComTrek (Licence GPL)

Spécifications organisationnels et techniques

ComTrek ?

ComTrek est un site permettant de fédérer une communauté de passionnés de la montagne plus précisément de trek. Le site ComTrek sera composé des grands domaines fonctionnels suivants :

- La gestion des membres de la communauté : inscriptions, modifications et suppression des membres de la communauté (CRUD).
- La gestion des trek : ajout, modification, suppression des trek (CRUD) ; les membres de la communauté peuvent enregistrer et présenter de nouveaux trek, ils peuvent, pour chaque trek, déposer leur propre évaluation du trek (temps, difficulté).
- la gestion des retour de trek : chaque membre peut sur un trek donné, publier une note et des commentaires, ainsi que les étapes atteintes et le temps effectif. Il est ensuite possible de comparer ces notes et temps avec ceux des même treks.

Si le temps le permet : gestion de photos, organisation complète du trek (formation de groupe, gestion de matériel,...)

Equipe à l'origine de ComTrek

Stéven BERNARD : chef de projet
Paolo DE VATHAIRE : responsable infrastructure
Hamza BEY : développeur / testeur
Bilel BAKINI : développeur / testeur

Planification

L'équipe de projet habitué aux méthodes agiles et notamment Scrum prévoit un versionnement de l'application à chaque sprint réalisé. Pour débiter le projet, un sprint d'initialisation ou sprint

0 (d'une durée non figé à l'avance) a été réaliser pour établir une base d'outillage à utiliser et homogénéiser l'environnement des développeur.

Organisation du code et de la documentation

Un repository GIT rend le code source et toute les versions finales de la documentation du projet accessibles à l'adresse suivante : <https://github.com/bakinibilel/comtrek>

Pour chaque nouvelle version, une branche sera créé correspondant à un sprint (hors sprint d'initialisation).

De plus, pour faciliter la rédaction des documents dans leur version non final, un répertoire est accessible sur Google Drive. Ces documents doivent permettre à tout developpeur de comprendre facilement l'organisation du projet et les outils à utiliser.

Environnement de développement

Nous conseillons vivement d'utiliser EclipseSTS avec le framework Grails 2.1.2. Pour commencer à développer :

- importer le projet Comtrek à partir de GIT en dehors d'Eclipse
- importer ComTrek en tant que projet Grails dans Eclipse

A noter :

- Si le projet grails dans eclipse apparait de manière inhabituelle ou non conforme, il se peut que votre workspace soit corrompu. Il est conseiller d'utiliser un nouveau workspace.
- Si certains répertoires de votre projet grails n'apparait pas, comme les services ou les taglib, c'est qu'il faut les rajouter dans le "Java Build Path".

Gestion de la licence

L'application est sous licence GPL, pour faciliter l'intégration de cette licence dans le code sources, un plugin eclipse est à utiliser : http://www.wdev91.com/?p=cpw_ug

Les instructions sont assez facile à suivre. A noter tout de même qu'il faut bien faire attention à paramétrer le format d'en-tête pour chaque type de fichier, afin que la licence apparaisse sous forme de commentaire.

Gestions des tests

Spocks

Spock est un framework de test et de spécification, il est compatible avec la plupart des environnements de développement, les outils de build et les serveurs d'intégration continue.

Il est similaire à JUnit mais avec une meilleure lisibilité grâce à son bloque pour faire du Data Driven Test (when, then, where). Cela permet de faire des tests plus facilement évolutifs et qui s'intéressent plus à l'interaction entre les objets. Cette approche permet de tester que la collaboration entre les objets s'effectue comme attendue.

Le plugin Spock 0.7 est déjà intégré sur la version 2.1.2 de Grails.

Selenium

Selenium est un outil (plugin Firefox) qui permet de réaliser des tests fonctionnels sur l'application. Il peut être simplement enregistré pour être rejoué plus tard. Ou de manière plus complexe décrit manuellement. On privilégiera la première méthode.

Afin de s'assurer de la bonne réussite de ces tests fonctionnels, il faut que le script de création de la base de données ait été joué pour avoir la structure et les données minimum qui sont un pré-requis.

Gestion de l'intégration continue

Pour ce faire nous utilisons Jenkins via Cloudbees. Pour le paramétrage, vous pourrez y accéder à l'URL suivante : <https://comtrek.ci.cloudbees.com/>. Néanmoins il vous faudra demander un accès à l'équipe de pilotage.

Le site en production est accessible ici : <http://comtrek.comtrek.cloudbees.net>

Configuration Jenkins

On a six jobs qui sont configurés dans Jenkins:

- [comtrek-launch unit tests](#): Un job est lancé à chaque push sur le repo GitHub. Il consiste à récupérer les changements et lancer les tests unitaires.
- [comtrek-launch tests then build](#): Un autre est lancé tous les soirs à minuit. Il récupère les changements sur GitHub et lance tous les tests.
- [comtrek-build and deploy](#): S'ils réussissent, le job suivant est lancé qui compile l'application et construit le WAR et le déploie en production.
- [comtrek-generate pom](#): Le job suivant construit le pom.xml.
- [comtrek - deploy on maven](#): Ce job déploie le war précédemment généré sur le repository maven grâce au pom déjà construit dans le job précédent.
- [comtrek-build war](#): Il existe un dernier job que l'on peut lancer manuellement pour récupérer les changements sur github construire le war et le déployer en production.

Note : le déploiement du war sur le repository maven de cloudbees est effectué grâce à l'exécution d'un script shell qui copie le fichier pom.xml créé dans le job précédant dans le répertoire du job Maven.

Analyse de code

L'analyse de couverture de tests est effectué avec Cobertura. Tandis que L'analyse de code est réalisé avec CodeNarc. Tout deux seront consultables sur Jenkins.

Architecture applicative

L'application suis une architecture Grails classique.

Voici le modèle du domaine métier :

