



zenika
ARCHITECTURE INFORMATIQUE

www.zenika.com

Compass

Optimiser les recherches avec Compass

Objectifs

- Utiliser Compass pour développer rapidement et efficacement une solution de moteur de recherche pour toute application et pour tout type de document indexable
- Simplifier et optimiser les recherches à l'aide des fonctionnalités offertes par Compass
- Comprendre l'intégration avec de nombreux frameworks ainsi que les services proposés
- Bénéficier de la puissance de Lucene avec une API haut-niveau

Contenu

Introduction

- Compass : une API basée sur Lucene qui permet un accès haut-niveau à la création et maintenance d'index pour une recherche optimisée et fiable
- Comment Compass simplifie l'utilisation du puissant Lucene

Les bases

- Lucene : la base de Compass
- Compass - CompassSession - CompassTransaction : 3 éléments centraux pour utiliser Compass

Profiter facilement de Lucene

- Compass se base sur Lucene qui est un moteur d'indexation de documents
- Fourniture d'une API de recherche plein-texte dans ces index
- Nombreuses interfaces afin de créer des index facilement à partir de toutes sortes de documents (y compris des objets)
- Création de recherches élaborées

Intégration complète avec des frameworks

courants

- Application et web : Spring et Spring MVC
- ORM : Hibernate, JDO, iBatis...
- Support de XML pour une intégration plus large...
- Développer une application Spring qui utilise Compass pour faire une recherche dans des objets construits par Hibernate

Une vision transactionnelle

- Création des index de recherche basée sur les transactions Lucene et qui peut s'appuyer sur plusieurs moteurs transactionnels (JTA, Local, XA)
- Voir comment Compass utilise les transactions pour éviter la corruption des index

Performance

- Compass s'intègre dans des outils de DataGrid (Gigaspaces, Coherence, Terracotta) pour simplifier la recherche dans des index distribués
- Utiliser Compass dans un « cluster » Terracotta

Un découpage simple

- Core :
 - Configuration (des index)
 - Connection (aux index qui peuvent être divers)
 - Search Engine (pour la recherche en tant que telle)
 - xSEM (pour divers mapping de documents vers un moteur de recherche)
- GPS : Pour l'intégration avec les bases de données et les frameworks ORM
- Spring : pour l'intégration avec le framework du même nom
- Needle : pour le déploiement et le questionnement des index sur des clusters

Les fonctionnalités de bases

- Créer un index à partir des documents (index local, distant, distribué)
- Tous les documents contenant du texte indexable sont manipulable (XML, HTML, Word, Excel...)
- Effectuer une recherche simplement et proprement (avec des classes de haut-niveau)
- Gérer les index pour permettre une mise à jour optimale sans trop dégrader les performances de recherche (utilisation des transactions)

Durée

2 jours

Répartition

50% théorie

50% pratique

Pré-requis

Connaissance de Java

Public

Architecte

Développeur

Chef de projet

Tarif (déjeuners inclus)

1200 € (HT)

Tarif (1 mois avant)

1100 € (HT)

Lieu

Paris 9ème

Sessions 2009

5 au 6 février

24 au 25 mars

29 au 30 juin

17 au 18 septembre

5 au 6 novembre

Intra-entreprise sur demande

Inscription et renseignements

+33(0)1.45.26.19.15

training@zenika.com

www.zenika.com