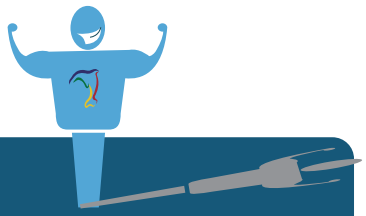
autentia

Avenida de Castilla,1 - Edificio Best Point - Oficina 21B
28830 San Fernando de Henares (Madrid)
tel./fax: +34 91 675 33 06
info@autentia.com - **www.autentia.com**

¿**Qué ofrece** Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

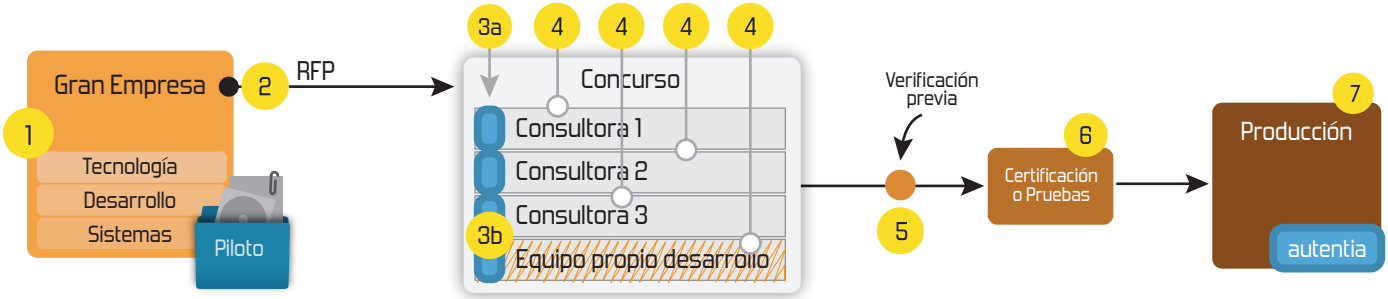
1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

- Spring MVC, JSF-PrimeFaces /RichFaces, HTML5, CSS3, JavaScript-jQuery
- Gestor portales (Liferay)
- Gestor de contenidos (Alfresco)
- Aplicaciones híbridas
- Control de autenticación y acceso (Spring Security)
- UDDI
- Web Services
- Rest Services
- Social SSO
- SSO (Cas)
- JPA-Hibernate, MyBatis
- Motor de búsqueda empresarial (Solr)
- ETL (Talend)
- Dirección de Proyectos Informáticos.
- Metodologías ágiles
- Patrones de diseño
- TDD
- Tareas programadas (Quartz)
- Gestor documental (Alfresco)
- Inversión de control (Spring)
- BPM (jBPM o Bonita)
- Generación de informes (JasperReport)
- ESB (Open ESB)

Compartimos nuestro conocimiento en:
www.adictosaltrabajo.com

Para más información visítenos en:
www.autentia.com

Powered by autentia Hosting patrocinado por enREDados

NUEVO ¿Quieres saber cuánto ganas en relación al mercado? pincha aquí...

Ver cursos que ofrece Autentia

Descargar comics en PDF y alta resolución

[¡NUEVO!] 2008-07-31

2008-07-08

2008-06-22

2008-06-09

Estamos escribiendo un libro sobre la profesión informática y estas viñetas formarán parte de él. Puedes opinar en la seccion comic.

Tutorial desarrollado por

Carlos García Pérez

Creador del pionero Web MobileTest.

Consultor tecnológico en el desarrollo de proyectos informáticos.

Ingeniero Técnico en Informática *

Puedes encontrarme en Autentia

Somos expertos en Java/J2EE

Catálogo de servicios de Autentia

Descargar (6,2 MB)

Descargar en versión comic (17 MB)

AdictosAlTrabajo.com es el Web de difusión de conocimiento de Autentia.

autentia real business solutions

Catálogo de cursos

Descargar este documento en formato PDF: lucene_intro.pdf

Fecha de creación del tutorial: 2008-08-17

Introdución a Lucene

En este tutorial, vamos a realizar una breve presentación del API de búsqueda Lucene y realizaremos un sencillo y completo ejemplo.

Lucene es una librería (no un producto) open source (con licencia de Apache) con que permite a los desarrolladores (Java, .Net, C++, et.) integrar funciones de indexación y búsquedas de información textual dentro de sus proyectos.

Lucene va mucho más alla que las búsquedas en BD a través de índices FullIndex, permitiéndonos indexar y realizar búsquedas sobre todo **tipo de información que pueda representarse de forma textual**.

Lucene se compone de dos procesos o fases para conseguir realizar búsquedas efectivas: Indexación y Búsqueda.

Indexación:

El proceso de indexación consiste en analizar y extraer de entre toda la dispobible, la verdaderamente relevante. Posteriormente, con esa información se crea el índice a partir del cual posteriormente se basarán las búsquedas. El índice es una estructura de datos que permite acceso rápido a la información, algo similar semánticamente a lo que podría ser el índice de un libro.

Búsqueda:

El proceso de búsqueda consiste en consultar el índice para obtener los documentos donde aparecen unas determinadas palabras o bien concuerdan con una determinada expresión de consulta.

Un ejemplo

A continuación vamos a ver un sencillo ejemplo de indexación (`LuceneCreateIndexApp.java`) y consulta (`LuceneSearchApp.java`) sobre los archivos con extensión `.txt` de un directorio.

Indexación de información: `LuceneCreateIndexApp.java`

Catálogo de servicios Autentia (PDF 6,2MB)

En formato comic...

Google

Web

www.adictosaltrabajo.com

Buscar

Últimos tutoriales

2008-08-17
Introducción a Lucene

2008-08-14
Uso de Filtros en Hibernate

2008-08-13
Modelaje de figuras de Poliuretano

2008-08-09
Instalar Internet Explorer 6 en Windows XP y Vista

2008-07-28
Acceso de escritura a las particiones de NTFS de Windows desde GNU/Linux Debian

2008-07-22
Introducción a JSON

2008-07-17
¿Cómo cambiar el nivel de log en tiempo de ejecución?

2008-07-14
Depurar en PHP: Eclipse PDT + Xdebug

2008-07-11
Triggers o Disparadores SQL

2008-07-05
Publicar una nueva release en SourceForge.net mediante SFTP, WebDav o rsync sobre SSH

Últimas ofertas de empleo

2008-08-11
Atención a cliente - Call Center - MADRID.

2008-07-28
Comercial - Ventas - MALAGA.

2008-07-11
Comercial - Ventas - MADRID.

2008-07-04
T. Información - Analista / Programador - MADRID.

2008-06-26
Comercial - Ventas - BARCELONA.

Anuncios Google
Empleo Java
Informatica Facil
Web XML Java
Tutorial Photoshop

view plain print ?

```java
package com.autentia.tutoriales.lucene;

import org.apache.lucene.index.IndexWriter;
import org.apache.lucene.analysis.standard.StandardAnalyzer;
import org.apache.lucene.document.Document;
import org.apache.lucene.document.Field;

import java.io.*;

/**
 * El siguiente ejemplo crea un índice a partir de todos los documentos con extensión .txt de un director
 * @author Carlos García. Autentia.
 * @see http://www.mobiletest.es
 */
public class LuceneCreateIndexApp {

    /**
      * Punto de entrada a la aplicación
      */
    public static void main(String[] args)  {
      IndexWriter   writer   = null;
      File          dir      = new File("c:/lucene_dir_to_index");
      Document      doc      = null;

      try {
          // Creamos el indice si no existe almacenandolo en un directorio del HDD
          writer = new IndexWriter(new File("c:/lucene_output_dir_index"), new StandardAnalyzer(), true);

          // Recorremos todos los archivos del directorio (No sus subdirectorios)
          File[] files = dir.listFiles();
          for (File file : files){
                if (file.isFile() && file.canRead() && file.getName().endsWith(".txt")) {
                    System.out.println("Indexando el archivo: " + file.getAbsolutePath());

                    // Cremos e inicializamos el Document con los datos del archivo que queremos guardar
                    doc = new Document();
                    doc.add(new Field("path",     file.getPath(), Field.Store.YES, Field.Index.UN_TOKENIZED));
                    doc.add(new Field("contents", new FileReader(file)));

                    // Añadimos el documento al indice.
                    writer.addDocument(doc);
                }
          }

          // Organizamos y guardamos los datos.
          writer.optimize();
          writer.close();


          System.out.println("OK");
       } catch (Exception ex){
          System.out.println(ex);
       }
    }
}
```

Introducción a las clases más comunes durante el proceso de indexación.

- **`IndexWriter:`**

  Es el corazón del proceso de indexación. Encapsula el índice y contiene entre otras cosas cierta información de documentos que lo componen.

- **`Directory`**

  Es una clase abstracta que representa la ubicación del índice: memoria RAM, archivo, etc. En el ejemplo se usa FSDirectory, que es la implementación de la clase para guardar el índice en un archivo.

- **`Analizer`**

  Es una clase Abstracta cuya función es filtrar la información relevante con la cual se creará el índice y eliminar el resto. Por ejemplo, elimiminar las palabras cortas (stop word) o que aparecen con mucha frecuencia. Por ejemplo, eliminaría palabras (dependiendo del idioma) del tipo: en, la, los, etc.

- **`Document`**

  Es una clase que almacena campos (Field) con la meta información que queremos indexar y/o guardar en el índice (emails, rutas, autor, fecha, etc.).

- **`Field`**

  Son las fragmentos de información de cada documento que deseamos almacenar en el Document.

Búsqueda de información: **`LuceneSearchApp.java`**

view plain print ?

```java
package com.autentia.tutoriales.lucene;

import org.apache.lucene.document.Document;
import org.apache.lucene.search.*;
import org.apache.lucene.store.*;
import org.apache.lucene.queryParser.QueryParser;
import org.apache.lucene.analysis.standard.StandardAnalyzer;
import java.io.File;

/**
 * El siguiente ejemplo realiza una búsqueda sobre el índice
 * @author Carlos García. Autentia.
 * @see http://www.mobiletest.es
 */
public class LuceneSearchApp {
    /**
     * Punto de entrada a la aplicación
     */
    public static void main(String[] args) {
        try {
            // Ubicación del índice
            FSDirectory     directory = FSDirectory.getDirectory(new File("c:/lucene_output_dir_index"));

            // Creamos un búscador
            IndexSearcher   searcher  = new IndexSearcher(directory);


            // Palaba a búscar dentro del índice aquellos documentos que contentan la palabra "Carlos" y n
            String          textToSearch = "+Carlos -Avanzado";

            // Creamos la consulta
            QueryParser     parser    = new QueryParser("contents", new StandardAnalyzer());
            Query           query     = parser.parse(textToSearch);

            // Realizamos la búsqueda
            Hits            hits      = searcher.search(query);
            Document        doc       = null;

            // Mostramos los resultados
            if (hits.length() == 0){
                System.out.println("No se han encontrado coincidencias.");
            } else {
                for (int i = 0; i < hits.length(); i++) {
                    doc = hits.doc(i);
                    System.out.println(doc.get("path"));
                }
            }
        } catch (Exception ex){
            System.out.println(ex);
        }
    }
}
```

Introducción a las clases más comunes durante el proceso de búsqueda.

- **`IndexSeacher:`**

  Es el corazón del proceso de búsqueda a través del cual se pueden realizar consultas (Query) que devuelven resultados (Hits).

- **`Term:`**

  Cuando querermos realizar una consulta definimos las condiciones de búsqueda a través de Term, es decir, una consulta es un conjunto de Term.

- **`Query:`**

  Es una clase abstracta que permite definir la consulta sobre la que queremos realizar la búsqueda. TermQuery, es una de las muchas implementaciones de Query y que nos permite búscar consultando los Field.

- **`Hits:`**

  Contiene los documentos (Document) que cumplen las condiciones de la búsqueda realiza.

Conclusiones

Bueno, para terminar cabe destacar que Lucene es mucho más de lo que se ve en este tutorial introductorio, si tienes inquietud de aprender más o necesitas más información puedes dirigirte al web oficial del proyecto Lucene.

También recordaros que Autentia tiene personal muy experimentado en este campo, así que si algún dia necesitais formación o algún tipo de colaboración podeis poneros en contacto con nosotros.

Un saludo.
Carlos García. Creador de MobileTest.

- Puedes opinar sobre este tutorial haciendo clic aquí.
- Puedes firmar en nuestro libro de visitas haciendo clic aquí.
- Puedes asociarte al grupo AdictosAlTrabajo en XING haciendo clic aquí.
- Añadir a favoritos Technorati. ADD THIS BLOG TO MY Technorati FAVORITES

SOME RIGHTS RESERVED Esta obra está licenciada bajo licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5

Recuerda

Autentia te regala la mayoría del conocimiento aquí compartido (Ver todos los tutoriales). Somos

expertos en: J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ... y muchas otras cosas.

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?, ¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos ...

Autentia = Soporte a Desarrollo & Formación.

info@autentia.com

Servicio de notificaciones:

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales.

Formulario de subcripción a novedades:

E-mail Aceptar

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	pdf
Creación de documentos PDF en sitios web utilizando el componente AspPDF	En este tutorial se muestran las características generales del componente ASP-PDF que permite gestionar documentos PDF desde una aplicación web	2007-03-20	4534	pdf
Lucene: Analyzers, stemming y búsqueda de documentos similares.	En este tutorial vamos a ver cómo implementar un analizador semántico en nuestro idioma, potenciando la indexación y búsqueda, para terminar analizando la viabilidad de realizar búsquedas de documentos similares.	2008-02-22	1841	pdf
Leer un documento de Microsoft Word usando la librería POI de jakarta	En este docuemento aprenderemos a leer un documento de Microsoft Word usando la librería POI de jakarta	2006-09-20	9143	pdf
Gestión documental	Este tutorial nos va a hablar sobre la gestión documental que podemos incluir dentro de nuestras aplicaciones	2007-05-23	3164	pdf
Primeros pasos con el motor de busqueda Java Lucene	Os mostramos los primeros pasos con uno de los más extendidos motores de búsqueda dentro del mundo OpenSource y Java: Tambien veremos lo sencillo que es monitorizar los índices con Luke e indexar PDFs con PDFBox	2006-07-31	7798	pdf
Extracción de texto de documentos Office desde Java	En este tutorial vamos a ver como podemos extraer texto de los documentos de Office (DOC, XLS y PPT) desde Java	2007-05-22	6866	pdf
Framework para indexación de documentos en Lucene: LIUS	En este tutorial vamos a ver como usar el framework LIUS (Lucene Index Update and Search) para indexar documentos en el motor de busqueda textual Lucene	2007-10-23	2759	pdf

Nota:

Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento. Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores. En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo. Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.

Ads by Google View ads about: »

Copyright 2003-2008 © All Rights Reserved | Texto legal y condiciones de uso | Powered by Autentia

XHTML 1.0 CSS XML RSS XML ATOM