



ENVIE DE BOOSTER TA CARRIÈRE ?
OPEN SOURCE-TOI !



IMAGE
GÉNÉRÉE
PAR
SORA



IMAGE
GÉNÉRÉE
PAR
SORA



IMAGE
GÉNÉRÉE
PAR
SORA



2011





Compass is an [open source](#) project built on top of [Lucene](#) aiming at simplifying the integration of [search](#) into any Java application.

The Future of Compass & ElasticSearch

[FRAMES](#) [NO FRAMES](#) [All Classes](#)
[DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)

org.apache.lucene.store.jdbc

Class JdbcDirectory

[java.lang.Object](#)

extended by [org.apache.lucene.store.Directory](#)

extended by [org.apache.lucene.store.jdbc.JdbcDirectory](#)

All Implemented Interfaces:

[MultiDeleteDirectory](#)

```
public class JdbcDirectory
extends Directory
implements MultiDeleteDirectory
```

A Jdbc based implementation of a Lucene Directory allowing the storage of a Lucene index within a database. Uses a jdbc DataSource, [Dialect](#) specific for the database used, and an optional [JdbcDirectorySettings](#) and [JdbcTable](#) for configuration.

The directory works against a single table, where the binary data is stored in Blob. Each "file" has an entry in the database, and different [FileEntryHandler](#) can be defines for different files (or files groups).

Most of the files will not be deleted from the database when the directory delete method is called, but will only be marked to be deleted (see [MarkDeleteFileEntryHandler](#). It is done since other readers or searchers might be working with the database, and still use the files. The ability to purge mark deleted files based on a "delta" is acheived using [deleteMarkDeleted\(\)](#) and [deleteMarkDeleted\(long\)](#). Note, the purging process is not called by the directory code, so it will have to be managed by the application using the jdbc directory.

For transaction management, all the operations performed against the database do not call commit or rollback. They simply open a connection (using [DataSourceUtils.getConnection\(javax.sql.DataSource\)](#)), and close it using [DataSourceUtils.releaseConnection\(java.sql.Connection\)](#). This results in the fact that transaction management is simple and wraps the directory operations, allowing it to span as many operations as needed.

For none managed applications (i.e. applications that do not use JTA or Spring transaction manager), the jdbc directory implementation comes with [TransactionAwareDataSourceProxy](#) which wraps a DataSource (should be a pooled one, like Jakarta DBCP). Using it with the [DataSourceUtils](#), or the provided [DirectoryTemplate](#) should make integrating or using jdbc directory simple.

Also, for none managed applications, there is an option working with autoCommit=true mode. The system will work much slower, and it is only supported on a portion of the databases, but any existing code that uses Lucene with any other Directory implementation should work as is.

If working within managed environments, an external transaction management should be performed (using JTA for example). Simple solutions can be using CMT or Spring Framework abstraction of transaction managers. Currently, the jdbc directory implementation does not implement a transaction management abstraction, since there is a very good solution out there already (Spring and JTA). Note, when using Spring and the DataSourceTransactionManager, to provide the jdbc directory with a Spring's TransactionAwareDataSourceProxy.

Author:

kimchy



Compass

[Main](#)[Overview](#)[Wiki](#)[Issues](#)[Forum](#)[Build](#)[Fisheye](#)

Compass is an [open source](#) project built on top of [Lucene](#) aiming at simplifying the integration of **search** into any Java application.

[The Future of Compass & ElasticSearch](#)

Search

Fork me on GitHub



elasticsearch.

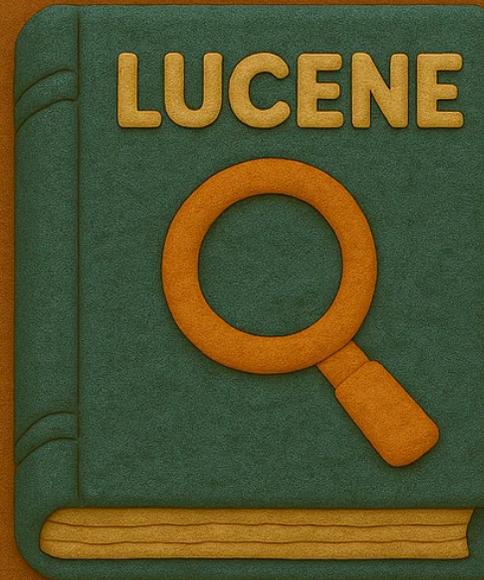
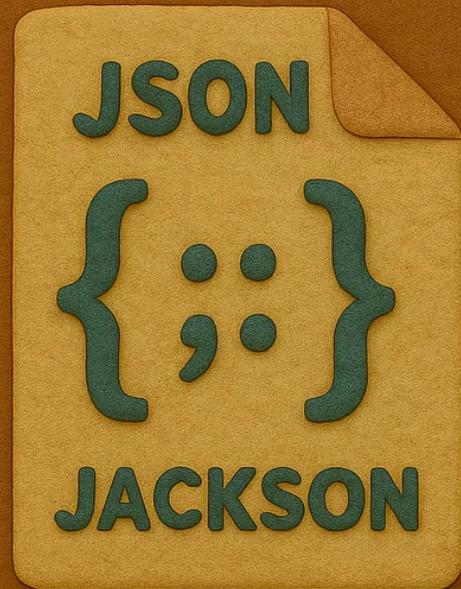
[home](#)
[download](#)
[guide](#)
[blog](#)
[community](#)
[tutorials](#)
[videos](#)



You know, for Search

So, we build a web site or an application and want to add search to it, and then it hits us: **getting search working is hard**. We want our search solution to be **fast**, we want a **painless setup** and a completely **free search schema**, we want to be able to index data simply using **JSON over HTTP**, we want our search server to be **always available**, we want to be able to start with one machine and **scale to hundreds**, we

Elasticsearch



15 years ago

IMAGE
GÉNÉRÉE
PAR
SORA



IMAGE
GÉNÉRÉE
PAR
SORA



IMAGE
GÉNÉRÉE
PAR
SORA



Elasticsearch FR



Conversations



À propos de :



Les raccourcis clavier de Groupes ont été mis à jour

Ignorer

Voir les raccourcis

Elasticsearch FR

1 à 30 sur 789



Attention : changement d'URL !

Merci dorénavant de poser vos questions sur <https://discuss.elastic.co/c/in-your-native-tongue/discussions-en-francais>



Safia G, David Pilato 15

Configuration Logstash : 'grep' non trouvé dans filter – Aaaah d'accord tout ...

01/06/2015



Franck Gir..., David P... 4

include_in_all et champ multi_field – Peut-être. Ca ne sert de toute façon à r...

22/05/2015



Franck..., ... Jérôme... 10

Gestion mémoire – En plus, c'est justement le rôle des machines virtuelles d...

13/05/2015



valentina...@elastic.co

Meetup Elasticsearch France #14 chez Le Bon Coin – Si vous ne l'avez pas ...

12/05/2015



Aurélien ... , David P... 2

Re: [elasticsearch-fr] [IMPORTANT] déplacement du forum vers discuss.elast

05/05/2015



David Pilato

[IMPORTANT] déplacement du forum vers discuss.elastic.co – Bonjour à to...

05/05/2015



Naguelfar 2

Parser un CSV avec des "" – En fait j'ai trouvé :) Si ca peut aider, j'ai défini le ...

04/05/2015





IMAGE
GÉNÉRÉE
PAR
SORA

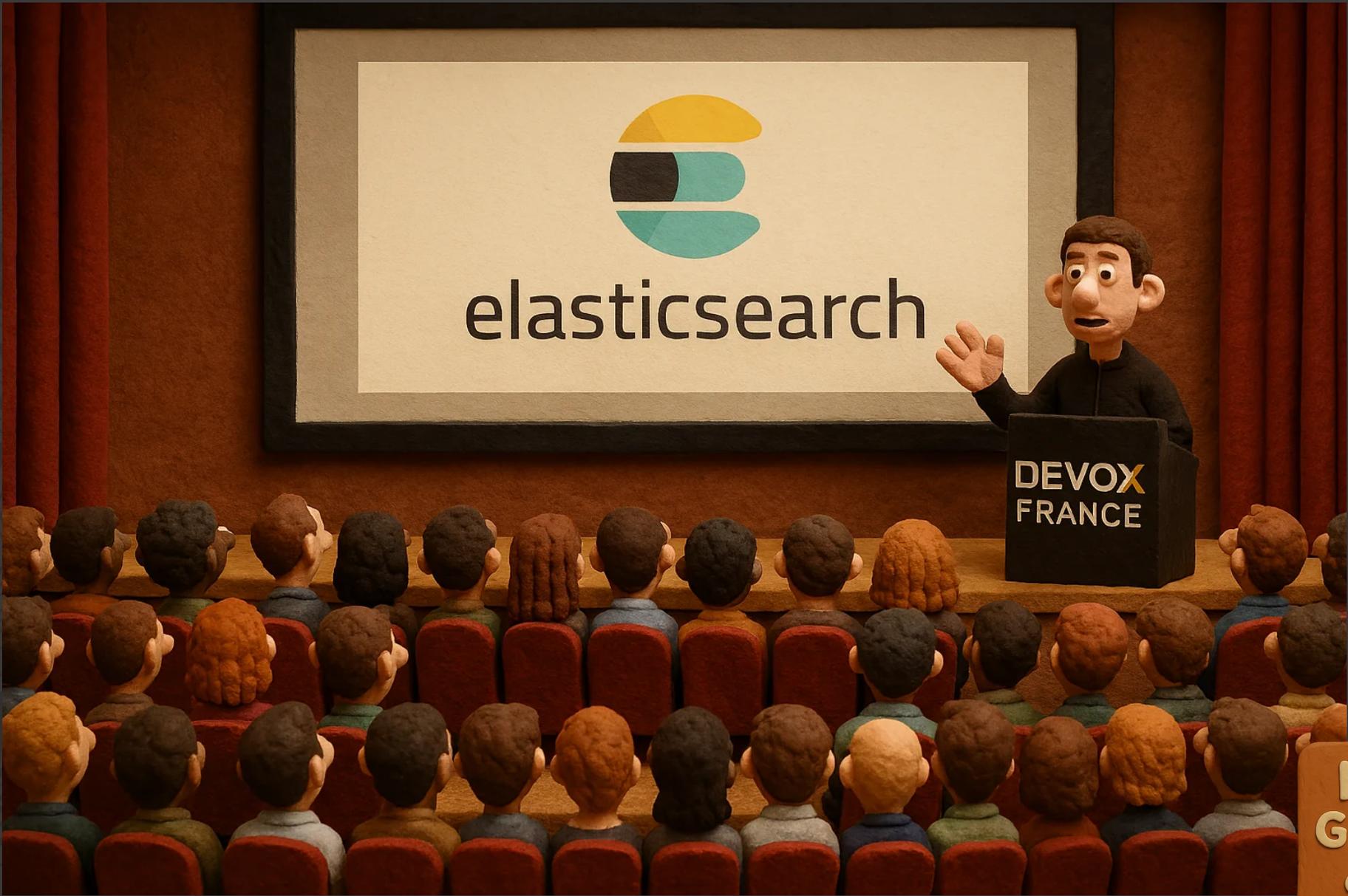


IMAGE
GÉNÉRÉE
PAR
SORA



IMAGE
GÉNÉRÉE
PAR
SORA



IMAGE
GÉNÉRÉE
PAR
SORA



IMAGE
GÉNÉRÉE
PAR
SORA



IMAGE
GÉNÉRÉE
PAR
SORA



IMAGE
GÉNÉRÉE
PAR
SORA



IMAGE
GÉNÉRÉE
PAR
SORA

CONCLUSION



IMAGE
GÉNÉRÉE
PAR
SORA



ENVIE DE BOOSTER TA CARRIÈRE ? OPEN SOURCE-TOI !

David Pilato

feedbacks

