**LUCENE IN ACTION**

# CHAPTER 5:

## Indexing a Microsoft Word document

Hiện nay bộ soạn thảo Microsoft Word vẫn là phần mềm được sử dụng nhiều nhất trong soạn thảo văn phòng. Vì vậy việc thực hiện indexing được MS Word là một việc thiết thực cần phải làm , cần phải biết cách parse tài liệu MS Word để có thể tìm kiếm với Lucene. Nhưng sự khó khăn nhất gặp phải đó là MS Word là chương trình có bản quyền vì vậy format của nó cũng được giấu kín. Nhưng may thay , có một vài project mã nguồn mở có thể khắc phục được vấn đề này . Trong phần này có giới thiệu 2 project mã nguồn mở Jakarta POI và TextMining.org dùng để rút trích text ra.

### Using POI

Poi là một Jakarta project , có thể tìm nó tại <http://jakarta.apache.org/poi> . Nó là project cung cấp các hàm API của java để thao tác với các dạng format cơ bản trên định dạng Microsoft’s OLE 2 Compound Document .

Các hàm trong lớp POI có thể lấy ra text của một file tài liệu MS Word , thêm nữa còn có thể sử lý với các meta-data như là summary , tên tác giả , … đi kèm theo file.

### Using TextMining.org’s API

Các hàm API của TextMining.org còn có thể cung cấp một số khả năng khác nữa so với POI . Có một số phần các hàm của nó được thực hiện tốt hơn POI ví dụ như các hàm thư viện hỗ trợ có tỉ lệ thành công khi lấy text ra từ file MS Word , các thư viện hàm hỗ trợ rút trích text hỗ trợ các phiên bản Word 6/95 mà POI không hỗ trợ được,...

## Indexing an RTF document

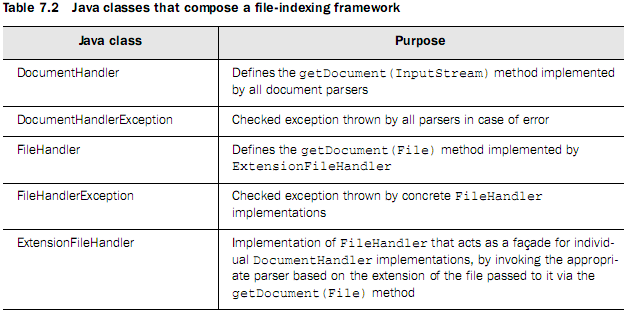
Với Rich Text Format (RTF) chúng ta có thể sử dụng bộ thư viện có sẵn trong gói Java chuẩn , chúng được ẩn trong javax.swing.text và javax.swing.rtf . Các text được lấy ra từ file RTF được chuyển đến Lucene trở thành như Field.Unstored . Cuối cùng DocumentHandler sẽ xử lý phần file plain-text .

## Indexing a plain-text document

Cuối cùng phần thực thi một DocumentHander cho plain-text , phần kết quả thì được tạo như một frameword cho parsing và indexing document cho các phần định dạng khác nhau.

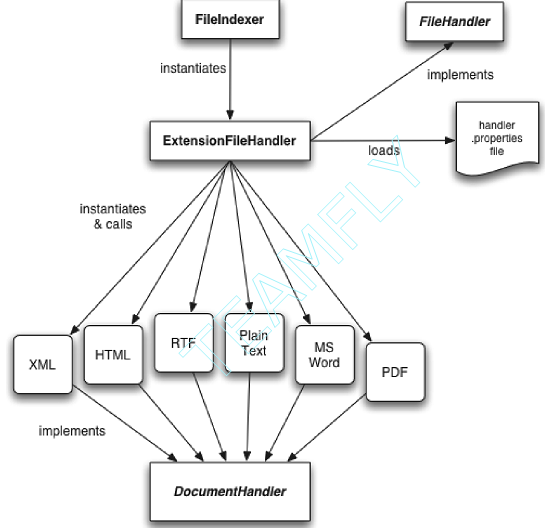
## Creating a document – handling framwork

Phần này nói về sự kết nối của các phần phức tạp khác nhau thành một khối thống nhất , là thi hành parsing của một vài định dạng dữ liệu khác nhau bởi một framwork duy nhất .

Cấu trúc gồm có DocumentHandle interface và kèm theo nó là FileHandlerException và nó thực thi với lớp ExtensionFileHandle. Bảng tóm tắt Framwork components : 

### FileHandler interface

FileHandler là một interface đơn giản giống như là DocumentHandle . Tuy nhiên , điểm khác đó là DocumentHandle dùng giống như InputStream để làm loại dữ liệu đưa vào , còn FileHandler interface định nghĩa File là loại dữ liệu đưa vào ,điều đó sẽ làm việc dễ dàng hơn .



### Extension FileHandler

ExtensionFileHandler chỉ thực thi với FileHandler interface và trả ra một Lucene Document chung . FileHandler thực thi với bất kỳ gói , nếu ngoại lệ sẽ được thông qua FileHandlerException .

### ExtensionFileHandler

ExtensionFileHandler chỉ thực thi trong FileHandler interface. Khi thực thi phương thức getDocument(File) sử dụng phần mở rộng của file để suy ra được loại file và gọi hàm thi hành parser thích hợp. Bởi vì tất phần thực thi parser đều thuộc phần chung DocumentHandler interface, ExtensionFileHandler có thể dò ra được đối tượng File đưa vào trong gói FileInputStream nó có thể biết được cần phải xử lý như thế nào.

### FileIndex application

FileIndex có thể parse và index tất cả các định dạng dữ liệu .

### Using Index

### FileIndexer drawbacks and how to extend the framework

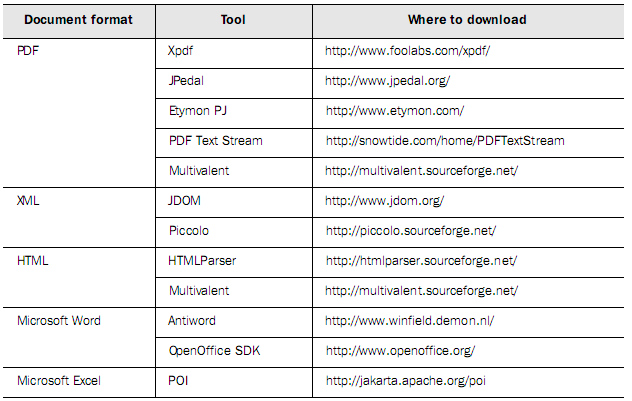
Framework bao có thể parser xử lý các loại file đưa vào như :XML, PDF, HTML, Microsoft Word, RTF , Plain text.

Nhưng nếu cần index và tạo ra file có thể tìm kiếm được với những loại file mà framwork không hỗ trợ thì có thể mở rộng framwork theo các bước sau:

* Viết một parser cho loại file mong muốn và thực thi DocumentHandler interface
* Thêm class parser đến file handler.properties , ánh xạ thích hợp với phần mở rộng của loại file mong muốn.
* Vẫn sử dụng FileIndexer .

## Other text-extraction tools

Trong phần này ta sẽ thấy sự hiện diện của các text được rút trích ra từ , phần indexing của các loại định dạng dữ liệu phổ biến nhất.



### Document-management systems and services

Thêm vào một số bộ thư viện hỗ trợ việc thi hành parser các loại dữ liệu và indexing bằng nhiều cách khác nữa , một vài gói phần mềm và services miễn phí có thể tin tưởng cho Lucene xử lý indexing :

* DocSearcher (<http://www.brownsite.net/docsearch.htm> ) có mô tả của tác giả: “DocSearcher sử dụng Open Source Lucene và POI Apache APIs tốt nhất cho mã nguồn mở Open Source PDF Box API cung ứng tìm kiếm TML, MS Word, MS Excel, RTF, PDF, Open Office (và Star Office) documents, và text documents.”
* Docco (<http://tockit.sourceforge.net/docco/index.html> ) thì nhỏ, theo dữ liệu tại trang chủ thì nó có thể xử lý một số định dạng dữ liệu sau: plain text, XML, HTML, PDF, Microsoft Word và Excel, OpenOffice, và StarOffice 6.0, có thể thực hiện tốt trên UNIX. Không thực hiện với RTF documents.
* SearchBlox (<http://www.searchblox.com/> ) là một tìm kiếm J2EE được triển khai dưới dạng chương trình web. Nó có thể index và search HTML, PDF, Word, Excel, và PowerPoint documents.
* Simpy (<http://www.simpy.com/> ) Một service miễn phí trực tuyến. Nó giúp lưu lại link trực tuyến cho dữ liệu của bạn dưới định dạng như HTML web pages; PDF, Microsoft Word, RTF documents; hay các định dạng khác. Bên cạnh meta-data bạn có thể đi sâu vào từng document, Simpy sẽ crawl và index the tất cả đầy đủ text trong document , đồng cho tìm kiếm trên bất cứ máy tính nào. Dữ liệu của bạn có thể che dấu hoặc chia sẻ trực truyến. Tất indexing và searching là sức mạnh của Lucene, và Nutch là phần cho end user.