

오픈소스 콘텐츠 관리 시스템의 이해

Alfresco의 콘텐츠 연동하기

팀 버너스리가 월드 와이드 웹(World Wide Web, WWW)을 만든 이후, 우리는 이를 통해 수많은 정보를 쉽게 공유하며 이용하고 있다. 팀 버너스리는 “누구든 사용할 수 있게 만드는 것이 바로 웹의 목적이다”라고 밝힌 바 있는데, 이러한 그의 철학대로 전 세계의 정보유통 혁명이 이뤄졌고, 그만큼 정보의 활용도가 높아졌다. 아무리 가치 있는 콘텐츠라 하더라도 이를 이용할 수 없다면, 그 콘텐츠는 죽은 것이나 마찬가지이다. Alfresco는 콘텐츠를 다양하게 이용할 수 있는 방법을 제공하고 있으며, 이를 통해 콘텐츠의 가치를 더욱 높이고 있다.

3

연재순서

- 1회 | 2007.9 | 디지털 콘텐츠와 Alfresco
- 2회 | 2007.10 | Alfresco로 시작하는 콘텐츠 관리
- 3회 | 2007.11 | BPM을 활용한 콘텐츠 관리
- 4회 | 2007.12 | Alfresco의 콘텐츠 연동하기

장선진 jangsunjin@java2game.com

<http://www.java2game.com> | 오픈소스 애플리케이션에 많은 관심을 가지고 있다. 사람들의 삶을 풍요롭게 할 수 있는 소프트웨어를 개발해 함께 나누고자 하는 꿈을 가진 개발자로서 현재 생각을 함께 나눌 공간을 만들어 가고 있다.

지난 연재에서는 jBPM을 활용한 Advanced Workflow의 활용법을 살펴봤다. Alfresco에서는 Advanced Workflow 외에도 다양한 Content Rule 관련 기능을 활용해 콘텐츠를 효과적으로 관리할 수 있는 방안을 제시하고 있다. 만약 더욱 자세한 활용법을 알고 싶다면 언제든지 Alfresco Web-Client의 상단에 있는 ?를 클릭해 온라인 도움말을 확인할 수 있다. 비록 영문으로 작성되어 있지만, 단계별로 설명되어 있어 쉽게 확인할 수 있다.

이제까지 Alfresco를 통해 콘텐츠를 생성, 수정, 승인하는 기능들을 살펴봤다. 이번 회에서는 Alfresco에서 관리되는 콘텐츠를 다른 시스템을 위해 배포하거나 전달하는 것과 관련된 콘텐츠 연동 기능을 소개한다.

Content Management System(CMS)에 콘텐츠가 아무리 많아도 이를 효과적으로 사용자들이 이용할 수 없다면 콘텐츠 활용도가 낮아지게 되고 결국 CMS는 파일 저장소 정도의 역할만 할 수 있으므로 콘텐츠 연동은 CMS에서 매우 중요한 요소이다.

Alfresco의 콘텐츠 연동

CMS를 사용하지 않고 콘텐츠를 공유하는 가장 쉬운 방법은 무엇일까? 윈도우 환경의 경우라면 공유 폴더를 생성해 그 폴더

에 파일들을 정리하는 방법일 것이다. 이 방법의 가장 큰 단점이 라면 공유 폴더는 CIFS(Common Internet File System)/SMB (Server Message Block) 프로토콜을 통한 접근만 가능한 탓에 회사 밖에서 공유 폴더의 콘텐츠를 쉽게 접근할 수 없다는 것이다. 또한 공유 폴더에 저장된 콘텐츠에 대한 이력 관리가 제대로 되지 않아 열심히 작업한 콘텐츠가 다른 사용자에 의해 변경되거나 삭제되는 경우도 종종 발생할 수 있다. 이러한 일이 계속 생긴다면 공유 폴더를 사용하는 사람들은 더 이상 공유 폴더를 콘텐



〈그림 1〉 Shared Drive의 기대 효과
(출처: Alfresco 2.1-Web 2.0 and Content Management)

츠 공유를 위해 쓰지 않을 것이고 결국 작업의 효율성이 떨어지게 된다.

〈그림 1〉은 Alfresco와 같은 CMS를 이용했을 경우 Shared Drive Interface를 통해 다양한 방법으로 콘텐츠를 쉽게 Alfresco에 연동할 수 있고, 연동된 콘텐츠는 Rule 관리를 통해 편리하게 관리하며, 나아가 Web Access를 지원해 쉽게 배포할 수 있음을 보여준다. 즉 CMS의 이용이 콘텐츠 활용의 효율을 높여 결국 CMS를 이용하는 모든 기업 및 사람들이 더 큰 효율을 얻을 수 있다는 의미다. Alfresco는 다음과 같은 다양한 방안을 통해 효율적인 콘텐츠 공유를 지원한다.

Shared Drive Interface 관련 방안

- CIFS/SMB Microsoft File Share Protocol
- FTP
- WebDAV

Web Access 관련 방안

- Web Services
- REST
- RSS

필자는 Alfresco에서 이렇게 다양한 연동 방안을 지원하는 것에 대해 놀라지 않을 수 없었다. CIFS/SMB 지원의 경우에 윈도우 환경에서 네트워크 드라이브 및 탐색기를 통해 쉽게 콘텐츠를 이용토록 할 수 있고 리눅스나 유닉스 환경하에서 Samba를 통해 콘텐츠를 공유할 수 있기 때문이다. 또한 FTP(File Transfer Protocol) 지원은 원격에서 빠른 속도로 콘텐츠를 주고받을 수 있고, WebDAV의 경우 웹사이트 제작 관련 작업을 할 때 어도비 드림위버(Adobe Dreamweaver)와 같은 툴을 주로 사용하는 웹 디자이너들이 쉽게 콘텐츠를 공유할 수 있게 지원하기 때문이다.

또한 Web Service와 REST(Representational State Transfer)는 웹 서비스를 통해 콘텐츠를 배포하려는 CP(Content Provider) 입장에서 무척 유용한 기능이며 콘텐츠 연동의 폭을 대폭 확대할 수 있다. RSS의 경우 원하는 사용자에게 CMS의 콘텐츠 상태를 알리고 배포할 수 있으므로 매우 유용한 배포방안이다. 이 밖에도 Alfresco는 JSON(JavaScript Object Notation)을 지원할 예정이므로 AJAX(Asynchronous JavaScript and XML)를 활용한 웹 애플리케이션에 쉽게 콘텐츠 관련 정보를 제공할 수 있다.

만약 자바로 CIFS(Common Internet File System) 연동이나 FTP(File Transfer Protocol) 및 NFS (Network File System) 등의 프로토콜을 통한 연동을 수행하는 프로그램을 개발해야 한

Portlet을 이용한 콘텐츠 연동

최근 많은 웹 사용자들이 LifeRay(<http://www.liferay.com/>)와 같은 Portlet을 제공하는 서비스를 이용해 자신의 취향에 맞는 정보만을 모아 이용하고 있다. LifeRay는 오픈소스 진영에서 선두를 다투고 있는 Portal Container이다. LifeRay의 웹2.0 기반 사용자 인터페이스를 바탕으로 Theme를 통해 사용자가 원하는 UI를 페이지별로 제공할 수 있다. 자바 진영에서는 Portlet에 대한 사용자의 요구에 맞춰 JSR-168을 통해 Portlet에 관한 스펙과 Portlet을 관리하는 Portlet Container에 관한 스펙을 정의했다. Portlet은 Portlet Container에 의해 관리되는 자바 기반의 Web Component를 의미한다.

JSR-168이 발표되기 전에는 각 Portal Container간에 Portlet 스펙이 상이해 Portlet을 공유해 사용하지 못했으므로 Portlet을 개발할 때 각 Portlet Container별로 개발해야만 했다. 하지만 이제 JSR-168을 통해 하나의 Portlet을 개발하면 JSR-168을 지원하는 모든 Portal Container에 배포할 수 있게 되어 Portlet 개발에 탄력이 붙은 상태이다. 또한 WSRP(Web Services for Remote Portals)를 통해 웹서비스를 사용하는 Portlet간에 데이터를 상호 교환할 수 있게 되었다.

주목할 점은 JSF(JavaServer Faces)를 기반으로 개발된 웹 애플리케이션의 경우 많은 변경이 없이도 쉽게 Portlet으로 바꿀 수 있다는 것이다. Alfresco 역시 이런 점을 놓치지 않았으며 Web Client를 JSF로 개발했다.

그럼 과연 JSR-168과 WSRP가 콘텐츠 연동과 무슨 연관이 있는 것일까? Alfresco는 이미 LifeRay에 콘텐츠 관련 Portlet들을 제공하고 있다. LifeRay를 이용해 본 독자들은 알겠지만 LifeRay에서 콘텐츠 관련 기능들과 토론 관련 기능들이 멋지진 않지만 효과적으로 동작하고 있음을 알 것이다. 이는 Alfresco의 기술을 감싸 않은 Portlet이기 때문에 가능하다. 앞으로 Portlet이 다양해지고 Portlet의 사용이 늘어나면 우리도 모르는 사이에 Alfresco를 Portlet 형태로 사용하게 될 것이다. 또한 Alfresco가 수집한 정보(콘텐츠)는 WSRP를 통해 Portlet간에 자유롭게 교환할 수 있다. 즉 A Portal Container 사이트의 콘텐츠 관련 Portlet에 생성한 내 콘텐츠를 친구가 사용하는 B Portal Container의 Portlet에 자유롭게 송수신할 수 있는 것이다.

이는 전혀 새로운 콘텐츠의 연동 방안이 될 것으로 보인다. 기존의 CIFS나 FTP 등을 이용한 콘텐츠 연동은 사용자가 직접 자신이 사용하는 콘텐츠를 옮겨야 했다면, WSRP를 이용한 콘텐츠 연동은 사용자가 이용하는 Portlet 간에 콘텐츠가 연동되므로 이메일에 첨부된 파일처럼 손쉽게 자료를 공유할 수 있게 된다. 특히 Portlet에서만 사용되는 콘텐츠의 경우 더욱 유용할 수 있는데, 예를 들어 가계부 Portlet의 정보를 세금계산 Portlet으로 보내 세금을 미리 확인하거나, MP3 플레이어 Portlet간에 음악 정보를 공유해 다른 사용자의 Portlet에 담긴 MP3를 내 Portlet에서 연주할 수 있다. 앞으로 발전할 Portlet의 모습과 Portlet에 담긴 콘텐츠들이 궁금하다.

다면 Alfresco의 구현 소스를 살펴볼 것을 강력히 권장한다. 이미 Alfresco는 AIFS(Alfresco Intelligent File System)를 통해 CIFS, FTP, NFS와의 연동을 충분히 구현했으며 이러한 기술을 Alfresco JLAN이란 이름으로 GPL하에 공개했다. 따라서 여러 분들이 고민하고 있는 문제를 Alfresco에서 이미 해결했으므로 Alfresco JLAN을 사용해 각종 연동 문제를 효과적으로 해결할 수 있다.

Alfresco JLAN의 소스는 소스포지의 Alfresco 사이트에서 다운로드할 수 있으며, 다양한 예제를 포함하고 있어 쉽게 응용할 수 있다. 이와 관련해 궁금한 점이 있다면 언제든지 필자에게 이메일을 보내길 바란다. 이렇게 다양한 콘텐츠 연동 방안 중에서 Alfresco를 통해 가장 가치 있게 콘텐츠를 연동할 수 있는 방안들을 중점적으로 살펴본다. 각 연동 방안은 상상하는 것보다 무척 쉬우므로 가볍게 한번 살펴보는 것만으로도 바로 활용할 수 있다.

각 연동 방안을 자세히 설명하기 전에 프로토콜의 특성상 윈도우를 사용하는 컴퓨터에 Alfresco가 설치되어 있다고 가정한다. 만약 Alfresco가 설치되어 있지 않다면 이전 글을 참고하길 바라며, 리눅스나 유닉스 환경의 사용자라면 참고자료의 Alfresco CIFS Linux를 참조하길 바란다.

CIFS를 활용한 콘텐츠 연동하기

CIFS는 인터넷을 통해 원격지 컴퓨터상의 파일이나 서비스에 대한 요구를 할 수 있게 하는 프로토콜이다. 보통 윈도우를 사용하는 컴퓨터간에 파일을 공유하거나 프린트 서비스 등을 공유하는 데 사용된다. CIFS/SMB에 대한 더욱 자세한 내용은 참고자료를 확인하길 바란다. CIFS/SMB를 활용한 연동은 매우 효과적이고 간단하므로 다음 절차를 통해 즉시 연동할 수 있다.

네트워크 폴더 열기

Alfresco Web-Client에 로그인해 특정 스페이스로 이동한 후,



(화면 1) 네트워크 폴더 열기

(화면 1)과 같이 스페이스명 옆에 있는 을 클릭한다.

(화면 1)에서 보는 것처럼 에 마우스 커서가 놓이면 풍선 도움말에 \\JANGSUNJIN_A\Alfresco\J2G-FX라는 내용이 보인다. 이는 J2G-FX라는 스페이스에 접근할 수 있는 주소이며 각 의미는 다음과 같다.

\\JANGSUNJIN_A\Alfresco\J2G-FX

- | | | | | — 스페이스명
- | | | | | — 접두 주소
- | | | | | — “_A”는 Alfresco의 호스트명 인식 기호
- | | | | | — “JANGSUNJIN”은 컴퓨터 이름

특이한 점은 컴퓨터 이름에 ‘_A’가 붙은 것이다. 이는 Alfresco가 구동되면서 NetBIOS 테이블에 자신의 호스트명을 등록할 때 자동으로 ‘컴퓨터명’ + ‘_A’라고 붙이기 때문이다. 이는 Alfresco가 설치된 컴퓨터마다 유일한 호스트명을 부여하게 함으로써 CIFS를 통한 접근시 혼동을 막기 위한 것으로 보인다. Alfresco는 접두 주소로 보면 되며 그 후의 주소들은 모두 스페이스명이다. Web-Client에서 각 스페이스의 주소를 쉽게 확인할 수 있으므로 주소를 기억하지 않아도 언제든지 쉽게 접근할 수 있다.

Alfresco에 연결

을 클릭한 후에는 CIFS를 통해 Alfresco에 연결할 수 있는 창이 나타난다. 각 사용자 계정의 ID와 패스워드를 입력한다. 이 예에서는 admin을 통해 로그인한다. 간혹 ID/패스워드를 정확히 입력했음에도 불구하고 연결이 지연되는 경우가 있다. 재차 시도하면 연결이 원활하게 이뤄진다.



(화면 2) 로그인 창

스페이스 내역 확인

〈화면 3〉은 Alfresco에 연결된 후의 J2G-FX 스페이스이다. Drafts라는 폴더는 J2G-FX의 하위 스페이스이며 더블클릭하면 하위 스페이스의 내용을 확인할 수 있다. 만약 새로운 콘텐츠를 생성하고 싶은 경우에는 작성하고 있던 콘텐츠를 바로 복사하면 콘텐츠 업로드가 완료된다. 또한 콘텐츠를 삭제하고 싶다면 해당 콘텐츠를 바로 삭제하면 된다. 공유 폴더를 사용할 때와 마찬가지로 편리하게 사용할 수 있다.



〈화면 3〉 스페이스의 내용

특이한 점은 〈화면 1〉과 비교할 때 Alfresco Web-Client에 보이지 않았던 __Alfresco라는 인터넷 바로가기 파일과 __CheckInOut.exe 및 __ShowDetails.exe라는 실행 파일이 있다는 점이다. 이 파일들은 각 스페이스마다 있는데 그 각각의 역할은 다음과 같다.

__Alfresco 바로가기: 웹 브라우저를 통해 Alfresco Web-Client의 해당 스페이스 화면으로 바로 이동함.

__CheckInOut.exe 실행 파일: 콘텐츠를 체크인하거나 체크아웃하고 싶은 경우 해당 콘텐츠를 끌어서 __CheckInOut.exe 실행 파일에 놓으면 체크 아웃된 콘텐츠는 체크인되며, 체크인한 콘텐츠는 체크아웃이 된다.

__ShowDetails.exe 실행 파일: 콘텐츠에 대한 자세한 정보를 확인하고 싶을 때 해당 콘텐츠를 끌어서 __ShowDetails.exe에 놓으면 웹 브라우저가 실행되면서 Alfresco Web-Client의 콘텐츠 상세정보 페이지를 보여준다.

이렇게 특별한 파일 및 Web-Client를 통해 제공되는 기능들을 네트워크 폴더상에서도 쉽게 실행시킬 수 있어 편리성이 더욱 증대되었다. 특히 CIFS를 통한 네트워크 연결은 일반적인 윈

도우 사용자라면 익숙한 기능이라 별도의 교육 없이 Alfresco의 활용성을 더욱 높일 수 있다. 자주 사용하는 스페이스가 있다면 네트워크 드라이브를 연결해 윈도우의 탐색기로 쉽게 접근할 수 있다.

FTP를 활용한 콘텐츠 연동하기

CIFS 연동을 살펴보면 무척 쉽다는 생각을 했을 것이다. 애플리케이션을 개발할 때 다른 시스템과의 연동 과정에는 많은 어려움이 따르지만 Alfresco는 각 연동 문제를 깔끔하게 해결한 좋은 예인 셈이다. FTP의 지원은 원격에 있는 사용자들에게 매우 훌륭한 연동 방법이다. 특히 대용량의 콘텐츠를 송수신해야 하는 사용자의 경우 CIFS보다 FTP가 콘텐츠 전송에 유리하므로 FTP를 활용한 콘텐츠 연동을 추천한다. CIFS와 마찬가지로 FTP를 활용한 콘텐츠 연동 역시 매우 간략하게 이뤄진다.

FTP를 통한 접속

FTP로 Alfresco에 접속하기 위해서는 FTP Client 프로그램이 필요하다. 접속할 호스트명에 Alfresco가 설치된 컴퓨터의 IP 주소나 호스트명을 입력한다. 만약 로컬 컴퓨터에 접속하러 한다면 localhost나 CIFS 연동과 마찬가지로 JANGSUNJIN_A를 입력한다. FTP 계정정보에는 CIFS와 마찬가지로 사용하고 싶은 계정의 ID 및 패스워드를 입력한다.

콘텐츠 업로드

FTP Client에 접속이 완료되었다면 원하는 스페이스에 콘텐츠를 바로 업로드할 수 있다. 콘텐츠를 업로드한 후 Alfresco Web-Client를 통해 확인해 보자. 만일 HTML 문서나 TXT 문서를 작성한 후에 업로드했다면, Alfresco Web-Client에서 직접 콘텐츠를 편집하려고 시도했을 것이다. 이상한 것은 이전 연재에서 생성한 HTML 문서의 경우 바로 인라인 편집기를 통해 편집할 수 있는데, 새로 업로드한 문서는 꼭 체크아웃해서 다운로드한 후 체크인해야만 편집할 수 있다는 것이다. 이는 해당 콘텐츠가



〈화면 4〉 인라인 편집 허용

인라인 편집이 허용되어 있지 않기 때문이다. 따라서 콘텐츠의 속성 창에서 <화면 4>와 같이 Allow Inline Editing을 클릭하면 인라인 편집을 할 수 있다.

주의해야 할 점은 FTP를 이용할 경우에는 CIFS처럼 체크인 및 체크아웃 기능이 제공되지 않으므로 공동으로 작업할 경우 다른 사람이 올린 콘텐츠를 덮어쓸 수 있다는 점이다. 따라서 FTP를 통해 작업할 스페이스에는 콘텐츠 버전이 자동으로 부여될 수 있도록 Rule을 부여함으로써 콘텐츠가 덮어써지더라도 버전별로 관리될 수 있게 해야 한다. 버전을 자동으로 부여하는 Rule을 부여하면 FTP로 업로드할 때마다 버전이 올라가므로 자동으로 히스토리를 관리할 수 있다.

RSS를 활용한 콘텐츠 연동하기

이제까지 콘텐츠 연동 방안 중 Shared Drive Interface 관련 방안을 살펴봤다. 그럼 지금부터는 Web Access 관련 방안을 살펴볼 차례이다. 최근 들어 블로그나 여러 웹사이트에서 RSS를 적극적으로 제공하고 있으며 RSS를 활용해 자신만의 포털 페이지를 꾸미는 일 또한 늘어나고 있다.

RSS는 여러 버전이 있으며 RSS라는 약어에 대한 정의도 분분하지만 대부분의 RSS Reader에서 거의 모든 RSS 버전을 지원하므로 RSS를 이용하는 데 큰 문제는 없다. Alfresco에서는 스페이스 단위로 RSS를 지원하고 있으며, 스페이스의 새로운 내용이 갱신될 때 RSS 내용 역시 갱신되는 구조를 가지고 있다. Alfresco에서 지원하는 RSS 버전은 RSS 2.0이다.

Alfresco는 RSS 연동을 매우 특별한 방법으로 지원하고 있는데 FreeMarker를 이용해 RSS 2.0 규격에 맞는 XML 문서를 생성해 내는 방식이다. 원래 Alfresco는 콘텐츠를 다양한 방식으로 표현하기 위해 FreeMarker와 XSLT(eXtensible Stylesheet Language Transforms)를 지원한다. 따라서 콘텐츠에 따라 템플릿 언어를 이용해 사용자의 요구에 맞는 다양한 뷰를 만들어 낼 수 있다는 강점을 지니게 되었다. 특히 FreeMarker는 Alfresco의 대부분의 커스텀 뷰를 생성할 때 편리하게 사용할 수 있다.

템플릿 언어를 사용한다고 해도 수많은 콘텐츠들을 표현하기 위해서는 데이터 모델이 논리 정연해야 한다. 이를 위해 Alfresco는 기본적인 콘텐츠의 모델을 가지고 있다. 이 콘텐츠 모델에서는 companyhome, person, space와 같은 모델을 통해 콘텐츠에 접근할 수 있는 방안을 제시하고 있다. 또한 템플릿 언어를 사용하면서 각 콘텐츠의 상태에 따른 분기 등도 필요한데 이를 위한 TemplateNode Model API를 제공하고 있어 FreeMarker와 같은 템플릿 언어를 사용한 뷰 개발을 돕고 있다. 템플릿 언어를 통

한 다양한 뷰를 제공하는 기능 역시 Alfresco의 새로운 가치라고 생각한다.

RSS를 통한 콘텐츠 연동 역시 너무도 쉽게 이뤄지는데, 여기서는 J2G-FX 스페이스를 중심으로 설명하고자 한다. 만약 다른 스페이스를 생성해 절차를 진행하고자 한다면, J2G-FX 스페이스를 대신 생성한 스페이스명으로 생각하면 된다.

스페이스에 RSS Feed Template 적용하기

RSS를 제공하려는 스페이스로 이동한 후, 스페이스의 상세정보 페이지 맨 하단에 있는 RSS Feed 영역의 Apply RSS Feed Template을 클릭한다.



<화면 5> RSS Feed Template 적용

RSS Template 선택하기

RSS Template을 선택하는 리스트 박스에서 RSS_2.0_recent_docs.ftl을 선택한 후 OK 버튼을 클릭한다.

RSS 접근 권한 추가하기

RSS Template을 선택한 후 다시 스페이스의 상세보기 페이지로 돌아오면 RSS Feed 영역에 RSS Feed Link가 나타난다. 이를 바로 클릭하면 403 에러가 발행한다. RSS Feed를 위해 접근하는 사용자에게 콘텐츠 접근 권한이 없기 때문이다. 따라서 스페이스에 접근할 수 있는 권한을 추가해야 한다.

스페이스 오른쪽에 있는 Manage Space Users를 클릭한다. Manage Space Users 'J2G-FX' 페이지에서 오른쪽 상단에 있는 Invite를 클릭한다.

Invite User Wizard 설정하기



Invite User Wizard가 시작되고 1단계인 Invite Users에서 Search 버튼을 클릭한다. 이어서 Guest를 선택하고 Role을 Consumer로 선택한다. 이는 Guest에게 Consumer란 Role을 부여하겠다는 의미다. 참고로 Consumer Role은 콘텐츠를 단순히 조회할 수만 있다. Add to List 버튼을 클릭하면 Guest (Consumer)와 같이 사용자와 Role이 나타난다. 더 자세한 내용은 이전 연재를 참고하길 바란다.

Next 버튼을 클릭한 Guest의 경우 메일 발송이 필요 없으므로 바로 Next 버튼을 클릭해 설정한 내역을 확인한다. 이상이 없다면 Finish 버튼을 클릭해 Invite Users Wizard를 종료한다.



〈화면 6〉 Invite User Wizard 설정

RSS Feed 확인

Guest 추가 후 다시 스페이스로 돌아왔다면 스페이스 명칭 옆에  이 추가된 것을 확인할 수 있다.  를 클릭하면 바로 RSS Feed를 위해 생성된 XML을 보거나 RSS Reader 기능이 탑재된 브라우저의 경우라면 〈화면 7〉처럼 정상적으로 RSS Feed를 수행했을 것이다.



〈화면 7〉
파이어폭스를
이용한 RSS Feed

여기서 주목해야 할 점은 실제 스페이스에 있는 콘텐츠의 내역과 RSS Feed된 데이터에 나타난 콘텐츠의 항목이 서로 다르다는 점이다. 이는 RSS Feed를 위해 적용한 RSS_2.0_recent_docs.ftl이 최근 일주일 사이에 변화가 있는 콘텐츠만 나타나도록 설정되어 있기 때문이다.

RSS_2.0_recent_docs.ftl 변경



보통 FreeMarker나 XSLT를 사용해 웹 개발을 하는 경우 FTL(FreeMarker Template Language) 파일이나 XSLT 파일을 서버 내의 특정 디렉터리에 모아놓고 사용하는 경우가 많다. 하지만 이럴 경우 FTL을 수정할 때마다 파일을 교체해야 하며, 실제 운영 중인 서버라면 파일 교체 작업의 위험 부담이 클 수밖에 없다.

Alfresco는 FTL 파일 역시 하나의 콘텐츠로 보고 자기 자신이

관리하고 있다. 〈화면 8〉과 같이 Alfresco는 스페이스에 RSS Template들만 모아놓고 있다. 따라서 다른 종류의 RSS Feed를 위한 템플릿을 생성하고 싶다면 Company Home > Data Dictionary > RSS Templates에 FTL 문서를 생성하면 된다. 여기서는 RSS_2.0_recent_docs.ftl을 복사한 후 편집한다.



〈화면 8〉 RSS Template 스페이스

편집 역시 너무도 쉽게 인라인 에디터를 통해 이뤄진다. 주저 없이 복사본 FTL의  를 클릭해 편집을 시작하자. 만일 FTL 문법을 잘 모르더라도 〈리스트 1〉과 같이 고치는 데 큰 문제는 없을 것이다. 이처럼 자신이 원하는 대로 편집해 사용할 수 있고, 또한 즉시 안전하게 적용될 수 있다. 

〈리스트 1〉 RSS_2.0_recent_docs.ftl 편집의 예

```
<?xml version="1.0"?>
<rss version="2.0">
  <channel>
    <title>J2G-FX Artifact RSS</title>
    <copyright>http://java2game.com</copyright>
    <#assign hostname="http://java2game.com/alfresco">
    ~ jfΣ ~
    <#assign weekms=1000*60*60*24*7*4>
    ~ jfΣ ~
  </channel>
</rss>
```

참고 자료

1. SourceForge.net의 Alfresco 프로젝트: <http://sourceforge.net/projects/alfresco/>
2. Alfresco JLAN: <http://www.alfresco.com/products/aifs/>
3. Alfresco REST API: http://wiki.alfresco.com/wiki/Alfresco_REST_API
4. CIFS/SMB: <http://en.wikipedia.org/wiki/CIFS>
5. Alfresco CIFS Linux: http://wiki.alfresco.com/wiki/CIFS_linux
6. Samba: www.samba.org/
7. WebDAV Resources: <http://webdav.org/>
8. Introducing JSON: <http://json.org/>
9. Alfresco Template Guide: http://wiki.alfresco.com/wiki/Template_Guide
10. FreeMarker: <http://www.freemarker.org/>