

E 드라이브 본문 검색 UI 개발 프로젝트

Search UI Development Project for E Drive Docs

2016104146 이승윤

2015104187 안형욱

요 약

조직의 보안 측면에서 문서중앙화 관련 관심이 높아지고 있다. 그런데 이렇게 대량의 파일들을 중앙화하면 파일명, 폴더명과 같은 단순 메타정보 기반의 검색은 사용자가 원하는 정보를 빠르게 검색하는 것이 불가능하다. 또한, 파일 본문 내용의 검색이 불가능하므로 문서 내 본문 내용을 추출할 수 있는 기술을 기반으로 추출 완료된 문서 내용을 효율적으로 검색할 수 있는 검색 엔진 인터페이스를 개발한다.

1. 서 론

조직의 보안 측면에서 문서중앙화에 대한 관련 관심이 높아지고 있다. 회사측면에서 주요 문서를 중앙화하여 통합하면 자산화를 할 수 있음과 동시에 분실 및 휴대성의 문제에서도 손쉽게 해결할 수 있는 방안이 된다. 네트워크 드라이브처럼 사용하기 때문에 사용의 편의성도 일부 보장될 수 있는 것이다. 하지만 여러 문서들을 중앙화함으로써 생기는 다양한 문제점도 존재한다. 그 중에 하나로 대량의 파일들을 중앙화하게 되면 파일명, 폴더명과 같은 단순 메타정보 기반의 검색은 사용자가 원하는 정보를 빠르게 검색하는 것이 불가능하다. 또한 주변에서 쉽게 볼 수 있는 검색엔진 시스템에서도 파일 본문 내용의 검색기능은 찾아볼 수 없다.

따라서 대량의 파일들을 중앙화하였을때 생기는 문제점을 해결함과 동시에, 사용자 편의성 개선능력도 향상시킬 수 있도록 하기 위해 본 프로젝트는 문서 내 본문 내용을 추출할 수 있는 기술을 기반으로 추출 완료된 문서 내용을 효율적으로 검색할 수 있는 검색 엔진을 개발하는 것을 목표로 한다.

기존의 파일시스템 검색의 경우를 비교 및 대조하였을때, 윈도우 운영체제의 경우 파일명에 대한 검색은 가능하지만 문서 내부의 문자를 검색해야할 때 문서 파일을 직접 열어 검색해야 하는 불편함을 찾아 볼 수 있었다. 본 프로젝트는 이러한 파일 시스템을 사용 하였을 때와 유사한 성격을 지니고 있기 때문에, 이를 참고하여 문서중앙화를 이용하는 사용자 측면에서의 불편함을 줄이기 위하여 스토리지 내에 공용으로 사용하는 사용자들에게 문서의 제목 뿐만 아니라 본문 내용까지 검색하고, 그 내용을 미리볼 수 있도록 하여 문서 내용을 효율적으로 검색 할 수 있는 사용자 인터페이스, 즉 UI를 웹을 통해 제공하도록 할 것이다.

2. 프로젝트 구조

개발되어 있는 문서 파서와 ELK Stack, Django 기반

백엔드 처리 기술을 활용하여 사용자 친화적인 UI를 가진 검색 S/W를 개발하고자 한다. 이를 통한 프로젝트 전체 구조는 다음 그림과 같다.

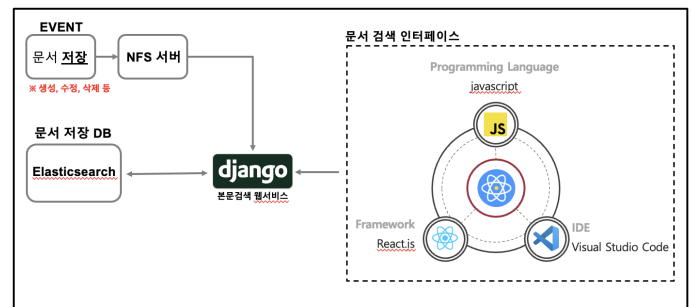


그림 1. 프로젝트 전체 구조

본 프로젝트는 사용자 친화적인 UI를 가진 검색 S/W를 개발하는 것이다. 활용 기술은 JS를 기반으로 사용할 수 있는 React.js를 중심으로 하며, 리액트의 상태관리를 Redux를 활용하여 개발한다.

검색 UI는 크게 검색 창, 검색 결과로 나눌 수 있으며 검색 창에는 검색 기준, 검색 방식에 따라 고려되며, 이에 따른 기본적인 요소는 다음 그림과 같다.

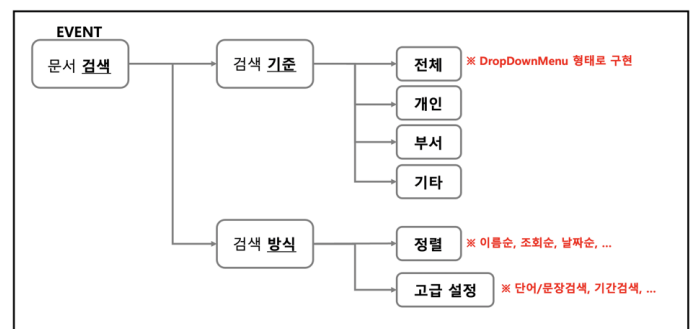


그림 2. 검색 창 기본 구조

검색 결과에는 다음과 같은 요소가 고려된다.
 검색 결과의 파일명, 작성자, 작성일, 경로와 같은 메타 정보
 확인, 연관 검색어, 검색 기준에 따른 카테고리 메뉴화
 이에 따른 기본적인 요소는 다음 그림과 같다.

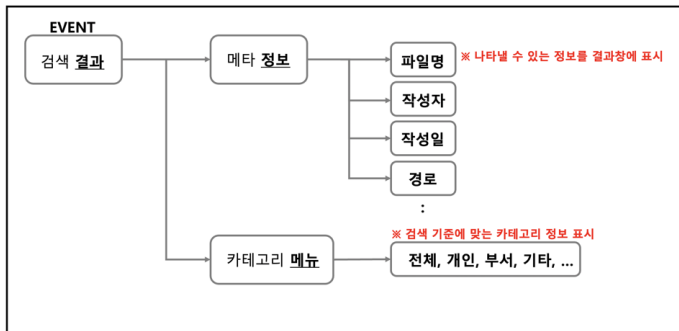


그림 3. 검색 결과 기본 구조

3. 관련 연구

해당 프로젝트의 개발 수행을 위해 아래의 기술을 사용한다.

3.1. 웹 인터페이스

3.1.1. React

리엑트는 컴포넌트 기반으로 사용자 인터페이스를 만들 수 있는 라이브러리이다. 컴포넌트들을 여러가지로 조합이 가능하고 재사용할 수 있기 때문에 코드의 효율과 그에 따르는 유지보수가 용이해지는 장점이 있다. 또한 리엑트는 Virtual DOM을 사용해 업데이트시에 변경된 부분만 DOM을 업데이트하기 때문에 최적화를 할 경우 빠른 성능적 장점을 얻을 수 있다. 따라서 위와 같은 장점을 얻고자 이번 프로젝트의 웹 인터페이스를 React.js를 사용하여 구현한다.

3.1.2. Redux

리덕스는 자바스크립트 어플리케이션을 위한 예측 가능한 상태관리 라이브러리로, 리엑트와 함께 사용하여 글로벌 상태를 관리하는 역할을 할 수 있다. 리덕스는 글로벌 상태 관리 뿐만아니라 각종 미들웨어를 함께 설치할 수 있어 확장성이 용이하다. 하지만 리덕스는 기본적으로 작성해야할 액션, 리듀서 함수에 대한 boilerplate 코드가 많지만 이번에 리덕스에서 공식적으로 출시한 Redux-Toolkit을 사용해 해당 문제점을 해결할 수 있기 때문에 이를 사용해 글로벌 상태 관리를 이루고자 한다.

3.2. 기타

3.2.1 Elasticsearch

Elasticsearch는 텍스트, 숫자, 위치 기반 정보, 정형 및 비정형 데이터 등 모든 유형의 데이터를 위한 무료 검색 및 분석 엔진으로 분산형 및 개방형적 특징을 가진다. Elasticsearch는 Apache Lucene을 기반으로 구축되었으며 간단한 REST API, 분산형 특징, 속도, 확장성으로 유명하다. Elasticsearch의 속도와 확장성, 그리고 수많은 종류의 콘텐츠를 색인할 수 있는 능력을 통해 이번 프로젝트의 웹 사이트 검색의 핵심 기능을 담당한다.

3.2.2 Django

장고는 high-level 파이썬 웹 프레임워크로 빠르고 적은 코드 사용으로 웹 어플리케이션을 작성할 수 있다. 이번 프로젝트에서 장고는 백엔드 서버를 담당하여 이번 프로젝트에 필요한 기능들에 대한 로직을 처리하고 이를 프론트엔드에 api로 제공하는 역할을 한다.

4. 요구 사항

본 프로젝트에서 구현할 필수 기능들에 대한 설명과 간단한 UI 디자인을 서술한다.

4.1. 기본 문서 검색(제목 및 파일 내용)

네이버, 구글과 같은 검색엔진과 같이 헤더부분에 검색 창, 입력란에 따른 검색 기준, 카테고리, 버튼, 고급 검색, 정렬기준등을 포함하고 메인 화면으로 돌아감과 동시에 상징하는 로고 아이콘, 그리고 우측 상단에 사용자 로그인 정보를 포함시킨다. 카테고리로는 전체, 개인, 부서에 따른 검색 결과를 보여줄 것이며, 이는 입력란에 검색할 시에 입력란 좌측에 있는 검색 기준으로도 한번에 검색할 수 있도록 하였다.

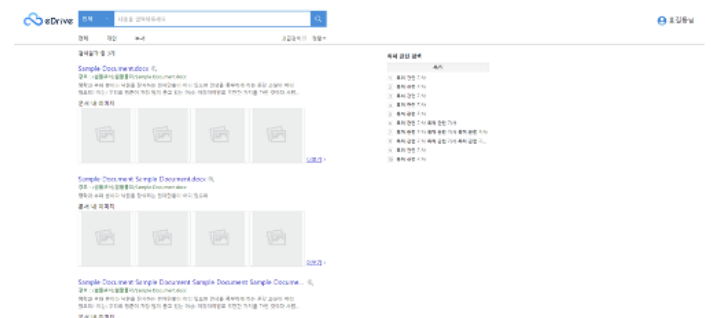


그림 4. 기본 문서 검색 결과 전체 화면

4.2. 고급 검색 / 더보기

검색 기준 외에 원하는 검색 옵션에 따라 사용자가 검색 할 수 있도록 고급 검색란을 추가하였다. 사용자는 고급 검색을 통해 조금 더 상세한 필터링 효과를 느낄 수 있으며, 기본적으로 사용할 고급 옵션에는 단어/문장 검색, 최초 작성자 검색, 최종 수정자 검색, 기간 검색등이 있다. 추후 사용자가 많이 사용할법할 검색옵션을 조사하여 추가 / 수정할 예정이다. 또한 검색 결과란의 파일명 우측에 파일 상세보기아이콘을 구현하여 파일에 대한 상세 정보를 보여준다.

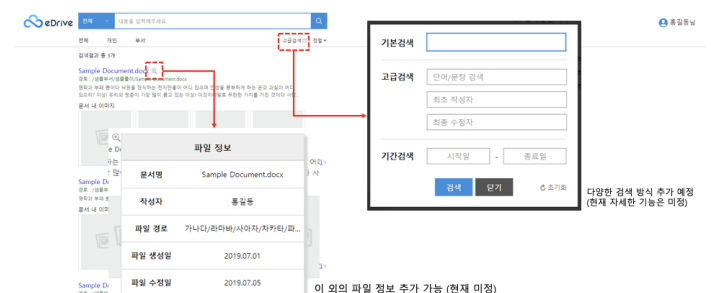


그림 5. 고급 검색 / 더보기

4.3. 로그인

기본적인 로그인 폼(Form)은 구글(Google)의 로그인 폼을 빗대어 제작한다.

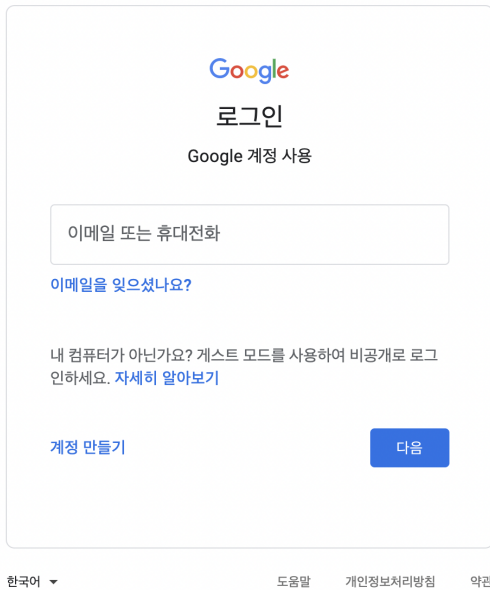


그림 6. 구글의 로그인 페이지

4.4. 문서 다운로드

문서 검색 결과에 따른 정보중, 파일명을 클릭시 다운로드가 되는 형태로 구현한다. 또는 파일명을 우클릭할 시, 파일 자세히 보기, 문서 파일 미리보기등의 기능도 추가로 구현할 예정이다.

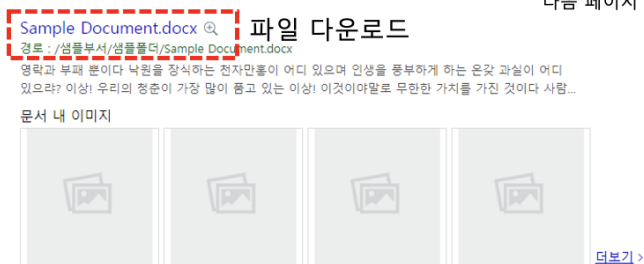


그림 7. 문서 다운로드

4.5. 결과 문서 미리보기

문서 미리보기의 경우, 문서 자체를 미리보기, 내용별, 이미지별로 구분하여 미리보기의 형식으로 나누어 보여줄 예정이다. 본문 검색 결과 페이지의 경우, 파일명 밑에 경로, 문서 내용, 문서 내 이미지를 차례대로 보여주며, 문서 자체의 미리보기일 경우, 사용자가 특정 상태(우클릭)을 할 시에 보여주는 것으로 구현할 예정이다.



그림 8. 결과 문서 미리보기

5. 결론 및 기대효과

본 프로젝트는 클라우드 기반 본문 검색 서비스를 제공하는 조직의 문서중앙화에 의한 검색엔진을 제작하는 것이다. 이는 기존에 사용하였던 문서중앙화된 파일의 검색및 조회를 더 쉽게 사용 할 수 있는 효과를 가져다준다. 또한 고급검색 기능을 통해 사용자가 직접 최적화된 맞춤 검색 시스템을 사용할 수 있는 서비스를 제공하고, 사용자 친화적인 UI/UX를 구현하여 서비스의 사용성을 높인다. 나아가 대량의 파일들에 대한 쉽고 빠른 검색을 조직의 중앙화문서에 이용되는 것 뿐만 아니라 대량 문서파일에 대한 접근을 필요로하는 모든 시스템에 서비스를 제공 할 수 있다.

이 서비스에 대한 수요는 private하게 문서 관리를 하고 싶어하는 회사나 공개적인 데이터를 제공하는 공공서나 서비스 기업들에 초점을 맞추고 제작되었지만, 앞서 말했듯이 범용적 서비스에서도 발전될 가능성을 가지고 있다.