## Omeka 수정

### 모두 선택 기능

Page 편집 기능 중에는 item을 선택해서 삽입하는 기능이 있습니다. 보통 Add attachment라는 버튼을 누르면 브라우저 오른편에 사이드바가 나와서 item을 검색하고 추가할 수 있도록 되어있는데 Quick add 기능이 있지만 수십개 item의 체크박스를 일일히 선택한 다음 추가하는 것이 굉장히 어렵다는 피드백이 있어서 모두 선택 기능을 간단히 추가했습니다. [[1]](#footnote-22)

수정한 파일은 application/view/omeka/admin/item/sidebar-select.phtml과 application/asset/js/resource-selector.js 두 가지 입니다.

**sidebar-select.phtml**

...  
<?php if ($itemsFound): ?>  
<button type="button" class="quick-select-toggle"><?php echo $this->translate('Quick add'); ?></button>  
<div id="duct-tape-1">  
 <input id="select-all" type="checkbox">  
 모두 선택  
</div>  
<div class="resource-list">  
....  
   
....  
<style>  
.sidebar #item-results #duct-tape-1 {  
 display: none;  
 padding: 6px 12px 5px;  
}  
  
.sidebar #item-results.active #duct-tape-1 {  
 display: block;  
}  
  
.sidebar #duct-tape-1 #select-all {  
 margin-right: 12px;  
}  
</style>

이 중에서 3~7 line과 12~25 line이 새로 추가해 넣은 코드인데 앞 부분은 '모두 선택' 글자와 체크박스를 추가하는 HTML이고 뒷 부분은 HTML을 정렬하는 CSS 코드입니다.

**resource-selector.js**

...  
$('#select-resource').on('click', '#duct-tape-1 #select-all', function () {  
 var selectCheckbox = $('#select-resource .resource-list .select-resource-checkbox');  
 if($(this).prop('checked')) {  
 selectCheckbox.prop('checked', true);  
 } else {  
 selectCheckbox.prop('checked', false);  
 }  
});  
...

이 부분은 모두 선택 체크박스를 클릭했을때 체크 상태를 확인해서 표시중인 item들을 모두 선택하거나, 선택 해제하도록 하는 javascript 코드입니다.

## 모듈 수정

### Folksonomy

사용자가 item에 직접 태그를 달 수 있는 이 모듈은 특정 글자(철, 애)로 시작하는 태그를 입력했을 때 ??로 글자가 깨지는 현상이 나타났습니다. 확인해본 결과 깨지는 글자들이 특정 바이트코드(\xa0)로 시작한다는 점을 확인했고 개발자에게 물어본 결과 프랑스어 입력기 프로그램들에서 공백 문자로 사용되는 것이 해당 바이트 코드이며, 통상적으로 사용자 입력을 받는 프로그램들은 시작하거나 끝나는 공백문자를 제거하기 때문에 문제가 발생한다는 것을 알 수 있었습니다.

해당 문제는 직접 코드를 수정한 후 github 저장소에 [반영 요청](https://github.com/Daniel-KM/Omeka-S-module-Folksonomy/pull/6)을 했고 받아들여졌습니다.

## 테마 수정 (centerrow)

테마는 [centerrow v1.2.0](https://github.com/omeka-s-themes/centerrow)을 fork하여 수정했습니다. 앞서 말한대로 가급적이면 디자인 상의 수정사항을 전부 테마에서 처리하는 것이 유지보수상 유리하다는 판단 하에 다른 모듈들이 출력하는 결과물들을 테마에서 수정하도록 했습니다.

예를 들어 metadata-browse 모듈의 경우 See all items with this value 같은 텍스트의 링크를 표시하게 되어 있는데 텍스트를 수정해 달라는 요구사항이 있었습니다. 모듈을 수정한다면 간단한 일이지만 모듈 업데이트를 할 수 없는 문제가 발생하기 때문에 테마 파일에 해당 텍스트를 수정하는 javascript 코드를 삽입했습니다. 이렇게 하면 모듈을 일일히 수정하지 않아도 테마 하나만 적용함으로써 디자인적인 요구사항을 충족할 수 있게 됩니다.

수정된 소스코드는 [Neo-Inspiration / centerrow](https://github.com/Neo-Inspiration/centerrow)에서 다운로드할 수 있습니다.

### metadata-browse

themes/centerrow/asset/js/centerrow.js 와 themes/centerrow/asset/css/sass/\_screen.scss를 수정했습니다.

텍스트 글자를 바꾸었습니다.

### Folksonomy

themes/centerrow/asset/css/sass/\_screen.scss를 수정했습니다.

키워드들이 태그가 많을수록 더 큰 글자로 표현하는 방식에서 더 짙은 색상으로 표시하는 방식으로 바꾸었습니다.

### Reference

themes/centerrow/asset/css/sass/\_screen.scss를 수정했습니다.

Line-height를 조정하여 줄 간격을 더 넓게 하고 키워드간의 좌우 여백을 늘려서 식별성을 개선했습니다.

## 모듈 생성

### Simple-Carousel

메인페이지에 표시할 목적으로 모듈을 새로 만들었습니다. 기존 모듈로는 디자인적인 요구사항을 충족하기 어렵다는 판단이 들어서 업로드된 item을 선택하면 페이지상에서 볼 수 있도록 만들어졌습니다.

[Neo-Inspiration/Omeka-S-SimpleCarousel](https://github.com/Neo-Inspiration/Omeka-S-SimpleCarousel)

## 사용된 모듈

### Search

Search 모듈은 크게 두가지 기능을 합니다.

1. 사용자가 검색어를 입력할 수 있는 입력 Form과 결과물을 보여주는 웹 페이지 제공
2. 내부 색인 어댑터를 관리하거나 외부 검색엔진 모듈과 연동시켜주는 기능

Search 모듈 설정 페이지에 접속하면 Indexes와 Pages를 설정하게 되어 있는데 이 중에서 Pages가 1번 기능을 담당하고 Indexes가 2번 기능을 담당합니다.

검색 기능은 우선 검색 엔진이 컨텐츠를 전부 색인해서 저장해 두었다가 추후 사용자가 입력한 문자열을 가지고 색인된 데이터를 보여주는 방식으로 동작하는데 색인 솔루션은 Omeka-S에 내장된 것도 있고 Solr같은 외부 프로그램을 사용할 수도 있습니다. Indexes는 이런 색인 솔루션을 관리하는 기능을 합니다.

Pages는 이 중에서 어떤 Index를 사용해서 어느 url을 통해 실질적인 검색 입력을 받을지를 설정할 수 있습니다.

Index가 Solr로 설정되어 있다면 Solr가 색인한 데이터를 검색하는 페이지가 되고, path가 solr/omeka로 설정되어 있고 site 주소가 http://114.202.196.101:11080/omeka/s/test/라면 http://114.202.196.101:11080/omeka/s/test/solr/omeka/solr/omeka/에서 검색어를 입력할 수 있습니다. Form은 Basic으로 선택될 경우 통상적인 HTML+CSS로 이루어진 웹페이지를 통해 검색어 입출력을 받게 되고 API로 선택하는 경우 웹프로그래머가 직접 검색 입출력을 구현할 수 있게 됩니다.

또한 Pages에는 facets와 sort를 설정할 수 있게 되어있는데 sort는 말 그대로 정렬 기준을 선택하는 것이고 facets는 검색기준을 선택할 수 있습니다.

### Solr

이 모듈은 Omeka-S가 Solr에 접속하는 방법을 설정합니다. Solr는 별도의 프로그램으로써 동작하고 HTTP 프로토콜로 통신할 수 있기 때문에 올바른 URL이 주어진다면 Solr가 Omeka-S에 접속하여 컨텐츠를 색인하고, Omeka-S가 Solr에 접속해서 색인된 데이터를 검색할 수가 있습니다.

## 외부 프로그램

### ImageMagick

ImageMagick은 이미지를 다루는 프로그램으로써 PHP와 연동하여 동작합니다. 이 프로그램을 필요로 하는 Omeka-S 모듈이 많기 때문에 거의 필수적입니다.

### Solr

실질적으로 컨텐츠를 색인하고 검색해주는 검색 엔진입니다. 앞서 말했듯이 프로그램 자체가 서버로써도 동작하기 때문에 상시 구동되며 다른 서버(Omeka-S)와 통신합니다.

지금까지 주로 설명한 기능은 색인이지만 검색 기능 역시 중요한데, 입력된 검색어를 형태소 단위로 분석하고 ~한다, ~은, ~는 과 같은 조사들을 불용어 처리하는 것이 검색 품질에 영향을 미치기 때문입니다.

불행해도 Solr는 국산 프로그램은 아니기 때문에 한글 검색기능이 좋지 못해서 이런 처리를 추가적으로 해줘야 합니다. 그래서 Solr에 아리랑 형태소 분석기를 설치하고 불용어 텍스트 파일을 업데이트했습니다.

## FTP 프로그램 사용

일반 사용자에게는 터미널 접속이 상당히 어려워보이는 일이기 때문에 처음 접속 이후에는 터미널로 서버에 접속하지 않고도 업무를 할 수 있도록 메뉴얼을 만들었습니다만 Sideload 모듈로 파일을 대량으로 업로드한다거나 설치된 모듈을 업그레이드 해야 하는 경우 서버 접속이 불가피합니다.

이때는 파일질라같은 FTP 프로그램을 사용하면 마우스 클릭만으로 서버상의 파일을 관리할 수 있습니다.

### File Sideload

File Sideload 모듈 설정에서 파일 업로드에 사용될 디렉터리 이름을 입력합니다. 이 디렉터리 이름은 Omeka-S 폴더 기준으로 정해지는데 만약 Omeka-S가 /www/html에 설치되어있고 Sideload directory는 'sideload/test'라면 /www/html/sideload/test 디렉터리에 파일을 업로드하면 됩니다.

FTP 프로그램으로 접속하면 왼쪽에는 현재 컴퓨터, 오른쪽에는 접속중인 서버 컴퓨터의 폴더 구조가 나오는데 오른쪽 아무 공간에다 마우스를 놓고 오른쪽 클릭을 하면 '디렉터리 만들기'라는 메뉴가 나타납니다. 이 메뉴를 클릭해서 설정에 명시된 경로에 해당하는 디렉터리를 만들고 업로드하고자 하는 파일을 드래그&드롭으로 업로드합니다.

### 모듈 업그레이드

모듈을 최초 설치할때에는 EasyInstall 모듈을 통해서 웹사이트에서 바로 설치할 수 있지만 새 버전이 있다는 알림이 있어서 다시 한번 EasyInstall로 설치하려고 이미 설치되어있다면서 업그레이드가 되지 않습니다.

이 경우 일단 module 페이지에 접근해서 업그레이드 하고자 하는 module을 deactivate합니다.

그 후 File Sideload 때와 마찬가지로 FTP로 접속해서 Omeka-s가 설치되어 있는 폴더로 이동합니다.

해당 폴더에는 module이라는 폴더가 있고 module 폴더 안에는 모듈마다 폴더들이 있습니다. 좀더 보기 쉽게 설명하자면 이렇습니다.

omeka 설치 폴더 > modules > 업그레이드할 모듈

업그레이드 하고자 하는 모듈명의 폴더를 삭제하고 웹사이트에 접속해서 처음에 설치할때와 마찬가지로 EasyInstall을 사용해서 모듈을 설치합니다.

1. 이 부분은 Omeka-S 소스를 직접 수정한 것이라 Omeka-S를 새로 설치하거나 업그레이드를 할 경우 사라지게 됩니다. [↑](#footnote-ref-22)