SSLAB 최종 보고서

2021년 12월 14일

2021년 2학기 소프트웨어시스템실습

담당교수: 최성종

이름 신범진

ieppersbj@office.uos.ac.kr

서울시립대학교 전자전기컴퓨터공학부

차 례

[1. 요약](file:///C:\Users\SpirA\Downloads\finalprojectSSL2021-2017440061.docx#_Toc90159140) 2

[2. 요구사항 도출 3](file:///C:\Users\SpirA\Downloads\finalprojectSSL2021-2017440061.docx#_Toc90159141)

[2.1. 모든 재난문자 표시 3](file:///C:\Users\SpirA\Downloads\finalprojectSSL2021-2017440061.docx#_Toc90159142)

[2.2. 세부 지역 선택 4](file:///C:\Users\SpirA\Downloads\finalprojectSSL2021-2017440061.docx#_Toc90159143)

[2.3. 페이지 이동 6](file:///C:\Users\SpirA\Downloads\finalprojectSSL2021-2017440061.docx#_Toc90159144)

[3. 설계 7](file:///C:\Users\SpirA\Downloads\finalprojectSSL2021-2017440061.docx#_Toc90159145)

[3.1. 사용 기술 7](file:///C:\Users\SpirA\Downloads\finalprojectSSL2021-2017440061.docx#_Toc90159146)

[3.2. 소프트웨어 구조도 8](file:///C:\Users\SpirA\Downloads\finalprojectSSL2021-2017440061.docx#_Toc90159147)

[3.3. 사용자 입력에 따른 흐름도 8](file:///C:\Users\SpirA\Downloads\finalprojectSSL2021-2017440061.docx#_Toc90159148)

[4. 구현 9](file:///C:\Users\SpirA\Downloads\finalprojectSSL2021-2017440061.docx#_Toc90159149)

[4.1. 코드의 주요 구조 9](file:///C:\Users\SpirA\Downloads\finalprojectSSL2021-2017440061.docx#_Toc90159150)

[4.2. 필수 라이브러리 21](file:///C:\Users\SpirA\Downloads\finalprojectSSL2021-2017440061.docx#_Toc90159151)

[5. 토론 22](file:///C:\Users\SpirA\Downloads\finalprojectSSL2021-2017440061.docx#_Toc90159153)

[6. 부록 23](file:///C:\Users\SpirA\Downloads\finalprojectSSL2021-2017440061.docx#_Toc90159157)

[7. 참고자료 28](file:///C:\Users\SpirA\Downloads\finalprojectSSL2021-2017440061.docx#_Toc90159158)

# 요약

코로나의 등장과 함께 수많은 재난 문자들이 개인에게 제공되고 있다. 현재는 위치를 근거로 하여 근처 지역의 문자만 받는 형식이지만 다른 지역의 재난 문자도 궁금하게 될 경우가 있을 것이다. 그렇기에 이번 프로젝트에서 재난 문자를 지역별로 분류하여 볼 수 있는 웹페이지를 만들었다.

제작에는 HTML5, JavaScript, AJAX, XML, JSON, jQuery 등의 기술을 사용했고 공공데이터 포털의 재난 문자 데이터 API와 Naver Map API를 받아서 사용했다.

# 요구사항 도출

## 모든 재난문자 표시

### 사용 시나리오

지도이(가) 표시된 사진

자동 생성된 설명

모든 재난문자를 보기 위해 아무것도 누르지 않은 상태로 검색을 누른다.

### 주요 요구사항

기능적 요구사항: 전국의 모든 재난문자들이 최근시간 순으로 최대 50건이 한 페이지에 나타나도록 했다.

비기능적 요구사항: 딜레이 없이 빠르게 재난문자들을 띄워주는 api를 찾아 사용한다.

## 세부 지역 선택

### 사용 시나리오

지도이(가) 표시된 사진

자동 생성된 설명 지도이(가) 표시된 사진

자동 생성된 설명

다음과 같이 시도선택을 클릭한 후 select box에서 원하는 지역을 선택한다. 그 후 세부 시군구도 필요하다면 클릭하여 선택하고 그렇지 않다면 검색을 클릭한다.

### 주요 요구사항

기능적 요구사항으로는 기존 전국의 재난문자가 나오던 것을 필터링 하여 원하는 시도 or 시군구의 재난문자만 나오도록 바꾸어 준다. 이때 지역 전체에 발송한 재난문자는 시군구를 선택하지 않았을 때만 나타나고 개별 시군구를 클릭했을 때에는 출력되지 않도록 설정했다.

## 페이지 이동

### 사용 시나리오

테이블이(가) 표시된 사진

자동 생성된 설명

재난문자를 출력할 때에 페이지가 나오도록 설정하였고 한 페이지씩 페이지를 이동하거나 원하는 페이지를 검색하여 이동하는 것이 가능하다.

### 주요 요구사항

기능적 요구사항으로는 우선 검색을 통해 재난문자를 출력했을 때에는 1페이지가 나오도록 하며 1페이지보다 앞의 페이지는 데이터가 없으므로 이동하는 것이 불가능하도록 했다.

# 설계

## 사용 기술

1. HTML5, CSS, JavaScript

웹페이지의 기본적인 구조, 디자인, 그리고 이들 간의 상호작용을 해주는 기술이다.

2. AJAX

웹서버에 접근하여 비동기적으로 데이터를 가져올 수 있도록 해준다. 여기서는 공공데이터 포털의 재난문자와 네이버 지도 API에서 데이터를 가져왔다.

3. XML, JSON

데이터를 저장하고 전달하기 위해 만들어진 언어로 기계뿐만 아니라 사람도 쉽게 읽는 것이 가능하다. Ajax를 통해 가져온 데이터들은 xml 또는 json의 형식으로 이루어져 있다.

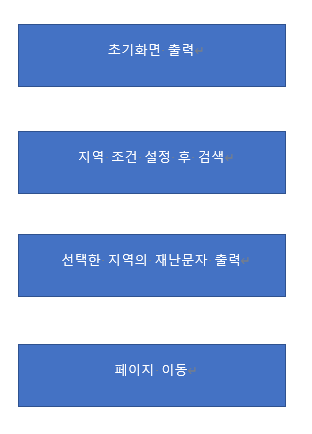
4. jQuery

자바스크립트를 쉽게 활용할 수 있도록 도와주는 오픈소스 기반의 JavaScript 라이브러리이다.

## 소프트웨어 구조도



## 사용자 입력에 따른 흐름도



# 구현

## 코드의 주요 구조



Div 태그를 사용하여 map이 생성될 틀을 만든다. 텍스트이(가) 표시된 사진

자동 생성된 설명

시도와 시군구를 선택할 수 있는 select box를 만든다. 각각에 id를 부여하여 밑에서 값을 받을 수 있도록 한다. 검색 버튼을 누르면 loadDoc 함수가 동작할 수 있게 한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Table 객체를 생성하여 후에 생성될 재난문자를 표로 받을 수 있도록 한다. 그후 페이지 변경을 할 수 있는 div를 만든다. 왼쪽 버튼을 누르면 이전 페이지로, 오른쪽 버튼을 누르면 다음 페이지로 갈 수 있도록 onclick에 함수를 연결한다. 또한 원하는 페이지로 이동 할 수 있는 find 함수를 가지고 있는 검색 버튼을 만들어 준다.

텍스트이(가) 표시된 사진

자동 생성된 설명

naver.maps에 있는 클라스인 Map 클라스의 객체인 map을 생성해준다.

Map에는 두개의 파라미터가 있는데

텍스트이(가) 표시된 사진

자동 생성된 설명

Mapdiv는 어디에 삽입할 것인가 즉 위에 만들었던 div에 삽입하는 것이고, mapOptions는 지도의 세부 옵션을 설정하는 것인데 mapOptions를 보면 많은 프로퍼티가 있는데 그중 사용된 프로퍼티 만을 확인해보면

zoom이란 줌 컨트롤 즉, 지도를 어느 정도로 확대할 것인지를 설정하는 것이고 mapTypeId는 지도의 초기 지도 유형 id를 선택해주는 것이고 center는 지도의 초기 중심 좌표를 어디로 설정하는 가이다. 참고로 여기서는 naver의 본사인 그린팩토리다.

HOME\_PATH에 네이버맵스 주소를 저장한다.



Event.once는 다음과 같다. 생성된 map 객체에 function() 내부의 이벤트를 발생시키는 것인데 funciton의 내부를 살펴보면 ajax가 사용되었다. Ajax란 클라이언트가 서버로 데이터를 요청하여 값을 받는 것으로 이를 이용하여 json이나 xml 형태의 필요한 데이터로 받을 수 있다. Ajax의 내부 프로퍼티를 보면 url은 data를 받아올 사이트의 link를 쓰는 것이고 dataType은 xml 형태인지 json 형태인지 선택하는 것이며 success는 받아오는데 성공했을 때에 실행되는 것으로 startDataLayer 함수가 실행된다.

Map이 load될 div와 재난정보가 출력될 table을 생성한다. 뒤의 부분은 1에서 설명한 것과 일치하므로 생략하겠다.

텍스트이(가) 표시된 사진

자동 생성된 설명

for문을 이용하여 총 17번을 반복하여 각각의 생성된 geojson을 위에서 생성한 공백리스트 regionGeoJson에 넣고 17번 반복하면 startDataLayer를 실행한다. 여기서 17번을 반복하는 이유는 대한민국의 행정구역이 총 17개이기 때문이다.

지도이(가) 표시된 사진

자동 생성된 설명

하나의 geojson만 실행해보면 다음과 같이 특정 행정구역을 표시해주고 있는 것을 알 수 있다

지도이(가) 표시된 사진

자동 생성된 설명

모든 geojson을 실행하면 다음과 같다.



tooltip이란 div 객체를 하나 생성한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Getpanes를 통해 지도 위에 표현 가능한 정보창을 만들 수 있게 되었다.

텍스트, 화면, 스크린샷이(가) 표시된 사진

자동 생성된 설명

1번과 같이 style을 설정해주는 것이다. 여기서 fillopacity, strokeopacity는 불투명도를 나타내고 #ff000은 red와 동일하다. 이를 적용하면

지도이(가) 표시된 사진

자동 생성된 설명

다음과 같이 바뀌게 된다. 내부색이 더 진한 이유는 중첩이 되었기 때문이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

이는 regionGeoJson 리스트 내부의 모든 geojson에 addGesoJson을 통해 데이터를 받아오는 것이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

마우스를 올리게 되면 json에서 area1의 값을 가져오게 된다. 여기서 area1이란 각 행정구역의 이름이다. 이를 regionName에 저장한 후, tooltip을 생성하는데 위치는 마우스의 위치를 offset.x와 offset.y로 받아서 사용하고 div 내에는 regoinName의 값을 넣는다. 또한, overrideStyle을 이용하여 올라간 영역의 스타일을 바꿔준다. 이를 적용하면

지도이(가) 표시된 사진

자동 생성된 설명

다음과 같이 툴팁 div가 생성되어 해당 행정구역의 이름을 나타내고 불투명도와 경계선의 굵기가 바뀐 것을 확인할 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

마우스를 떼어내면 tooltip을 사라지게하고 안에 들어있던 regionName의 값을 지운다. 그리고 revertStyle을 사용하여 스타일을 원래대로 되돌린다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Click을 하면 loadDoc 함수가 생성되는데 이때 파라미터는 ‘area1’의 값, 즉 행정구역의 이름을 사용한다.

텍스트, 화면, 스크린샷이(가) 표시된 사진

자동 생성된 설명loadDoc 함수를 살펴보면 XMLHttprequeset의 xhr 객체를 생성하고 url과 queryParams를 만들어준다. a에는 area1의 값 즉, 행정구역의 값을 받고 b에는 area2 즉 세부지역의 값을 받는다. 여기서 location\_name에 b의 값이 없으면 a를 넣고 아니면 b의 값을 넣어준다. 이를 통해 선택한 지역의 재난문자만 받는 것이 가능해진다. 이후 xhr객체에 ulr+queryParams를 합친 링크를 넣은 후 open을 통해 서버로 ajax 요청을 보낸다. 이때 서버로부터 응답을 받는데 onreadystatechange란 readystate의 값이 바뀔 때마다 호출하는 것으로 readystate의 값이 4가 되면 myfunction을 실행한다. 여기서 readystate의 값은

텍스트이(가) 표시된 사진

자동 생성된 설명

다음과 같은 의미를 갖는다. 즉, readystate==4라는 것은 응답할 준비가 완료되었음을 의미한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Myfunction을 통해 xml 데이터에서 값을 가져와 표를 생성한다.

텍스트, 화면, 스크린샷이(가) 표시된 사진

자동 생성된 설명

Before과 after는 pageNum을 바꿔서 페이지를 이동할 수 있게 해주는 함수들이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Find 함수는 원하는 페이지로 이동할 수 있게 해준다.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

Area 함수는 area1에서 행정구역을 선택했을 때 그 행정구역의 하위구역인 시군구를 띄워 선택할 수 있도록 하는 함수이다.

## 필수 라이브러리

1. Web Dynamic Map

Naver map API 로 웹에서 지도를 볼 수 있게 해준다.

2. 공공데이터 포털 API

재난문자 발송 현황을 보여준다.

3. jQuery

행정구역의 경계를 그리는데 사용한다.

# 토론

## 도출한 요구사항의 완성도

전체적으로 도출한 요구상항에 대해서는 비기능적으로나 기능적으로나 문제없이 잘 작동하는 것을 확인할 수 있었다. 페이지 이동 같은 경우에는 이전페이지는 1보다 작게는 가지 못하게 설정하였으나 전체 데이터의 양을 파악하지 못해 최대 페이지르 넘어가는 경우를 막는 것은 하지 못했다.

# 부록

## 전체 소스코드

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

# 참고자료

1. 공공데이터포털

<https://www.data.go.kr/data/15091495/openapi.do>

2. 네이버 map API

<https://navermaps.github.io/maps.js.en/>

3. AJAX

<https://www.w3schools.com/js/js_ajax_intro.asp>

4. XML

<https://www.w3schools.com/xml/default.asp>

5. JSON

<https://www.w3schools.com/js/js_json_intro.asp>

6. jQuery

https://www.w3schools.com/js/js\_jquery\_selectors.asp