예제 코드 분석

2021년 11월 30일

2021년 2학기 소프트웨어시스템실습

담당교수: 최성종

변형조

2017440061@office.uos.ac.kr

서울시립대학교 전자전기컴퓨터공학부

차 례

[1. GeoJSON + NAVER map API 예제 1](#_Toc89286279)

[1.1. 사용한 클래스 1](#_Toc89286280)

[1.2. 실습코드 분석 9](#_Toc89286281)

[2. 지역별 재난문자 실습 13](#_Toc89286282)

[2.1. 사용한 클래스 13](#_Toc89286283)

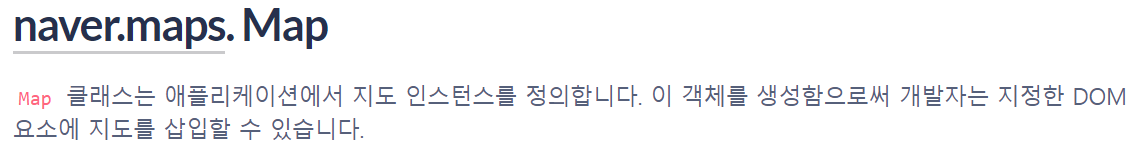
[2.2. 실습코드 분석 14](#_Toc89286284)

# GeoJSON + NAVER map API 예제

## 사용한 클래스

### naver.maps.Map

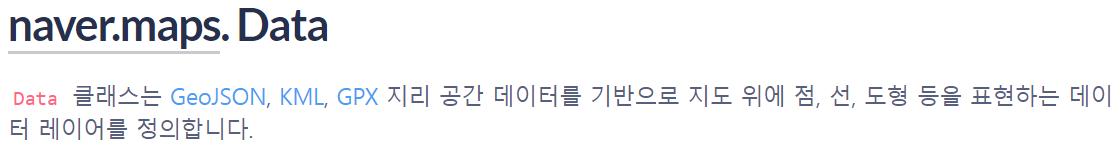
#### 클래스의 주된 목적



#### 사용한 프로퍼티

1)map.data

**1.1) 프로퍼티의 주된 기능**



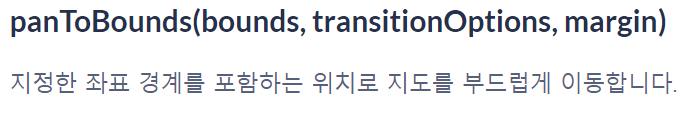
**1.2) 실습 코드 내에서 기능**

addGeoJson, setStyle, addListener 메소드로 각각 GeoJson데이터를 저장, 스타일 지정, 이벤트 리스너 등록을 한다. 자세한 내용은 naver.maps.Data에 나와있다.

#### 사용한 메소드

1) panToBounds(bounds, transitionOptions, margin)

**1.1) 프로퍼티의 주된 기능**



**1.2) 실습 코드 내에서 기능**

텍스트이(가) 표시된 사진

자동 생성된 설명

더블클릭을 하면 해당 bound의 지점으로 이동하도록 하였다.

### naver.maps.LatLng

#### 클래스의 주된 목적

텍스트이(가) 표시된 사진

자동 생성된 설명

**1.1) 실습 코드 내에서 기능**

텍스트이(가) 표시된 사진

자동 생성된 설명

map객체를 생성할 때 지도가 비출 중심점 좌표를 등록해 준다. Map option객체로 naver.maps.Map의 파라미터로 입력

텍스트이(가) 표시된 사진

자동 생성된 설명

### naver.maps. MapTypeRegistry

#### 클래스의 주된 목적

텍스트이(가) 표시된 사진

자동 생성된 설명

#### 사용한 메소드 설명

1) set(mapTypeId, mapType)

**1.1) 메소드 주된 기능**

텍스트이(가) 표시된 사진

자동 생성된 설명

**1.2) 실습 코드 내에서 기능**

텍스트이(가) 표시된 사진

자동 생성된 설명

mapTypeId로 ‘normal’을 입력하고 있다.

### naver.maps.Event

#### 클래스의 주된 목적

텍스트이(가) 표시된 사진

자동 생성된 설명

#### 사용한 메소드 설명

1) once(target, eventName, listener)

**1.1) 메소드 주된 기능**

텍스트이(가) 표시된 사진

자동 생성된 설명

**1.2) 실습 코드 내에서 기능**

텍스트이(가) 표시된 사진

자동 생성된 설명

‘init\_stylemap’이벤트가 발생하면 ajax로 정보를 불러온 후 startDataLayer함수를 호출한다.

2)addListener

**2.1)메소드의 주된 기능**

텍스트이(가) 표시된 사진

자동 생성된 설명

**2.2) 실습 코드 내에서 기능**

텍스트, 스크린샷, 화면이(가) 표시된 사진

자동 생성된 설명

“click”과 “dbclick”등의 이벤트가 발생할 때 data에 수행할 이벤트를 등록한다.

### naver.maps. Data

#### 클래스의 주된 목적

텍스트이(가) 표시된 사진

자동 생성된 설명

#### 사용한 메소드

1) addGeoJson(geojson, autoStyle)

**1.1)메소드의 주된 기능**

텍스트이(가) 표시된 사진

자동 생성된 설명

**1.2) 실습 코드 내에서 기능**

텍스트이(가) 표시된 사진

자동 생성된 설명

AJAX로 받아온 GeoJSON정보를 startDataLayer를 이용하여 data에 등록한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

2) setStyle(style)

**2.1)메소드의 주된 기능**

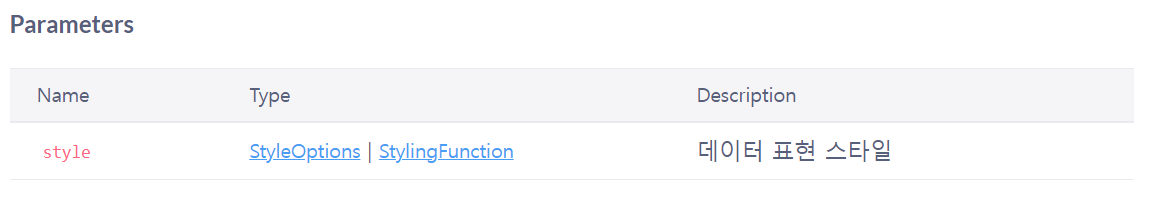
텍스트이(가) 표시된 사진

자동 생성된 설명

**2.2) 실습 코드 내에서 기능**

텍스트이(가) 표시된 사진

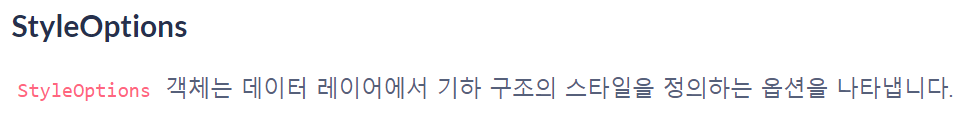
자동 생성된 설명



여기서는 StylingFunction을 사용하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명



3) overrideStyle(feature, style)

**3.1)메소드의 주된 기능**

텍스트이(가) 표시된 사진

자동 생성된 설명

**3.2) 실습 코드 내에서 기능**

텍스트이(가) 표시된 사진

자동 생성된 설명

마우스가 올라가면 데이터 표현 스타일을 재정의 한다.

4) revertStyle(feature)

**4.1)메소드의 주된 기능**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**4.2) 실습 코드 내에서 기능**

텍스트이(가) 표시된 사진

자동 생성된 설명

마우스가 자리에서 나가면 스타일을 원래대로 되돌린다.

### naver.maps.GroundOverlay

#### 클래스의 주된 목적

텍스트이(가) 표시된 사진

자동 생성된 설명

#### 사용한 메소드

1) getBounds()

**1.1)메소드의 주된 기능**

텍스트이(가) 표시된 사진

자동 생성된 설명

**1.2) 실습 코드 내에서 기능**

텍스트이(가) 표시된 사진

자동 생성된 설명

더블클릭된 영역의 경계를 얻는다.

## 실습코드 분석

### 전체적인 흐름도

처음에 Naver map AIP와 AJAX를 불러오고 기본적인 설정을 한다. 이후 Naver map API의 MAP클래스를 활용하여 map객체를 만든다. map객체는 로딩 이벤트 발생 시 일회성으로 수행하는 함수를 호출한다. 함수는GeoJSON으로 생성된 정보를 등록해 주고 각각의 요소에 대해 스타일을 적용시킨다. 또 map객체와 GeoJSON요소에 대해 여러가지 이벤트가 발생할 때 어떻게 변할지 eventListener를 등록해 준다.

### 각 요소별 설명 & 흐름도

#### API호출

텍스트, 스크린샷, 모니터, 화면이(가) 표시된 사진

자동 생성된 설명

브라우저가 실행되면 Naver map AI, AJAX를 불러오고 기본적인 설정을 한다.

#### 객체 생성

텍스트이(가) 표시된 사진

자동 생성된 설명

map을 그릴 공간을 div에 지정해 주고 map객체를 생성한다. document.getElementById로 map을 그릴 공간 div를 찾고 어떤 속성으로 map을 그릴지 정보를 준다. zoom값은 18이고 mapType는 normal이다. getTypeIds()로 어떤 mapTypeId가 있는지 확인할 수 있다. 지도가 처음 시작 할 때 보여지는 중심점의 좌표는 naver.maps.LatLng클래스를 이용하여 지정하고 현재 좌표는 네이버 본사로 설정 되어 있다.

#### GeoJSON정보 불러오기

텍스트이(가) 표시된 사진

자동 생성된 설명

GeoJSON정보를 불러올 경로를 정하고 init\_stylemap이벤트가 발생하면 AJAX로 정보를 받아온다. 정보를 받아오는 것이 성공하면 startDataLayer함수를 실행한다.

#### startDataLayer함수

텍스트이(가) 표시된 사진

자동 생성된 설명

받아온 GeoJSON정보를 지도에 등록하고 초기 스타일을 설정한다. map.data.setStyle의 argument로 들어간 함수에서 color가 있으면 feature.getProperty(‘color’)로 색을 받아오고 아니면 red로 지정한다. return에서 내부에 채우는 색, 테두리 색, 테두리 두께, icon여부 등을 지정한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

이후 필요한 eventListener를 등록한다. ‘click’을 하면 해당 개체의 ‘isColorful’속성이 true로 바뀌어 기존 red에서 다른 색으로 color속성이 바뀐다. ‘dbclick’을 하면 bound를 알아내어 해당 위치로 지도의 초점이 이동하게 된다.

‘mouseover’을 하면 테두리가 굵어진다. icon은 핀의 모양을 바꾸게 된다.

마지막으로 ‘mouseout’을 하면 ‘mouseover’을 하기 이전에 설정되었던 스타일대로 되돌아간다.

# 지역별 재난문자 실습

## 사용한 클래스

GeoJSON + NAVER map API 예제에 공통적으로 사용된 클래스는 생략하였다.

### naver.maps.Map

#### 클래스의 주된 목적

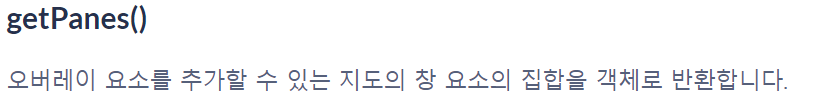
텍스트이(가) 표시된 사진

자동 생성된 설명

#### 사용한 메소드

**1) getPanes()**

**1.1) 메소드 주된 기능**

****

**1.2) 실습 코드 내에서 기능**



지도의 창 요소 집합체에 툴팁을 적용할 때 사용되었다.

## 실습코드 분석

### 각 요소별 설명

#### API호출

텍스트이(가) 표시된 사진

자동 생성된 설명

Naver map API를 호출하고 기본 설정을 한다. JQuery라이브러리를 사용하기 위해 받아온다.

#### 스타일 및 형태 지정

텍스트이(가) 표시된 사진

자동 생성된 설명

표를 출력할 때 필요한 스타일을 설정한다. 지도를 표시할 div, 표를 표시할 table위치를 작성한다.

#### Map 객체 생성

텍스트, 스크린샷, 화면이(가) 표시된 사진

자동 생성된 설명

GeoJSON정보를 불러올 주소를 설정하고 map객체를 생성한다. GeoJSON정보가 도 단위 행정구역의 경계이므로 zoom을 7로 하여 대한민국 전국이 보이도록 하였다.

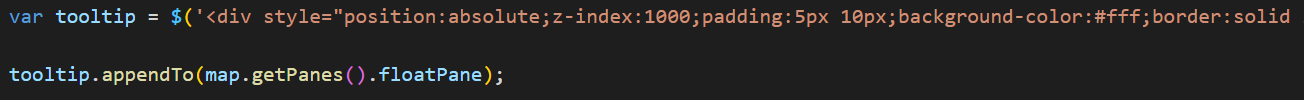
#### Map생성 및 정보 입력

텍스트이(가) 표시된 사진

자동 생성된 설명

지역마다 for를 돌면서 GeoJSON정보를 받아온다. 지역에 따른 정보를 받아올 url을 만들기 위해 keyword를 교체한다. 받아온 각각의 정보는 regionGeoJson배열에 저장한다. 그리고 모두 받아졌으면 startDataLayer함수를 호출하여 스타일을 지정하고 이벤트를 등록한다.

#### Tooltip 등록



tooltip이 표시될 구역과 스타일을 지정하고 map객체에 tooltip을 추가한다.

#### startDataLayer

텍스트이(가) 표시된 사진

자동 생성된 설명

초기 스타일과 포커스때 스타일을 지정한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

regionGeoJson에 담긴 정보를 반복적으로 map에 등록한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

여러가지 이벤트 리스너를 등록한다. ‘click’일 때 우선 클릭된 곳의 feature를 받아오고 feature중 지역명 ‘area1’을 추출하여 region에 저장한다. 이후 loadDoc함수를 region을 인자로 호출한다. ‘mouseover’가 발생하면 ‘click’과 마찬가지로 지역 이름을 얻어서 tooltip으로 표시한다. 동시에 style을 override하여 불투명하고 테두리가 두껍게 만든다. 마지막으로 ‘mouseout’이 발생하면 tooltip을 숨기고 ‘mouseover’이전의 스타일로 바꾼다.

#### 재난문자 목록 불러오기

텍스트이(가) 표시된 사진

자동 생성된 설명

공공 데이터 포털에서 재난문자 정보를 받아오기 위해 필요한 형식의 query를 만들어 AJAX의 XMLHttpRequest클래스의 객체 xhr을이용하여 정보를 받아온다. xhr의onreadystatechange메소드로 상태가 변하면 함수를 수행한다. 함수 내용에서 readyState 4는 요청이 끝났고 응답을 받은 상태이고 status 200은 “OK”상태이다. 해당 상태이면 정보가 다 받아진 것이므로 myFunction을 호출한다.

#### 표 출력하기

텍스트이(가) 표시된 사진

자동 생성된 설명

공공 데이터 포털에서 전송해준 xml데이터에서 날짜, 지역, 내용을 추출하기 위해 getElementsByTagName을 사용하여 “create\_data”, “location\_name”, “msg”를 찾는다. 찾은 내용을 for문을 돌면서 표에 한 행 씩 추가한다. 문자열로 이루어진 표에 대한 html을 Dom object를 사용하여 innerHTML에 작성한다.