**SISENSE MAP CHART WIDGET**

1. **요약**

D3(v4)의 TOPOJSON기능을 활용하여 한국 지형정보를 가진 TOPOJSON(.json)으로 지도를 렌더링한 후, {지역명, 목표매출, 실제매출} 데이터를 이용해 해당 위치에 차트를 렌더링하고, 해당 차트 클릭 시 지역단위 DASHBOARD FILTERING을 수행하는 WIDGET입니다.

* 지도 이미지를 이용해 간단히 렌더링하고, 경/위도 좌표와 이미지 좌표의 비율을 계산해서 차트를 렌더링할 수도 있지만, TOPOJSON에는 지역의 범위와 경계 등의 정보가 포함되어 있어 원하는 지역만 특별한 스타일로 렌더링하는 방식 등, 지도를 원하는대로 자유롭게 렌더링할 수 있습니다.

( 본 WIDGET은 지도 렌더링의 목적으로만 TOPOJSON을 이용 )

지도이(가) 표시된 사진

자동 생성된 설명

<WIDGET의 출력>

1. **REFERENCE**

[한국 TOPOJSON](https://github.com/southkorea/southkorea-maps) : 3가지 다른 LEVEL로 지역을 나눈 TOPOJSON 제공 (GITHUB)

[D3 TOPOJSON 활용](https://m.blog.naver.com/PostView.nhn?blogId=jhc9639&logNo=221195987416&proxyReferer=https:%2F%2Fwww.google.com%2F) : D3 로 TOPOJSON을 활용하는 방법 설명 (BLOG)

[SISENSE DASHBOARD FILTER](https://sisense.dev/reference/js/dashboard/dashboard-filters.html) : DASHBOARD FILTERING을 수행하는 함수 설명

[SISENSE METADATA-ITEM](https://sisense.dev/reference/js/metadata-item.html) : FILTER정보를 가진 객체 설명

[SISENSE JAQL SYNTAX](https://sisense.dev/reference/jaql/) : FILTER정보를 가진 객체 설명

1. **설명**

보다 자세한 내용은 소스코드의 주석을 참고바랍니다.

**processResult**

buildQuery에서 넘어온 데이터들을, 차트출력에 알맞은 데이터로 가공합니다.

[{name:서울, lat:…, lon:…, data1:…, data2:…}, {name:부산, …}, …]

와 같은 형식으로, 각각의 데이터들을 key, value로 구성하고 배열 형태로 가공합니다.

* **d3.data(), .enter() 함수들을 쉽게 활용하기 위한 형태**

**render**

JQUERY getScript함수를 이용해 D3 와 TOPOJSON 자바스크립트를 웹에서 불러오고, d3.json()함수를 이용해 한국 지형 TOPOJSON(.json)파일 또한 웹에서 불러옵니다.

* **SISENSE PLUGIN로컬 폴더에 D3(v4) 자바스크립트와 한국지형 TOPOJSON(.json)파일을 저장한 후 불러오는 방식을 시도해 보았지만,**
* SISENSE자체에서 사용하는 D3 때문에, 로컬에서 불러온 D3(v4) 자바스크립트는 사용 불가.(version4 문법 사용시 오류 발생)
* .json파일은 PLUGIN폴더에 존재해도, PLUGIN실행 시 웹 서버 상에 같이 올라가지 않아서 사용 불가.

1. D3 TOPOJSON기능을 활용해 메르카토르 투사법으로 한국 지도를 출력합니다.

(자세한 코드 설명 REF. [D3 TOPOJSON 활용](https://m.blog.naver.com/PostView.nhn?blogId=jhc9639&logNo=221195987416&proxyReferer=https:%2F%2Fwww.google.com%2F) 참고)

1. d3.data(), .enter() 함수를 이용하여 processResult에서 가공된 데이터들을 한 번에 차트로 렌더링 합니다.
2. 차트가 클릭(선택)되면, **Filtering(d) 함수**가 호출되고, 이 함수에서 DASHBOARD FILTER를 적용한 후, DASHBOARD를 REFRESH합니다.

* 나머지 차트의 OPACITY(투명도)를 감소시켜 해당 차트를 강조하는 부분은, REFRESH후에 다시 RENDER가 호출될 때 render함수에서 시행됩니다.

**Filtering(d) 함수**에서 시행할 수 있지만, REFRESH후에 투명도 조절 애니메이션이 나타나는 것이 사용자가 보기에 더 편하다고 판단했습니다.