

```

import React, { useState } from 'react'
import { Card, Row, Col, Badge, Grid } from 'antd';
import PropTypes from 'prop-types'
import {
  ComposableMap,
  Geographies,
  Geography
} from 'react-simple-maps';
import ReactTooltip from 'react-tooltip'
import WorldMap from 'assets/maps/world-countries-sans-antarctica.json'
import utils from 'utils'

const { useBreakpoint } = Grid;
const geoUrl = WorldMap;
const mapColor = '#F5F4F6';
const hoverPercentage = -10;

const getHighlightedRegion = (name, data) => {
  if(data.length > 0 || name) {
    for (let i = 0; i < data.length; i++) {
      const elm = data[i];
      if(name === elm.name) {
        return elm.color
      }
    }
    return mapColor
  }
  return mapColor
}

const getRegionHoverColor = (name, data) => {
  if(data.length > 0 || name) {
    for (let i = 0; i < data.length; i++) {
      const elm = data[i];
      if(name === elm.name) {
        return utils.shadeColor(elm.color, hoverPercentage)
      }
    }
    return utils.shadeColor(mapColor, hoverPercentage)
  }
  return utils.shadeColor(mapColor, hoverPercentage)
}

const getRegionValue = (name, data) => {
  if(data.length > 0 || name) {
    for (let i = 0; i < data.length; i++) {
      const elm = data[i];
      if(name === elm.name) {
        return `${elm.name} - ${elm.value}`
      }
    }
    return ''
  }
  return ''
}

const MapChart = ({ setTooltipContent, data, mapSource, mapType }) => {
  return (
    <ComposableMap style={{transform: `${mapType === 'world' ? 'translateY(20px)' : 'none'}`}} data-tip="" height={380} projectionConfig={{ scale: 145 }}>
      <Geographies geography={mapSource}>
        {{{ geographies }}} =>
          geographies.map(geo => {
            const geoName = mapType === 'world' ? geo.properties.name : geo.properties.NAME_1
            return (
              <Geography
                key={geo.rsmKey}
                geography={geo}
                onMouseEnter={() => {
                  setTooltipContent(getRegionValue(geoName, data));
                }}
                onMouseLeave={() => {
                  setTooltipContent("");
                }}
                fill={getHighlightedRegion(geoName, data)}
                stroke="#D6D6DA"
                style={{
                  hover: {
                    fill: getRegionHoverColor(geoName, data),
                    outline: "none"
                  }
                }}
              />
            )
          })
      </Geographies>
    </ComposableMap>
  )
}

const Map = props => {
  const { data, mapSource, mapType } = props
  const [content, setContent] = useState("");
  return (
    <>
      <MapChart data={data} mapSource={mapSource} mapType={mapType} setTooltipContent={setContent} />
      <ReactTooltip>{content}</ReactTooltip>
    </>
  );
}

const renderDataList = data => {
  const list = data.map(elm => (
    <div className="d-flex align-items-center justify-content-between mb-3" key={elm.name}>
      <div>
        <Badge color={elm.color} />
        <span className="text-gray-light">{elm.name}</span>
      </div>
    </div>
  ))
}

```

```

    </div> <span className="font-weight-bold text-dark">{elm.value}</span>
  ))
  return list
}

export const RegiondataWidget = props => {
  const { data, mapSource, mapType, title, content, list } = props
  const isMobile = !utils.getBreakPoint(useBreakpoint()).includes('lg')
  return (
    <Card bodyStyle={{padding: 0}}>
      <Row>
        <Col xs={24} sm={24} md={24} lg={7} className="border-right">
          <div className="d-flex flex-column p-3 justify-content-between h-100">
            <div>{title} && <h4 className="font-weight-bold">{title}</h4></div>
            <div>{content}</div>
            <div>{list ? list : renderDataList(data)}</div>
          </div>
        </Col>
        <Col xs={24} sm={24} md={24} lg={17}>
          <div className="d-flex flex-column justify-content-center" style={{minHeight: isMobile ? 200 : 435 }}>
            <div className="p-3 w-100" >
              <Map data={data} mapSource={mapSource} mapType={mapType}/>
            </div>
          </div>
        </Col>
      </Row>
    </Card>
  )
}

RegiondataWidget.propTypes = {
  title: PropTypes.string,
  data: PropTypes.array,
  mapSource: PropTypes.object,
  mapType: PropTypes.string,
  content: PropTypes.element,
  list: PropTypes.element
}

RegiondataWidget.defaultProps = {
  data: [],
  mapSource: geoUrl,
  mapType: 'world'
};

export default RegiondataWidget

```