Label Studio 코드분석

데이터서비스플랫폼팀 안다용사원

1. 라이브러리 및 구성분석
   1. react를 사용한다. 기본적으로 react 구문법인 class component 를 기본으로 사용하고 있음.
   2. 라벨스튜디오에서 사용중인 상태관리 라이브러리 MST(mobx-state-tree)는 typescript 사용을 권장하므로 typescript 3.0 이상을 사용해야함.
   3. MST와 Mobx-react-lite(mobx-react) 가 같이 사용중인것으로 확인됨.
   4. 캔버스 라이브러리(react-konva)
2. style, css
   1. 라벨스튜디오의 style 프레임워크와 css preprocessor는 Ant Design과 SCSS를 사용하고 있으며, stylus가 사용되고 있음.

* ant Design
* npm i styled-components antd @ant-design/icons
* SCSS
* stylus

1. 상태관리, 전역데이터
   1. 라벨스튜디오는 상태관리 라이브러리로 mobx-state-tree, mobx-react를 사용하고 있음.

* mobx-state-tree
  + npm install mobx mobx-state-tree –save
  + yarn add mobx mobx-state-tree
* mobx-react
  + npm i mobx-react
  + yarn add mobx-react
* mobx
  + npm i mobx
  + yarn add mobx
* mobx-react-lite
  + npm i mobx-react-lite
  + yarn add mobx-react-lite
  1. 그 외 Context 등이 사용되고 있는 것으로 확인됨.

1. 캔버스 라이브러리
   1. react-konva라는 캔버스 그래픽 js 라이브러리를 사용.
2. component 및 js파일 별 코드 분석

* label studio의 코드 양이 상당하고, 기능도 많아 모든 코드 분석에는 어려움이 있으나, 우선적으로 라벨링이라는 중요 기능에 대해서 코드분석을 실시함.
* 나머지 코드들에 대해서도 지속적으로 분석하며 업데이트 할 예정.
  1. src -> component -> imageView -> ImageView.js
     1. 어노테이션툴에서 사용하는 기본적인 image 라벨링 기능 코드가 담겨있음.
     2. const Regions : 캔버스에 그려진 라벨링
     3. class ImageView -> handleMouseDown : 이 메서드를 통해 마우스 좌클릭시 해당 라벨링에 대한 좌표값을 출력할 수 있음.
     4. handleMouseDown -> e.target.attrs
        1. target : region 라벨 종류
        2. x : x값이 있는 경우 rectangle은 왼쪽 상단 point의 좌표 값. key point의 경우 해당 지점 좌표값.
        3. y : y값이 있는 경우 rectangle은 왼쪽 상단 point의 값. key point의 경우 해당 지점 좌표값.
        4. points : points가 있는 경우 polygon은 각 꼭지점의 좌표 값
        5. height : 높이
        6. width : 넓이
  2. src -> tools -> index.js
     1. Export 목록에서 brush, ellipse 등 라벨링 tool 메뉴를 추가, 삭제 할 수 있음.
  3. src -> tools -> Brush.js
     1. Brush 기능으로 자유로운 라벨링이 가능하나 좌표기능을 제공하지 못하므로 기능 삭제.
  4. src -> tools -> Ellipse.js
     1. Ellipse 기능으로 타원형의 라벨링이 가능함. 현재는 사용할 계획이 없는 기능으로 삭제.
  5. src -> tools -> Erase.js
     1. brush 기능을 사용하여 라벨링을 그렸을 경우에만 사용할 수 있는 지우개 기능이다.
     2. 최초에 brush를 선택하여 그린 직후에만 사용이 가능하며, Rectangle, Polygon 등 다른 라벨링 기능을 선택하면 그 이후론 지워지지 않는다. delete를 이용하여 삭제해야 한다.
  6. src -> tools -> KeyPoint.js
     1. 점 하나를 찍을 수 있는 라벨링 기능이다. 그려진 라벨링을 선택을 하면 해당 지점의 좌표를 볼 수 있다.
  7. src -> tools -> Polygon.js
     1. 임의의 점을 선으로 연결하여 다각형을 만들 수 있는 라벨링 기능이다. 그려진 라벨링을 선택을 하면 해당 도형의 꼭지점에 대한 좌표를 볼 수 있다.
  8. src -> tools -> Rect.js
     1. 사각형 모양의 라벨링을 할 수 있는 기능이다. 그려진 라벨링을 선택을 하면 사각형의 네 꼭지점의 위치에 해당하는 좌표를 볼 수 있다.
  9. src -> tools -> Selection.js
     1. 마우스 드래그로 생성된 라벨링들을 다중 선택할 수 있다.
  10. src -> tools -> Zoom.js
      1. <Tool> -> IconHandTool : Zoom-in 기능으로 확대된 캔버스영역을 마우스 클릭한 상태로 끌어서 화면이동을 할 수 있다.
      2. <Tool> -> IconMagnifyTool : item.handleZoom() 으로 Zoom-in, Zoom-out 기능을 생성 할 수 있다.
  11. src -> component -> Entities 폴더 -> Entities.js
      1. 테스트용 checkbox를 제어하기 위한 테스트 버튼 생성
  12. src -> component -> Entities -> RegionItem.js
      1. Regionitem 내부에 checkbox를 넣은 코드
  13. src -> component -> ErrorMessage.js
      1. error 발생 시 에러메시지 출력
  14. src -> TopBar -> Actions.js
      1. 상단에 메뉴를 생생해줌.
  15. src -> TopBar -> Annotations.js
      1. 새로운 어노테이션을 생성할 수 있으며 만들어진 어노테이션 목록을 펼쳐볼수있다.