**React编程思想：**

组件化：react里没有页面这一概念，全部由组件化实现，而且组件会被定义在不同的文件之中。

Ruiji\_list2页面组件：

**RuijiList2.tsx**

这个文件定义了一个页面级别的组件，负责进行页面级别的数据处理。

React 的函数组件本质上是一个 **函数**，但它通常是作为一个 **变量** 来定义和使用的。

const RuijiList2: React.FC = () => {

这个定义的过程，首先声明了一个变量 RuijiList2，它来自一个React.FC的typescript类型注解，React.FC是React.Functioncomponent的简写，它是React提供的一个泛型接口，用于定义React函数组件。

也就是通过React的泛型接口定义一个自己的React函数组件。

它的值等于：()=>{}，这是一个箭头函数，（）表示函数的参数列表，{}是函数体，决定了这个组件渲染时会显示什么内容。

**useState函数用于创建和管理状态**

通过 useState 返回的数组进行解构赋值，你可以轻松地在 React 函数组件中管理状态：

* **状态变量**：用于在组件的整个生命周期中保存和读取状态值。
* **状态更新函数**：用于更新状态值，并自动触发 React 组件的重新渲染。

const [selectedIds, setSelectedIds] = useState<number[]>([]);

userState是一个钩子函数，定义了selectIds，setSelectIds是一个状态更新函数。用setSelectedIds函数可以对selectIds进行更改，这是一个[变量，函数]形式，其中setSelectedIds

可以接受一个回调函数用于改变selectIds的值

setSelectedIds((prevSelectedIds: number[]) => {

        const newSelectedIds = isChecked

          ? [...prevSelectedIds, id]

          : prevSelectedIds.filter((selectedId) => selectedId !== id);

      // setSelectAll(newSelectedIds.length === tableData.length);

      return newSelectedIds;

    });

这里就是通过把一个回调函数传递给setSelectedIds来更新selectIds的值。其setSelectedIds()里装的就是这个回调函数

 useContext, useEffect

这两个是react的hook，一个是用于在函数组件中访问上下文，useEffect用于执行副作用的操作。副作用指的是不属于组件渲染逻辑的操作。

useEffect(()=>{},[])方括号为依赖项，花括号为副作用代码

useEffect(() => {

    if (questionnaire == null) return;

    setColumns(getColumns([], [], true, false, false, 0, ()=>{}));

    setConditionText(`データセット名 = ${questionnaire.name}`);

  }, [questionnaire]);

这里的questionnaire实际指的是页面上蓝色的部分

useEffect(() => {

    setTableData((prevTableData) =>

      prevTableData.map((row) => ({

        ...row,

        isSelected: selectAll || selectedIds.includes(row.id),

      })),

    );

  }, [selectAll, selectedIds]);

const [tableData, setTableData] = useState<AnswerTextTableRow[]>([]);

这里是一个很巧妙的地方，通过useState将[tableData, setTableData]的绑定来更新状态变量，在useEffect函数里，会因为绑定，所以会自动把tableData的值传递给setTableData里回调函数的参数preTableData

对于 prevTableData.map的理解：首先，这里的row指的不是行号，而是每一行的数据，在getTableData函数里可以看到为

answerTexts.map((answerText) => ({

    id: answerText.id,

    answer: answerText.answer,

    question: answerText.question,

    ...arrayToObject(

      answerText.categories.map((category, index) => ({

        key: `category${index + 1}`,

        value: category.value,

      })),

    ),

    llm: answerText.llm,

    manualName: answerText.manualName,

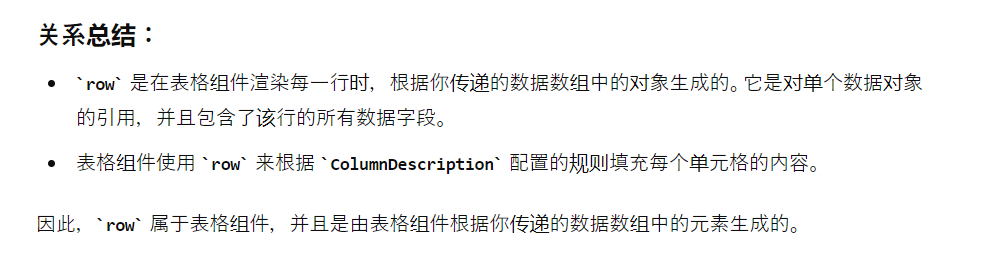
    importDatetime: answerText.importDatetime,

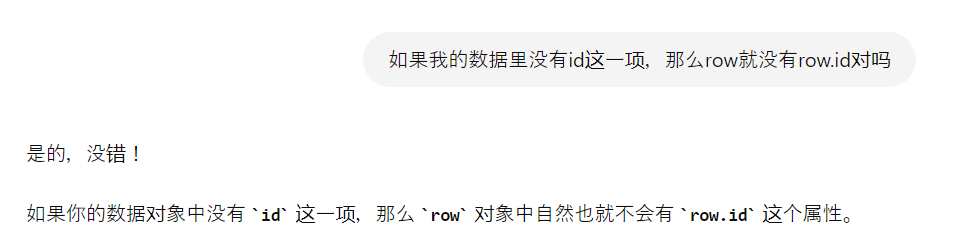
    memo: answerText.memo,

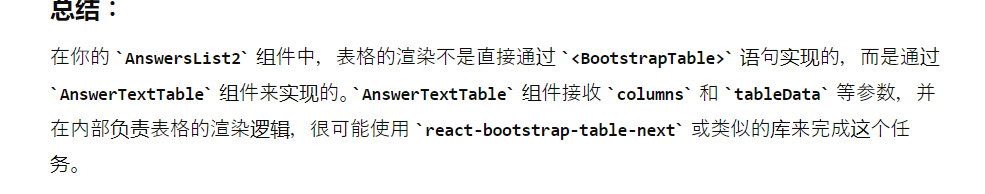
  }));

getColumns([], [], true, false, false, 0, ()=>{})

这个函数是为了返回有哪些列要被显示出来的







**ruiji\_tabledata\_logic.tsx**

这个文件定义了一些关于ruiji的表格的逻辑处理函数。