**ETable设计思路:**

**Antd 中table的常用属性：**

**updateSelectedItem 更新表格选中数据**

**bordered: 是否展示外边框和列边框**

**columns: 表格列的配置描述（表头）**

**dataSource: 数据数组**

**pagination: 分页器**

current 当前页数

pageSize 每页条数

total 数据总数

**rowKey: 表格行key值**

**rowSelection: 是否可被选择**

selectedRowKeys指定选中项的key数组，需要和onChange进行配合

**title: 表格标题**

**onChange: 分页、排序、筛选变化时触发**

**onRow: 设置行属性**

**设计思路：**

**1、先创建一个组件**

**2、在render中只添加封装表格的函数**

render(){ return ( <div> { this.tableInit() } </div> ) }

**3、在class组件中对表格的封装 方法名为：tableInit**。

**a、**首先我们要从父元素给我们的数据中解构出rowSelection属性(是否开启选择，单选还是多选)；还要解构出selectedRowKeys属性(表格列的key值集合)

**b、**然后给一个初始值rowSelection为单选框{type:'radio',selectedRowKeys}

**c、**再之后进行判断，判断是否开启选择的那一列，判断是单选还是多选，如果没有开启给rowSelection设置为false，如果开启那就解构出rowSelection的值，如果值为'checkbox'，就给它的type属性设为'checkbox'，值为'radio'，就给它的type属性设为radio

**d、**设置这些前提条件后对组件进行封装；当调用一个函数返回我们需要的组件，执行函数的return 方法开始完成组件的封装

**e、**我们需要给组件<Table />添加父元素传过来的属性(一般采用解构的方式{...this.props}) 、一些必要属性和组件特别的属性，如bordered，rowSelection和onRow属性，

**f、**像rowSelection,我们需要判断选择列是单选还是多选

**g、**像onRow属性，它的函数有两个参，一个是列数据，一个index索引，我们对列操作可能不同，像我们要操做列数据，我们要判断它是否有可选(单选、多选)，要是不可操作，则reutrn 空；

**h、**要是可操作就可以调用函数将这两个参数，通过函数调用对这列进行操作

《==============================================================》

：注：单选多选在页面上的展示

《

我们将这两个参数代入函数内，获取父元素rowSelection的值，然后进行判断，该值是多选还是单选，

a、如果是多选，我们就需要获取父元素的列key的集合（selectedRowKeys）、列数据(selectedItem)和选中id的集合(selectedIds)，然后在进行判断是否有选中的列；

如果有选中的列，我们需要查看点击的列是否已在选中，如果是就将其三个值移除，如果不是，则将点击列的id赋值给selectedIds,index赋值给selectedRowKeys,列数据赋值给selectedItem, 添加到选中id的数组里,

如果没有选择的列，就将点击列的id赋值给selectedIds,index赋值给selectedRowKeys,列数据赋值给selectedItem, 添加到选中id的数组里,

b、如果是单选，我们就将将点击列的id赋值给selectedIds,index赋值给selectedRowKeys,列数据赋值给selectedItem最后调用updateSelectedItem函数将这三个值当参数传入updateSelectedItem函数：{

判断是多选还是单选，多选就this.setState({三个参})，单选就是selectedIds和seelctedItem

}

》

**分页器pagination**

我们会在刚获得数据的时候，运行封装函数调用pagination（数据，函数可以刷新页面和储存当前页数）

在封装函数中返回参数

pagination(

onChange:回调重新调用刷新，

current: 当前页数，

pageSize: 每页条数，

total: 数据总数，

showTotal：数据总量和当前数据顺序，

showQuickJumper: 是否可以快速跳转至某页）