12、在react中配置less

查看文档：<https://www.jianshu.com/p/7fe9e435eb7c>，如果还不生效，查看src/index.js文件，是否是import "antd/dist/antd.min.css";如果是，改为import "antd/dist/antd.css";

11、react项目antd css样式生效问题？

yarn add antd时，我们在页面里引入button按钮时是不生效的，那该怎么办呢？

①（全部加载）：就是在我们自己less文件里面输入 @import "~antd/dist/antd.less"; 然后，每个css样式都要引入那个含有这句话的less文件；

②（按需加载）：babel-plugin-import插件

输入：npm i babel-plugin-import -D，再次执行，会报错，说缺少css-loader

输入：yarn add css-loader，再次执行，会报下面的错：

./node\_modules/antd/es/style/index.less (./node\_modules/css-loader!./node\_modules/less-loader/dist/cjs.js!./node\_modules/antd/es/style/index.less)    这个错误，这是我们的less版本不兼容问题

输入：npm install less@2.7.2，再次执行，OK!

1、使用G2时，报“**Please specify the container for the chart!**”

翻译为：请指定图表的容器！

我这边是使用antV来可视化，和echarts差不多，首先会有一个div来承载展示内容

，下面当然会有js来渲染展示，我这边的主要内容就是js在渲染展示内容的时候，指定的div的id不正确

2、JavaScript 获取数组的最后一个元素

①index取值

args[args.length - 1]

②pop方法

args.pop()

注意： pop方法会删除args最后一个元素,并返回

3、React的context就是一个全局变量，可以从根组件跨级别在React的组件中传递。

Reactcontext的API有两个版本，React16.2之前的是老版本的context，16.3之后的是新版本的context。

案例：

注意：如果想在16.2以下版本使用16.3以上的接口，可以引入create-react-context的npm包，此方法可以适用于其他API的向下兼容。

4、使用react-codemirror(页面编辑工具)总结：

①github：<https://github.com/uiwjs/react-codemirror>

②CodeMirror使用说明书：<https://blog.csdn.net/jlu_lei/article/details/80259697>

③Editor：用refs组件获取到CodeMirror对象后，要使用codeMirror.editor获取到editor对象，然后对编辑器内容进行操作。（注意：不要覆写onChange方法）

5、JSON处理：

①json格式化校验

import jsonlint from 'jsonlint-mod'

jsonlint.parse(json数据)

②json格式美化

import json\_beautifier from 'csvjson-json\_beautifier/json\_beautifier'

json\_beautifier(json数据, {可配置参数});

③json树形结构：react-json-view

**<https://www.cnblogs.com/soyxiaobi/p/9713814.html>**

**6、**node 内存溢出的解决方案

①解决方案一：通过**package.json**中的 **"pro"**加大内存(适用于react自建项目，脚手架搭建不可使用)

/\* package.json 文件 \*/
"scripts": {
"pro": "node --max\_old\_space\_size=9000 build/build.js "
}

②解决方案二：使用 **increase-memory-limit**插件，增加node服务器内存限制

/\* package.json 文件 \*/

"devDependencies": {

"increase-memory-limit": "^1.0.6",

"cross-env": "^5.2.0",

},

"scripts": {

/\* 添加 fix-memory-limit \*/

"fix-memory-limit": "cross-env LIMIT=4096 increase-memory-limit"

}

1. 执行npm run fix-memory-limit（只需执行一次即可）
2. 重启项目即可

**7、**对比两个json的差异工具：json-diff

导包：npm install json-diff

使用：

var jsonDiff = require('json-diff')
console.log(jsonDiff.diffString({ foo: 'bar' }, { foo: 'baz' }));

console.log(jsonDiff.diff({ foo: 'bar' }, { foo: 'baz' }));

**8**、复杂数组以及对象的深度拷贝

链接：https://blog.csdn.net/mozhi111/article/details/79454556

方法一：利用JSON

var newObj=JSON.parse(JSON.stringify(str));

方法二:递归（略）

**9、**JS实现并集，交集和差集

var set1 = new Set([1,2,3]);

var set2 = new Set([2,3,4]);

并集

let union = new Set([...set1, ...set2]);

交集

let intersect = new Set([...set1].filter( x => set2.has(x)));

差集

let difference = new Set([...set1].filter(x => !set2.has(x)));

**10、**集合排序

sort/sorter: (a, b) => {

**// 数字**

**if** (a.type > b.type) **return** -1;

**if** (a.type < b.type) **return** 1;

**// 字母**

**if** (a.name.length > b.name.length) **return** -1;

**if** (a.name.length < b.name.length) **return** 1;

**// 汉字 - 或者其他情况**

**return** b.xxx - a.xxx;

}