import React,{Component} from ‘react’

import {

StyleSheet,

Text,

View,

AppRegistry,

TouchableOpacity,

TouchableHighlight,

FlatList,

ScrollView,

SectionList,

Image,

AlertIOS

}from ‘react-native’

class test01 extends Component{

render(){

return (

<View style={styleList.container}>

<Text>我是一个text</Text>

</View>

)

}

}

const styleList=StyleSheet.creat({

container{

backgroundColor:’red’,

}

})

AppRegistry.registerComponent(‘test01’,()=>test01)

flexBox:

flexDirection:row|column (决定是横向排列还是竖立排列)

justifyContent:flex-end （决定在一个方向上是怎么排列）

alignItems:flex-end （决定每个块头是上面对齐还是下面对齐还是中间对齐）

flexWrap:wrap (决定如果一个方向放不下就换行)

alignSelf:flex-start (脱离统一设定样式，独自对齐)

flex 默认为 0 1 auto （flex-grow flex-shrink flex-basis）

宽度=父盒子宽度\*（flexGrow/sum(flexGrow)）

flex:1(若父盒子里无其他子盒子，此设置为占满父盒子的样式设定)

var Dimensions=require(‘Dimensions’)

var {width,height,scale}=Dimensions.get(‘window’);//scale为分辨率

<TextInput style={styles.inputType} keyboardType={‘number-pad’} multiline={true} clearButtonMode={‘always’}/>

## Touchable 系列组件

<View>

<TouchableOpacity onPress={this.renderPress} activeOpacity={0.5}><Text>我是按钮</Text></TouchableOpacity>

</View>

<View>

<TouchableHighlight onPress={this.renderPress}>

<Image

style={styles.button}

source={require('./myButton.png')}

/>

</TouchableHighlight>

</View>

renderPress(){

AlertIOS.alert(333)

}

TouchableNativeFeedback:

TouchableHighlight ;

TouchableWithoutFeedback:

TouchableOpacity

ScrollView FlatList:并不立即渲染全部，渲染可见部分，可进行增删，必要的属性：data renderItem

<View>

<FlatList

data={[

key:’aaa’,

key:’bbb’,

key:’ccc’

]}

renderItem={({item})=><Text>{item.key}</Text>}

></FlatList>

</View>

如果是要渲染一组需要分组的数据 SectionList

属性 sections renderItem

renderSectionHeader keyExtractor

<View>

<SectionList

sections={

{title:’0’,data:[‘aaa’,’ewww’,’rrrr’]},

{title: 'J', data: ['Jackson', 'James', 'Jillian', 'Jimmy', 'Joel', 'John', 'Julie']},

]}

renderItem={({item})=><Text>{item}</Text>}

renderSectionHeader={({section})=><Text>{section.title}</Text>}

keyExtractor={(item.index)=>index}

>

</SectionList>

</View>

Fetch

fetch(‘http://22XXXX’,{

method:’POST’,

headers:{

Accept:’application/json’,

‘conten-type’:’application/json’

},

body:JSON.stringify({

firstName:’yourvalue’,

secondParam:’yourotherview’

})

})











