**React-native中svg字体图标的使用**

1. 需要引入的库：

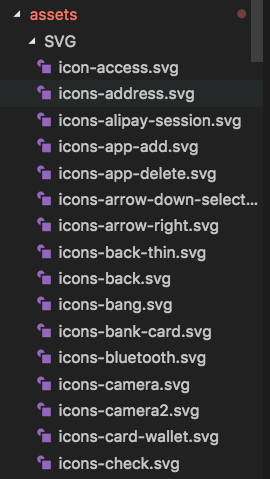
yarn add react-native-svg --save

react-native link react-native-svg

yarn add react-native-svg-uri –save

1. 将所需要的svg图片存入assets->SVG文件夹下，同级别建立脚本文件getSvg.js

SVG文件夹如下：



脚本文件getSvg.js内容如下：

// getSvg.js

var fs = require('fs');

var path = require('path');

const svgDir = path.resolve(\_\_dirname, './SVG');

// 读取单个文件

function readfile(filename) {

return new Promise((resolve, reject) => {

fs.readFile(path.join(svgDir, filename), 'utf8', function(err, data) {

if (err) reject(err);

resolve({

[filename.slice(0, filename.lastIndexOf('.'))]: data,

});

});

});

}

// 读取SVG文件夹下所有svg

function readSvgs() {

return new Promise((resolve, reject) => {

fs.readdir(svgDir, function(err, files) {

if (err) reject(err);

Promise.all(files.map(filename => readfile(filename)))

.then(data => resolve(data))

.catch(err => reject(err));

});

});

}

// 生成js文件

readSvgs().then(data => {

let svgFile = 'export default ' + JSON.stringify(Object.assign.apply(this, data));

svgFile = svgFile.replace(/\\n/g,'');

svgFile = svgFile.replace(/>[ ]+</g,"><");

fs.writeFile(path.resolve(\_\_dirname, './Svg.js'), svgFile, function(err) {

if(err) throw new Error(err);

})

}).catch(err => {

throw new Error(err);

});

打开新的终端，进入项目的assets文件下，执行node getSvg.js，生成Svg.js文件，Svg.js文件结构如下：

export default {

'svgName1': 'xmlData1...',

'svgName2': 'xmlData2...',

...

}

1. 同级别建立Svgs.js，封装Svg Component。

Svgs.js内容如下：

// Svgs.js

import React, { Component } from 'react';

import SvgUri from 'react-native-svg-uri';

import svgs from './Svg';

export default class Svgs extends Component<SvgProperties,void>{

render() {

const {

icon,

color,

size,

style,

} = this.props;

let svgXmlData = svgs[icon];

if (!svgXmlData) {

let err\_msg = `没有"${icon}"这个icon，请下载最新的icomoo并 npm run build-js`;

throw new Error(err\_msg);

}

return (

<SvgUri

width={size}

height={size}

svgXmlData = {svgXmlData}

fill={color}

style={style}

/>

)

}

}

同级别，建立SingleSvg.js文件（实现一个图片中包含多种颜色效果），SingleSvg.js文件内容如下：

import React,{Component} from 'react';

import {Dimensions} from 'react-native';

import Svg,{Path,Circle}from 'react-native-svg';

const { width, height } = Dimensions.get('window');

export default class SingleSvg extends Component{

render(){

return(

<Svg height={height \* 0.5} width={width \* 0.5} viewBox="0 0 100 100">

<Path

fill={this.props.pathColor}

class="cls-1"

d="M8.41,19.22H3.89a2.64,2.64,0,0,1-2.82-2.77V4.53a2.79,2.79,0,0,1,2.79-3H20.15a2.82,2.82,0,0,1,2,.89,2.72,2.72,0,0,1,.75,2V16.41a2.67,2.67,0,0,1-2.78,2.81H15.64L13,21.83a1.39,1.39,0,0,1-.95.4,1.34,1.34,0,0,1-.94-.39ZM3.86,2.66A1.68,1.68,0,0,0,2.17,4.49v12a1.56,1.56,0,0,0,.45,1.2,1.58,1.58,0,0,0,1.24.45h5l.16.16,2.86,2.78a.23.23,0,0,0,.17.07.25.25,0,0,0,.18-.08l3-2.93h5a1.56,1.56,0,0,0,1.19-.45,1.6,1.6,0,0,0,.45-1.23v-12a1.7,1.7,0,0,0-.45-1.27,1.68,1.68,0,0,0-1.23-.53Z"

/>

<Circle

fill={this.props.circleColor}

class="cls-2"

cx="19.5"

cy="4.5"

r="4.5"

/>

</Svg>

)

}

}

若需要实现特别样式，如圈中包含有数字的效果，则只需在<Svg>的<Circle>标签的同级别下面添加<Test>标签。

<Svg height={height \* 0.5} width={width \* 0.5} viewBox="0 0 80 80">

<Path

fill={this.props.pathColor}

class="cls-1"

d="M8.41,19.22H3.89a2.64,2.64,0,0,1-2.82-2.77V4.53a2.79,2.79,0,0,1,2.79-3H20.15a2.82,2.82,0,0,1,2,.89,2.72,2.72,0,0,1,.75,2V16.41a2.67,2.67,0,0,1-2.78,2.81H15.64L13,21.83a1.39,1.39,0,0,1-.95.4,1.34,1.34,0,0,1-.94-.39ZM3.86,2.66A1.68,1.68,0,0,0,2.17,4.49v12a1.56,1.56,0,0,0,.45,1.2,1.58,1.58,0,0,0,1.24.45h5l.16.16,2.86,2.78a.23.23,0,0,0,.17.07.25.25,0,0,0,.18-.08l3-2.93h5a1.56,1.56,0,0,0,1.19-.45,1.6,1.6,0,0,0,.45-1.23v-12a1.7,1.7,0,0,0-.45-1.27,1.68,1.68,0,0,0-1.23-.53Z"

/>

<Circle

fill={this.props.circleColor}

class="cls-2"

cx="19.5"

cy="4.5"

r="4.5"

>

</Circle>

<Text fill='#fff' x='16.5' y='6.0' fontSize='5'>12</Text>

</Svg>

效果如：

1. 运行时，引入Svgs.js.和SingleSvg.js

具体代码如下：

import React, {Component} from 'react';

import {StyleSheet, View} from 'react-native';

import Svgs from './assets/Svgs';

import SingleSvg from './assets/SingleSvg';

type Props = {};

export default class App extends Component<Props> {

render() {

return (

<View style={styles.container}>

<Svgs icon='icon-access' size='30' color='green'></Svgs>

<Svgs icon='icons-pay' size='40' color='red'></Svgs>

<Svgs icon='icons-reset' size='50' color='blue'></Svgs>

<Svgs icon='icons-me-help' size='50' color='orange'></Svgs>

<Svgs icon='icons-bang' size='40' color='blue'></Svgs>

<Svgs icon='icons-tab-store-outline' size='50' color='red'></Svgs>

<Svgs icon='icons-back' size='50'></Svgs>

<Svgs icon='icons-weixin' size='50' ></Svgs>

<Svgs icon='icons-time' size='50' color='red'></Svgs>

<Svgs icon='icons-selected' size='50' color='yellow'></Svgs>

<Svgs icon='icons-write' size='50' color='yellow'></Svgs>

<Svgs icon='icons-up' size='50' color='yellow'></Svgs>

<Svgs icon='icons-share' size='50' color='red'></Svgs>

<SingleSvg pathColor='#000000' circleColor='red'></SingleSvg>

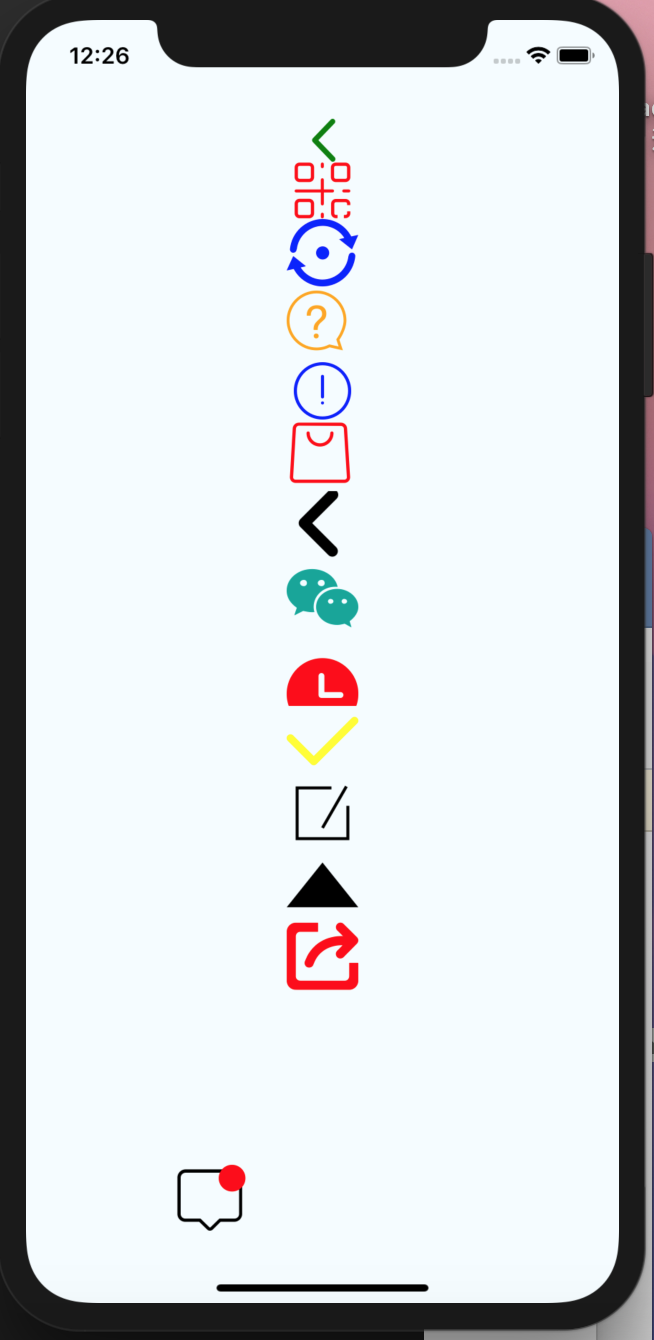
</View>

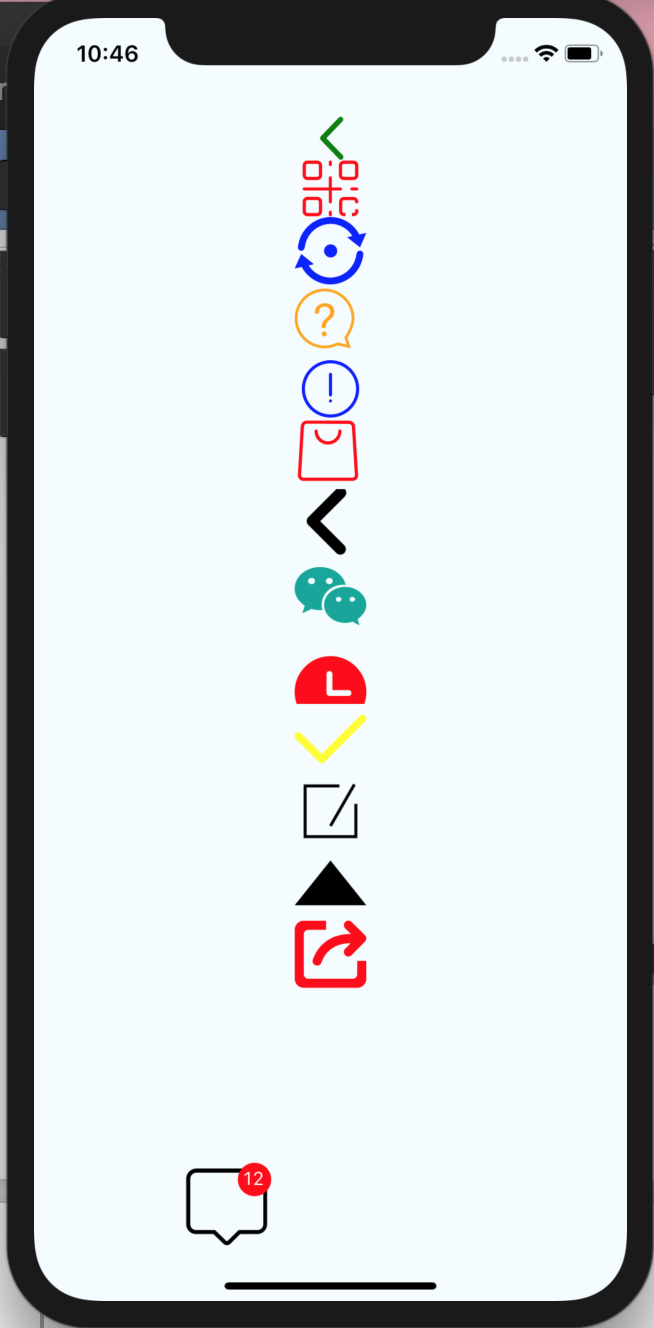
);

}

}

1. 运行结果如下：





注：原svg格式的文件中必须有fill，否则不能修改颜色，如：icons-back.svg图标