“‘{r} APP.js 路由核心模块

```
import React from 'react';
import './App.css';
import Lists from './components/list';
import OrderPage from './components/orderPage';
import Detail from './components/detail';
import Progress from './components/progress';
import Disclaimer from './components/disclaimer';
import ActionView from './components/Action/action';
import Health from './components/Health/health';
import HealthCondition from './components/Health/health-condition';
import BreedView from './components/Breed/breed';
import GenealogicalTreeView from './components/Breed/genealogical-tree';
import BreedDetailedView from './components/Breed/Breed-tab/breed-detail';
import Advice from './components/Health/advice';
import ScrollToTop from './components/scrollToTop';
// import wx from 'weixin-js-sdk';
// import axios from 'axios';

import { BrowserRouter as Router, Route } from "react-router-dom";

// const ua = window.navigator.userAgent.toLowerCase()
// 如果不在微信浏览器内，微信分享也没意义了对吧？这里判断一下
// if (ua.indexOf('micromessenger') < 0) return false
// 最好在在 router 的全局钩子里调用这个方法，每次页面的 URL 发生变化时，都需要重新获取
// 如果你的 router 使用的是 hash 形式，应该不用每次都重新获取微信分享参数

// axios({
//   method: 'get',
//   url: '/api/signature',

// })
// .then(response => {
//   console.log('none')
// })

// wx.config({
//   debug: true, // 开启调试模式，调用的所有 api 的返回值会在客户端 alert 出来，若要
//   appId: '', // 必填，公众号的唯一标识
//   timestamp: '', // 必填，生成签名的时间戳
//   nonceStr: '', // 必填，生成签名的随机串
//   signature: '',// 必填，签名，见附录 1
//   jsApiList: [
//     'onMenuShareAppMessage',
//     'onMenuShareTimeline'
//   ] // 必填，需要使用的 JS 接口列表，所有 JS 接口列表见附录 2
// })

// wx.ready(() => {
//   //分享给朋友
//   wx.onMenuShareAppMessage({
//     title: '检测报告', // 分享标题
//     desc: '这是我家主人的检测报告', // 分享描述
//     link: window.location.href, // 分享链接，该链接域名或路径必须与当前页面对应的公
//     imgUrl: '', // 分享图标
//     type: '', // 分享类型,music、video 或 link，不填默认为 link
//     dataUrl: '', // 如果 type 是 music 或 video，则要提供数据链接，默认为空
//     success: function () {
//       // 用户确认分享后执行的回调函数
//     },
//     cancel: function () {
//       // 用户取消分享后执行的回调函数
//     }
//   })

//   //分享到朋友圈
//   wx.onMenuShareTimeline({
//     title: '检测报告', // 分享标题
//     link: '这是我家主人的检测报告', // 分享链接，该链接域名或路径必须与当前页面对应
//     imgUrl: window.location.href, // 分享图标
//     success: function () {
//       // 用户确认分享后执行的回调函数
//     },
//     cancel: function () {
//       // 用户取消分享后执行的回调函数
//     }
//   })
// })
```

```
if(!global.consoleSwitch){
  console.log = ()=>{console.clear();}
}

const App = () => (
  <Router>
    <ScrollToTop>
      <div>
        {/* <Route exact path="/:type" component={OrderPage} /> */}
          <Route exact path="/:type/" component={Lists} />
          <Route path="/:type/detail" component={Detail} />
          <Route path="/:type/progress" component={Progress} />
          <Route path="/:type/disclaimer" component={Disclaimer} />
          <Route path="/:type/action" component={ActionView} />
          <Route path="/:type/breed" component={BreedView} />
          <Route path="/:type/genealogical_tree" component={GenealogicalTreeView} />
          <Route path="/:type/health" component={Health} />
          <Route path="/:type/advice" component={Advice} />
          <Route path="/:type/condition" component={HealthCondition} />
          <Route path="/:type/breed_detailed" component={BreedDetailedView} />
      </div>
    </ScrollToTop>
  </Router>
);

export default App;
```

detail.js 检测报告页

```
import React, { Component } from 'react';
import './detail.css';
import Header from './Header/header';
import {Link} from 'react-router-dom';
import _ from 'lodash';
import axios from 'axios';
import strain from '../Api/犬类品系.json';
import simpleDogHealth from '../Api/犬类单基因疾病.json';
import simpleCatHealth from '../Api/猫类单基因疾病.json';
import complexDogHealth from '../Api/犬类复杂疾病.json';
import dogReport from '../img/icon_introduction_green.svg';

class Detail extends Component {
    /**
     * 这是报告详情页面
     *
    */
    state = {
        showDog: window.location.pathname.split('/')[4],
    }

    componentDidMount() {
        // console.log(this.props)
        let title = null;
        let name = null;
        let text = null;
        let body = null;
        title = this.props.location.state.status;
        name = this.props.location.state.name;
        text = this.props.location.state.item;
        if (name === '行为') {
            text = '性格行为';
        }
        body = this.props.location.state.body;
        let status = null;
        if (typeof(text) === 'string') {
            switch (name) {
                case '行为':
                    status = text;
                    break;
                case '淀粉酶':
                    status = text.split(' , ')[0];
                    break;
                case '儿童安全性':
                    status = text.split(' , ')[0];
                    break;
                default:
                    break;
            }
        }
        axios({
            method: 'get',
            url: `../data/db/交互式报告素材.json`,
            headers: {
                'Content-Type': 'application/json',
                'Cache-Control': 'max-age=86400'
            }
        })
        .then(response => {
            // console.log('hhhh',response.data[name])
            let content = null;
            let mapObj = response.data[name];
            if (name === '行为') {
                mapObj['养育建议'] = mapObj['养育建议'][this.props.location.state.it
            }
            if (this.props.location.state.strain) {
                _.map(this.props.location.state.strain, (item, index) => {
                    mapObj[item] = strain[item]['犬种特点'];
                })
            } else if (this.props.location.state.disease) {
                _.map(this.props.location.state.disease, (item, index) => {
                    if (simpleDogHealth[item]) {
                        mapObj[item] = simpleDogHealth[item]['诊疗建议'];
                    } else if (complexDogHealth[item]) {
                        mapObj[item] = complexDogHealth[item]['诊疗建议'];
                    } else if (simpleCatHealth[item]) {
                        mapObj[item] = simpleCatHealth[item]['诊疗建议'];
                    }
                })
            }

            let result = _.map(mapObj, (item, index) => {
                // console.log('检查',name,index,text)
                let display = {display: 'block', whiteSpace: 'pre-wrap'};
                if (typeof(item) === 'string') {
                    if (text === '复杂疾病性状') {
                        if (index === '单基因疾病') {
                            display = {display: 'none', whiteSpace: 'pre-wrap'};
                        }
                    }
                    if (index.indexOf('年') !== -1) {
                        let secondContent = _.map(body, (seconditem, secondIndex) =>
                            if (secondIndex === index) {
                                return (
                                    <li key={index} style={{position:'relative'}}>
                                        <p className='dog-print' style={display}><img
                                        <span style={display}>{'养育建议'}</span>
                                        <div
                                            style={display}
                                            dangerouslySetInnerHTML={{__html: item}}
                                        >
                                        </div>
                                    </li>
                                )
                            }
                        })
                        return secondContent;
                    } else if (index.indexOf('学') !== -1) {
                        console.log('jjjiiii', index)
                        let secondContent = _.map(body, (seconditem, secondIndex) =>
                            if (secondIndex === index) {
                                return (
                                    <li key={index} style={{position:'relative'}}>
                                        <p className='dog-print' style={display}><img
                                        <span style={display}>{'推荐游戏'}</span>
                                        <div
                                            style={display}
                                            dangerouslySetInnerHTML={{__html: item}}
                                        >
                                        </div>
                                    </li>
                                )
                            }
                        })
                        return secondContent;
                    } else {
                        return (
                            <li key={index} style={{position:'relative'}}>
                                <p className='dog-print' style={display}><img src={do
                                <span style={display}>{index}</span>
                                <div
                                    style={display}
                                    dangerouslySetInnerHTML={{__html: item}}
                                >
                                </div>
                            </li>
                        )
                    }
                }
                if (typeof(item) === 'object') {
                    // console.log(item['复杂疾病进阶研究'])
                    if (text === '复杂疾病性状') {
                        localStorage.setItem('key',2)
                        if (item['复杂疾病进阶研究']) {
                            return (
                                <li key={index} style={{position:'relative'}}>
                                    <p className='dog-print' style={display}><img src
                                    <span style={display}>{index}</span>
                                    <div
                                        style={display}
                                        dangerouslySetInnerHTML={{__html: item['复杂
                                    >
                                    </div>
                                </li>
                            )
                        }
                    }
                    if (item[status]) {
                        return (
                            <li key={index} style={{position:'relative'}}>
                                <p className='dog-print' style={display}><img src={do
                                <span style={display}>{index}</span>
                                <div
                                    style={display}
                                    dangerouslySetInnerHTML={{__html: item[status]}}
                                >
                                </div>
                            </li>
                        )
                    }
                }
            })
            let linkDetail = '';

            switch (name) {
                case ' 行为':
                    linkDetail = <Link className="link-detail" to={`/${this.props.mat
                    break;
                case ' 健康':
                    linkDetail = <Link className="link-detail" to={`/${this.props.mat
                    break;
                case ' 品系':
                    linkDetail = <Link className="link-detail" to={`/${this.props.mat
                    break;
                default:
                    linkDetail = ''
                    break;
            }
            if (title === ' 详情') {
                content =
                    <ul className="list">
                        {result}
                        {linkDetail}
                    </ul>
                title = `${name}${title}`
            } else {
                let item = null;
                let index = null;
                if (response.data[name][' 简介']) {
                    index = ' 简介';
                    item = response.data[name][' 简介']
                }
                if (response.data[name][' 测算依据信息']) {
                    index = ' 测算依据信息';
                    item = response.data[name][' 测算依据信息']
                }
                if (response.data[name][' 淀粉消化能力知识点']) {
                    index = ' 淀粉消化能力知识点';
                    item = response.data[name][' 淀粉消化能力知识点']
                }
                if (response.data[name]['犬智商相关知识点']) {
                    index = '犬智商相关知识点';
                    item = response.data[name]['犬智商相关知识点']
                }
                if (text === '单基因疾病') {
                    if (response.data[name]['单基因疾病']) {
                        index = '单基因疾病';
                        item = response.data[name]['单基因疾病']
                    }
                } else {
                    if (response.data[name]['复杂疾病']) {
                        index = '复杂疾病';
                        item = response.data[name]['复杂疾病']['复杂疾病进阶研究']
                    }
                }
                content =
                    <ul className="list">
                        <li>
                            <span>{index}</span>
                            <div>{item}</div>
                        </li>
                        <li><a href="javascript:void(0);">点击购买解锁功能</a></li>
                    </ul>
            }
            this.setState({
                name,
                title,
                content,
                text
            })
            // console.log(content)
        })
        .catch(error => {
            // console.log('错误',error)
        });
    }

    render() {
        return (
            <div className="detail">
                <Header
                    tab="genome"
                    dog_name={this.props.location.state.dog_name}
                    showAction={this.props.location.state.showAction}
                    showBreed={this.props.location.state.showBreed}
                    showHealth={this.props.location.state.showHealth}
                />
                <div className='condition-wapper health-warpper'>
                    <div className='condition-header'>
                        <Link to={`/${this.props.match.params.type}`}> 检测报告</Link
                        <i>></i>
                        <span>
                            {
                                this.state.name === ' 健康' || this.state.name === ' 
                                ?
                                this.state.text
                                :
                                this.state.name
                            }
                        </span>
                    </div>
                    <div className="detail-content">
                        {this.state.content}
                    </div>
                    <ul className="render-list">
                    </ul>
                </div>
            </div>
        );
    }
}

export default Detail;
```

list.js 报告详情

```
import React, { Component } from 'react';
import {Link} from 'react-router-dom';
import './list.css';
import ImgDog from '../img/infoDog.jpg';
import Header from './Header/header';
import Share from './Share/share';
import ShareHtml from './Share/shareHtml';
import _ from 'lodash';
import axios from 'axios';
import PetHospitalView from './Health/petHospital';
import Modal from 'antd/lib/modal';
import { List } from 'react-content-loader';
import lockIcon from '../img/btn_locked.png';
import shareIcon from '../img/btn_share.png';
import { rejects } from 'assert';
import strain from '../Api/犬类品系.json';

class Lists extends Component {
    constructor(props) {
        super(props);
        this.state={
            type: window.location.pathname.split('/')[1],
            dog_name: null,
            dog_image: null,
            summary: null,
            visible: false,
            shareVisible: false,
            petHospital: false,
            advice: false,
            render: 0
        }
    }

    componentDidMount () {
        // 获取报告的相应 json 数据
        let pathParams = this.props.match.params.type;
        axios({
            method: 'get',
            url: `../data/json/${pathParams}.json`,
            headers: {
                'Content-Type': 'application/json',
                'Cache-Control': 'max-age=86400'
            }
        })
        .then(response => {
            let type = response.data.header.物种;
            function getSigleGene () {
                return axios({
                    method: 'get',
                    url: `../../data/db/${type}类单基因疾病.json`,
                    headers: {
                        'Content-Type': 'application/json',
                        'Cache-Control': 'max-age=86400'
                    }
                })
            }
            function getComplexGene () {
                return axios({
                    method: 'get',
                    url: `../../data/db/${type}类复杂疾病.json`,
                    headers: {
                        'Content-Type': 'application/json',
                        'Cache-Control': 'max-age=86400'
                    }
                })
            }
            if (response.status === 200) {
                // console.log('相应数据',response.data)
                let data = {};
                let progressData = {};
                let lockList = [];
                _.map(response.data, (item, index) => {
                    // console.log('遍历过程',item)
                    let name = index;
                    let body = item.body;
                    if (item.header) {
                        if (item.visible) {
                            if (item.header.summary) {
                                _.map(item.header.summary, (item, index) => {
                                    if (index === '品种成分') {
                                        let obj = body['品系组成'];
                                        let str = '';
```

```
    let arr = [];
    _.map(obj, (item, index) => {
        str += `${index}、`;
        arr.push(index);
    })
    str = str.substring(0,str.length - 1);
    data[index] =
        <div className="summary-text">
            <Link to={{
                pathname: `/${pathParams}/detail`
                state: {
                    status: '详情',
                    name,
                    item,
                    body,
                    dog_name: response.data.heade
                    showAction: response.data.行
                    showBreed: response.data.品系
                    showHealth: response.data.健
                    strain: arr
                }
            }}
            ><div className="content"><span>{str}
        </div>
} else if (index === '品种高发疾病') {
    let obj = body['品系组成'];
    let arr = [];
    _.map(obj, (item, index) => {
        arr.push(index);
    })
    let str = '';
    let newStr = '';
    _.map(arr, (item, index) => {
        str += `${strain[item]['常见遗传病']}、`
    })
    let strUnique = (str) => {
        var ret = [];
        str.replace(/[^、]+/g, function($1, $2){
            (str.indexOf($1) == $2) && ret.push($
        });
```

```
                                            return ret;
                                        }
                                        let newArr = strUnique(str);
                                        if (newArr.length > 2) {
                                            newStr = `${newArr[0]}、${newArr[1]} 等 $
                                        } else {
                                            _.map(newArr, (item, index) => {
                                                newStr += `${item}、`;
                                            })
                                            newStr = newStr.substring(0,newStr.length
                                        }
                                        data[index] =
                                            <div className="summary-text">
                                                <Link to={{
                                                    pathname: `/${pathParams}/detail`
                                                    state: {
                                                        status: ' 详情',
                                                        name,
                                                        item,
                                                        body,
                                                        dog_name: response.data.heade
                                                        showAction: response.data. 行
                                                        showBreed: response.data. 品
                                                        showHealth: response.data. 健
                                                        disease: newArr
                                                    }
                                                }}
                                                ><div className="content"><span>{newS
                                            </div>
                                    } else {
                                        data[index] =
                                            <div className="summary-text">
                                                <Link to={{
                                                    pathname: `/${pathParams}/detail`
                                                    state: {
                                                        status: ' 详情',
                                                        name,
                                                        item,
                                                        body,
                                                        dog_name: response.data.heade
                                                        showAction: response.data.行
                                                        showBreed: response.data.品系
                                                        showHealth: response.data.健
                                                    }
                                                }}
                                                ><div className="content"><span>{item
                                            </div>
                                    }
                                })
                            }
                        } else {
                            if (item.header.summary) {
                                _.map(item.header.summary, (item, index) => {
                                    lockList.push(index)
                                    data[index] = (
                                        <div className="summary-text">
                                            <Link to={{
                                                pathname: `/${pathParams}/detail`,
                                                state: {
                                                    status: '未解锁功能',
                                                    name,
                                                    item,
                                                    body,
                                                    dog_name: response.data.header.宠
                                                    showAction: response.data.行为.v
                                                    showBreed: response.data.品系.vi
                                                    showHealth: response.data.健康.v
                                                }
                                            }}
                                            ><div className="content"><span><img src=
                                        </div>
                                    )
                                })
                            }
                        }
                    }
                    // 健康项处理
                    if (index === '健康') {
                        // console.log(item)
                        _.map(item.body, (item, index) => {
```

```
// console.log(item)
let {normal = [], carry = [], lock = []} = {};
if (item.body) {
    _.map(item.body, (item, index) => {
        if (item.visible === false) {
            item.name = index;
            lock.push(item);
        } else if (item.body[' 检测结果'] === ' 未携带致病
            item.name = index;
            normal.push(item);
        } else {
            item.name = index;
            carry.push(item);
        }
    })
}
// console.log(' 未解锁',unlock,' 正常',normal,' 携带',car
lockList[index] = lock;
data[index] = (
    <div className="summary-text">
        {
            lock.length
            ?
            <Link to={{
                pathname: `/${pathParams}/detail`,
                state: {
                    status: ' 未解锁功能',
                    name,
                    item: index,
                    dog_name: response.data.header. 宠物
                    showAction: response.data. 行为.visib
                    showBreed: response.data. 品系.visibl
                    showHealth: response.data. 健康.visib
                }
            }}
            >
                <span className="content">
                {/*
                  <span>{`${normal.length}项正常、`}</spa
                */}
```

```
                                                {carry.length>0 ? <span>{`${carry.len

                                                {
                                                    lock.length
                                                    ?
                                                    <span>{`、${lock.length} 项未解锁
                                                    :
                                                    ""
                                                }
                                            </span>
                                            <i className="fas fa-angle-right"></i>
                                        </Link>
                                        :
                                        <Link to={{
                                            pathname: `/${pathParams}/detail`,
                                            state: {
                                                status: '详情',
                                                name,
                                                item: index,
                                                dog_name: response.data.header.宠物
                                                showAction: response.data.行为.visib
                                                showBreed: response.data.品系.visibl
                                                showHealth: response.data.健康.visib
                                            }
                                        }}
                                        >
                                            <span className="content">
                                            {/*
                                              <span>{`${normal.length}项正常、`}</spa
                                            */}
                                                {carry.length>0 ? <span>{`${carry.len
                                                {
                                                    lock.length
                                                    ?
                                                    <span>{`、${lock.length} 项未解锁
                                                    :
                                                    ""
                                                }
                                            </span>
                                            <i className="fas fa-angle-right"></i>
                                        </Link>
                                    }
                                </div>
                            )
                        })
                    }

                    if (index === '进阶研究') {
                        _.map(item.body, (item, index) => {
                            if (item.visible) {
                                let progressName = index;
                                let progressContent = item.body;
                                let progressFooter = item.footer;
                                if (item.header.summary) {
                                    _.map(item.header.summary, (item, index) => {
                                        let content = item;
                                        if (index === '后代毛型预测' || index === '
                                            let reg0 = new RegExp('"',"g");
                                            let reg1 = new RegExp('{',"g");
                                            let reg2 = new RegExp('}',"g");
                                            let reg3 = new RegExp('，配',"g");
                                            content = JSON.stringify(progressContent)
                                            content = content.replace(reg0, "").repla
                                        }

                                        progressData[index] =
                                            <div className="summary-text">
                                                <Link to={{
                                                    pathname: `/${pathParams}/progres
                                                    state: {
                                                        name: progressName,
                                                        body: progressContent,
                                                        footer: progressFooter,
                                                        dog_name: response.data.heade
                                                        showAction: response.data.行
                                                        showBreed: response.data.品
                                                        showHealth: response.data.健
                                                    }
                                                }}
                                                ><div className="content"><span>{cont
                                            </div>
                                    })
                                }
                            }
                        })
                    }
                })
                axios.all([getSigleGene(), getComplexGene()])
                .then(axios.spread((resSingleGene, resComplexGene) => {
                    //当这两个请求都完成的时候会触发这个函数，两个参数分别代表返回的结果
                    // console.log('报告数据', response, '单基因数据', resSingleGene
                    if (resSingleGene.status === 200 || resComplexGene.status === 200
                        const healthData = () => {
                            let healthData = [];
                            _.map(response.data.健康.body.单基因疾病.body, (item, i
                                if (item.visible) {
                                    item.name = index;
                                    item.indexName = index.substring(0,index.indexOf(
                                    item.site = index.substring(index.indexOf("(") +
                                    item.advice = '';
                                    item.describe = '';
                                    item.symptoms = '';
                                    item.raise = '';
                                    if (resSingleGene.data[item.indexName]) {
                                        item.advice = resSingleGene.data[item.indexNa
                                        item.describe = resSingleGene.data[item.index
                                        item.symptoms = resSingleGene.data[item.index
                                        item.raise = resSingleGene.data[item.indexNam
                                        item.type = resSingleGene.data[item.indexName
                                        item.refer = resSingleGene.data[item.indexNam
                                    }
                                    healthData.push(item);
                                }
                            });
                            return healthData;
                        }
                        const complexHealthData = () => {
                            let complexHealthData = [];
                            _.map(response.data.健康.body.复杂疾病性状.body, (item,
                                if (item.visible) {
                                    item.name = index;
                                    item.advice = '';
                                    item.describe = '';
                                    item.symptoms = '';
                                    item.raise = '';
                                    if (resComplexGene.data[index]) {
                                        item.advice = resComplexGene.data[index].诊疗
                                        item.describe = resComplexGene.data[index].致
                                        item.symptoms = resComplexGene.data[index].临
                                        item.raise = resComplexGene.data[index].养育
                                        item.type = resComplexGene.data[index].所属类
                                        item.refer = resComplexGene.data[index].参考
                                    }
                                    complexHealthData.push(item);
                                }
                            });
                            return complexHealthData;
                        }
                        let raise = _.map(response.data.健康.footer,(item, index) =>
                            return (
                                <div key={index}>{item}</div>
                            )
                        })
                        // console.log('检查 res',response.data)
                        this.setState({
                            healthData: healthData(),
                            complexHealthData: complexHealthData(),
                            raise
                        })
                    }
                }))
                this.setState({
                    dog_name: response.data.header.宠物名,
                    dog_image: response.data.header.宠物照片,
                    pet_type: response.data.header.物种,
                    character: response.data.行为.header.summary.性格标签,
                    summary: data,
                    progress: progressData,
                    showAction: response.data.行为.visible,
                    showBreed: response.data.品系.visible,
                    showHealth: response.data.健康.visible,
                    searchData: response.data,
                    render: 1
                })
            }
        })
        .catch(error => {
            // console.log('错误',error)
            this.setState({
                dog_name: null,
                dog_image: null,
                pet_type: null,
                character: null,
                summary: null,
            })
        });
    }

    goToDirect = (item) => {
        console.log("click", item)
    }

    hideModal = () => {
        this.setState({
            visible: false,
            shareVisible: false,
            petHospital: false,
            advice: false
        });

    }

    shareBtn = () => {
        if (this.state.render) {
            this.setState({
                shareVisible: true
            });
        }
    }

    petHospital = ()=> {
        this.setState({
            petHospital:true
        })
    }

    render() {
        let { data = [], complexData = [] } = {};
        if (this.state.healthData) {
            data = this.state.healthData;
        }
        if (this.state.complexHealthData) {
            complexData = this.state.complexHealthData;
        }
        let HEALTHILLDATA = data.filter(item => item.body.检测结果 === '患病');
        let HEALTHCARRYDISEASE = data.filter(item => item.body.检测结果 === '携带致
        let HIGHCOMPLEXDATA = complexData.filter(item => item.body.检测结果 === '高
        let NORMALCOMPLEXDATA = complexData.filter(item => item.body.检测结果 === '
        let adviceData = [...HEALTHILLDATA, ...HEALTHCARRYDISEASE, ...HIGHCOMPLEXDATA
        console.log('summaryContent',this.state.summary,this.state.dog_name,this.stat
        let summaryContent = _.map(this.state.summary, (item, index) => {
            console.log(index,'index');
            return(
                <li className='stats-info' key={index}>
                    <div className='wolfiness'>
                        <span>{index}:</span>
                    </div>
                    <div className='mdium'>
                        <span>{item}</span>
                    </div>
                </li>
            )
        })

        let progressContent = _.map(this.state.progress, (item, index) => {
            return(
                <li className='stats-info' key={index}>
                    <div className='wolfiness'>
```

```
                        <span>{index}:</span>
                    </div>
                    <div className='mdium'>
                        <span>{item}</span>
                    </div>
                </li>
            )
        })

        let linkToData = [
            {
                'id': 1,
                'title': ' 血统',
                'description': `发现${this.state.dog_name}的血缘秘密和家族信息 `,
                'icon': 'title-icon breed-icon',
                'background': 'stats breed-bg',
                'is_direct': ' 血统',
                'linkTo': 'breed'
            }, {
                'id': 2,
                'title': ' 健康',
                'description': `了解${this.state.dog_name}的单基因和多基因遗传病携带情
                'icon': 'title-icon health-icon',
                'background': 'stats health-bg',
                'is_direct': ' 健康',
                'linkTo': 'health'
            },{
                'id': 3,
                'title': ' 行为',
                'description': ' 从攻击性、恐惧感和追逐、逃逸等 17 个维度了解爱宠的行
                'icon': 'title-icon action-icon',
                'background': 'stats action-bg',
                'is_direct': ' 行为',
                'linkTo': 'action'
            }
        ];

        if (!this.state.showAction) {
            linkToData = linkToData.filter(item => item.id !== 3)
        }
        if (!this.state.showHealth) {
            linkToData = linkToData.filter(item => item.id !== 2)
        }

        if (!this.state.showBreed) {
            linkToData = linkToData.filter(item => item.id !== 1)
        }

        let renderData = linkToData.map((item) => {
            return (
                <li key={item.id} className={item.background}>
                    <div className='title'>
                        <i className={item.icon}></i>
                        <h4>{item.title}</h4>
                    </div>
                    <div className="description">
                        {item.description}
                    </div>
                    <Link to={`/${this.state.type}/${item.linkTo}`} className="link-t
                </li>
            )
        });

        let showType = <div className='list-warpper health-warpper'>
            <div className=''>
                <div className='dog-information'>
                    <h3 className='dog-name-md'>{this.state.dog_name}</h3>
                    <div className='head-portrait'><img src={ImgDog} alt="" /></div>
                </div>
                <section className='stats'>
                    <div className='title'>
                        <i className="title-icon report-icon"></i>
                        <h4>检测报告</h4>
                    </div>
                    {
                        this.state.summary
                        ?
                        <ul className='text'>
                            {summaryContent}
                        </ul>
                        :
                        <div className='text'>
                            <List />
                        </div>
                    }
                </section>
                <section className='stats'>
                    <div className='title'>
                        <i className="title-icon progress-icon"></i>
                        <h4> 进阶研究</h4>
                    </div>
                    {
                        this.state.progress
                        ?
                        <ul className='text'>
                            {progressContent}
                        </ul>
                        :
                        <div className='text'>
                            <List />
                        </div>
                    }
                </section>
            </div>
            <ul className="render-list">
                {renderData}
            </ul>
            <div className="more">
                <h4>更多内容</h4>
                <div className='stats-block'>
                    <span> 点击</span>
                    <strong>
                        <Link to={{
                                pathname: `/${this.props.match.params.type}/advice`,
                                state: {
                                    status: 'genome',
                                    name: ' 检测报告',
                                    dog_name: this.state.dog_name,
                                    showAction: this.state.showAction,
                                    showBreed: this.state.showBreed,
                                    showHealth: this.state.showHealth,
                                    adviceData,
                                    searchData: this.state.searchData
                                }
                            }}
                        >
                            科学的养育建议
                        </Link>
                    </strong>
                    <span>了解并与您的宠物医师分享精准宠爱<sup>TM</sup> 的建议。</span
                </div>
                <div className='stats-block'>
                    <span> 点击</span>
                    <strong onClick={this.petHospital}>附近可提供协助的宠物医院</stron
                    <span> 查看精准宠爱<sup>TM</sup>推荐的专业机构。</span>
                </div>
                <div className='stats-block'>
                    <span>点击完成</span>
                    <strong><a href="https://jinshuju.net/f/UBURpn"> 调查问卷</a></st
                    <span>，查看</span>
                    <strong>
                        <Link to={{
                                pathname: `/${this.props.match.params.type}/disclaime
                                state: {
                                    status: ' 详情',
                                    name: ' 免责声明',
                                    dog_name: this.state.dog_name,
                                    showAction: this.state.showAction,
                                    showBreed: this.state.showBreed,
                                    showHealth: this.state.showHealth
                                }
                            }}
                        >
                            免责声明
                        </Link>
                    </strong>
                    <span>内容</span>
                </div>
            </div>
```

```
        </div>;
        return (
            <div className="list-content">
                <Header
                    tab="genome"
                    dog_name={this.state.dog_name}
                    showAction={this.state.showAction}
                    showBreed={this.state.showBreed}
                    showHealth={this.state.showHealth}
                />
                {showType}
                <ShareHtml
                    dog_name={this.state.dog_name}
                    pet_type={this.state.pet_type}
                    character={this.state.character}
                />
                <div className="share-btn" onClick={this.shareBtn}>
                    <img src={shareIcon}/>
                </div>
                {
                    this.state.render
                    ?
                    <Share
                        visible={this.state.shareVisible}
                        hideModal={this.hideModal}
                    />
                    :
                    ""
                }
                <Modal
                    title=" 宠物医院"
                    visible={this.state.petHospital}
                    onOk={this.handlePetHospital}
                    onCancel={this.hideModal}
                    footer={null}
                    width={700}
                >
                    <div className='pet-hospital'>
                        <PetHospitalView />
                    </div>
                </Modal>
            </div>
        );
    }
}

export default Lists;
```

progress.js 进阶研究页

```
import React, { Component } from 'react';
import './detail.css';
import Header from './Header/header';
import {Link} from 'react-router-dom';
import _ from 'lodash';
import dog_advancedstudy from '../img/icon_introduction_green.svg';
import axios from 'axios';

class Progress extends Component {
    /**
     * 这是报告详情页面
     *
    */
    state = {
        showDog: window.location.pathname.split('/')[4],
    }

    componentDidMount() {
        // console.log(this.props)
        let name = null;
        let body = null;
        name = this.props.location.state.name;
        body = this.props.location.state.body;
        let title = null;
        console.log(name)
        switch (name) {
            case '毛色预测':
                title = '位点基因型'
                break;
            case '毛型预测':
                title = '位点基因型'
                break;
            case '后代毛色预测':
                title = '毛色预测'
                break;
            case '后代毛型预测':
                title = '毛型预测'
                break;
            default:
                break;
        }

        axios({
            method: 'get',
            url: `../data/db/交互式报告素材.json`,
            headers: {
                'Content-Type': 'application/json',
                'Cache-Control': 'max-age=86400'
            }
        })
        .then(response => {
            // console.log(response.data[name])
            let content = null;
            let display = {display: 'block', whiteSpace: 'pre-wrap'};
            let result = _.map(response.data['进阶研究'][name], (item, index) => {
                return (
                    <li key={index} style={{position:'relative'}}>
                        <p className='dog-print' style={display}><img src={dog_advanc
                        <span style={display}>{index}</span>
                        <div
                            style={display}
                            dangerouslySetInnerHTML={{__html: item}}
                        >
                        </div>
                    </li>
                )
            })
            let bodyResult = _.map(body, (item, index) => {
                if (typeof(item) === 'object') {
                    let returnContent = _.map(item, (secondItem, secondIndex) => {
                        return (
                            <div className='list-text' key={secondIndex}>
                                <p style={{fontWeight: 'bold'}}>{secondIndex}</p>
                                <p>{secondItem}</p>
                            </div>
                        )
                    })
                    return (
                        <li key={index}>

                            <span style={display}>{index}</span>
                            <div style={display}>
                                {returnContent}
                            </div>
                        </li>
                    )
                } else {
                    return (
                        <div className='small-box' key={index}>
                            <p style={{marginTop: 0}}>{index}</p>
                            <span className='conditions' style={{fontWeight: 'normal'
                        </div>
                    )
                }
            })

            content =
                    <ul className="list">
                        <div className="list-first" style={{position:'relative'}}>
                            <p className='dog-print' style={display}><img src={dog_ad
                            <span style={display}>{title}</span>
                            <div className='health-condition-wapper' style={{display:
                                {bodyResult}
                            </div>
                        </div>
                        {result}
                    </ul>
            this.setState({
                content
            })
            // console.log(content)
        })
        .catch(error => {
            // console.log('错误',error)
        });
    }

    render() {
        return (
            <div className="detail">
                <Header
                    tab="genome"
                    dog_name={this.props.location.state.dog_name}
                    showAction={this.props.location.state.showAction}
                    showBreed={this.props.location.state.showBreed}
                    showHealth={this.props.location.state.showHealth}
                />
                <div className='condition-wapper health-warpper'>
                    <div className='condition-header'>
                        <Link to={`/${this.props.match.params.type}`}> 检测报告</Link
                        <i>></i>
                        <span>{this.props.location.state.name}
                        </span>
                    </div>
                    <div className="detail-content">
                        {this.state.content}
                    </div>
                </div>
            </div>
        );
    }
}

export default Progress;
```

discliamier.js 免责声明页

```
import React, { Component } from 'react';
import './detail.css';
import Header from './Header/header';
import {Link} from 'react-router-dom';
import _ from 'lodash';
import axios from 'axios';

class Disclaimer extends Component {
    /**
     * 这是报告详情页面
     *
    */
    state = {
        showDog: window.location.pathname.split('/')[4],
    }

    componentDidMount() {
        let content = null;
        axios({
            method: 'get',
            url: `../data/db/交互式报告素材.json`,
            headers: {
                'Content-Type': 'application/json',
                'Cache-Control': 'max-age=86400'
            }
        })
        .then(response => {
            let display = {display: 'block', whiteSpace: 'pre-wrap'};
            let result = _.map(response.data['免责声明'], (item, index) => {
                    return (
                        <li key={index}>
                            <span style={display}>{index}</span>
                            <div
                                style={display}
                                dangerouslySetInnerHTML={{__html: item}}
                            >
                            </div>
                        </li>
                    )
            })

            content = <ul className="list">{result}</ul>
            this.setState({
                content
            })
            // console.log(content)
        })
        .catch(error => {
            // console.log('错误',error)
        });
    }

    render() {
        return (
            <div className="detail">
                <Header
                    tab="genome"
                    dog_name={this.props.location.state.dog_name}
                    showAction={this.props.location.state.showAction}
                    showBreed={this.props.location.state.showBreed}
                    showHealth={this.props.location.state.showHealth}
                />
                <div className='condition-wapper health-warpper'>
                    <div className='condition-header'>
                        <Link to={`/${this.props.match.params.type}`}>检测报告</Link
                        <i>></i>
                        <span>免责声明</span>
                    </div>
                    <div className="detail-content">
                        {this.state.content}
                    </div>
                </div>
            </div>
        );
    }
}

export default Disclaimer;
```

action.js 行为页

```
import React, { Component } from 'react';
import Header from '../Header/header';
import './action.css';
import { Radar } from 'react-chartjs-2';
import Icon from 'antd/lib/icon';
import Modal from 'antd/lib/modal';
import _ from 'lodash';
import axios from 'axios';
import ContentLoader from 'react-content-loader';
import Share from '../Share/share';
import ShareHtml from '../Share/shareHtml';
import shareIcon from '../../img/btn_share.png';

const TextLoader = () => (
    <ContentLoader height="60">
        {/* Pure SVG */}
        <rect x="150" y="20" rx="4" ry="4" width="100" height="15" />
    </ContentLoader>
)

const ChartLoader = () => (
    <ContentLoader height="260">
        {/* Pure SVG */}
        <rect x="80" y="0" rx="4" ry="4" width="240" height="200" />
    </ContentLoader>
)

const chartData = (data, tpye) => {
    // console.log(' 传过来的 data',data);
    let result = '';
    let labels = [];
    let meansData = {};
    let unityData = {};
    let means = [];
    let unity = [];
    let resultData = {
        labels: labels,
        datasets: [
            {
                label: 'null',
                backgroundColor: 'rgba(179,181,198,0.2)',
                borderColor: 'rgba(179,181,198,1)',
                pointBackgroundColor: 'rgba(179,181,198,1)',
                pointBorderColor: '#fff',
                pointHoverBackgroundColor: '#fff',
                pointHoverBorderColor: 'rgba(179,181,198,1)',
                data: meansData
            },
            {
                label: 'null',
                backgroundColor: 'rgba(255,99,132,0.2)',
                borderColor: 'rgba(255,99,132,1)',
                pointBackgroundColor: 'rgba(255,99,132,1)',
                pointBorderColor: '#fff',
                pointHoverBackgroundColor: '#fff',
                pointHoverBorderColor: 'rgba(255,99,132,1)',
                data: unityData
            }
        ]
    };
    switch (tpye) {
        case 'title':
            result = 'null';
            break;
        case 'describe':
            result = 'null';
            break;
        case 'data':
            result = resultData;
            break;
        default:
            break;
    }
    if (data) {
        _.map(data.body, (item, index) => {
            // console.log('内容',item,'索引',index)
            // meansData[index] = item;
            if (index === '犬类均值') {
                meansData.label = index;
                meansData.data = item;
            } else {
                unityData.label = index;
                unityData.data = item;
            }
            // console.log('处理结果',meansData,unityData)
        })
        _.map(meansData.data, (item, index) => {
            labels.push(index);
            means.push(item);
            unity.push(unityData.data[index]);
        })
        resultData = {
            labels: labels,
            datasets: [
                {
                    label: meansData.label,
                    backgroundColor: 'rgba(52, 174, 166, 0.2)',
                    borderColor: '#34aea6',
                    pointBackgroundColor: '#34aea6',
                    pointBorderColor: '#fff',
                    pointHoverBackgroundColor: '#fff',
                    pointHoverBorderColor: '#34aea6',
                    borderWidth: 1,
                    data: means
                },
                {
                    label: unityData.label,
                    backgroundColor: 'rgba(210, 86, 80, 0.2)',
                    borderColor: '#d25650',
                    pointBackgroundColor: '#d25650',
                    pointBorderColor: '#fff',
                    pointHoverBackgroundColor: '#fff',
                    pointHoverBorderColor: '#d25650',
                    borderWidth: 1,
                    data: unity
                }
            ]
        }
        switch (tpye) {
            case 'title':
                result = data.header[0];
                break;
            case 'describe':
                result = data.header[1];
                break;
            case 'data':
                result = resultData
                break;
            default:
                break;
        }
    }
    return result;
}

const option = {
    tooltips: {
        mode: 'index'
    },
    scale: {
        ticks: {
            display: false,
            beginAtZero: true,
            max: 4,
            min: 0,
            stepSize: 1,
        },
        gridLines: {
            display: false
        },
        pointLabels: {
            fontSize: 12
        }
    },
    layout: {
        padding: {
            left: 80,
            right: 0,
            top: 10,
            bottom: 0
        },
    },
    legend: {
        display: true,
        position: 'bottom',
        fullWidth: true,
        labels: {
            boxWidth: 0,
            fontSize: 12,
            padding: 10,
            fontColor: '#595959',
            usePointStyle: true
        }
    },
};

class ActionView extends Component {
    constructor(props) {
        super(props);
        this.state = {
            type: window.location.pathname.split('/')[1],
            sumDescription: '',
            visible: false,
            dog_name: null,
            character: null,
            fightData: null,
            fearData: null,
            totalData: null,
            describe: null,
            title: '',
            display: {
                display: 'none'
            },
            shareVisible: false,
            render: 0
        };
    }
    componentDidMount () {
        // 获取报告的相应 json 数据
        let pathParams = this.props.match.params.type;
        axios({
            method: 'get',
            url: `../../data/json/${pathParams}.json`,
            headers: {
                'Content-Type': 'application/json',
                'Cache-Control': 'max-age=86400'
            }
        })
        .then(response => {
            if (response.status === 200) {
                // console.log('检查数据',response.data)
                this.setState({
                    dog_name: response.data.header.宠物名,
                    pet_type: response.data.header.物种,
                    character: response.data.行为.header.summary.性格标签,
                    fightData: response.data.行为.body.好斗性,
                    fearData: response.data.行为.body.恐惧感,
                    totalData: response.data.行为.body.综合行为,
                    showAction: response.data.行为.visible,
                    showBreed: response.data.品系.visible,
                    showHealth: response.data.健康.visible,
                    render: 1
                })
            }
        })
        .catch(error => {
            this.setState({
                dog_name: null,
                character: null,
                fightData: null,
                fearData: null,
                totalData: null,
                render: 1
            })
        });
    }
    handleOk = (e) => {
        // console.log(e);
        this.setState({
            visible: false,
        });
    }

    handleCancel = (e) => {
        // console.log(e);
        this.setState({
            visible: false,
        });
    }

    describe = (title) => {
        if (title !== this.state.title) {
            axios({
                method: 'get',
                url: `../../data/db/交互式报告素材.json`,
                headers: {
                    'Content-Type': 'application/json'
                }
            })
            .then(response => {
                if (response.status === 200) {
                    // console.log('检查数据',response.data['行为'][title]['简介'
                    let describe = response.data['行为'][title]['简介'];
                    if (title === '综合行为') {
                        let content = _.map(describe, (item, index) => {
                            return (
                                <li key={index}>
                                    <span>{index}</span>
                                    {item}
                                </li>
                            )
                        })
                        this.setState({
                            title,
                            content
                        });
                    } else {
                        this.setState({
                            title,
                            content: <p style={{'whiteSpace': 'pre-line'}}>{describe}
                        })
                    }
                }
            })
            .catch(error => {
                this.setState({
                    content: '',
                    title: ''
                })
            });
        }

        // if (title === '综合行为') {
            this.setState({
                visible: true
            });
        // }

        return this.state.describe;
    }

    hideModal = () => {
        this.setState({
            visible: false,
            shareVisible: false,
        });

    }

    shareBtn = () => {
        if (this.state.render) {
            this.setState({
                shareVisible: true
            });
        }
    }
    render() {
        // console.log(' 检查 actions',this.state);
        return (
            <div className="health">
                <Header
                    tab="action"
                    dog_name={this.state.dog_name}
                    showAction={this.state.showAction}
                    showBreed={this.state.showBreed}
                    showHealth={this.state.showHealth}
                />
                <div className='health-warpper action'>
                    {
                        this.state.character
                        ?
                        <div className="sum">{`${this.state.dog_name}是${this.state.c
                        :
                        <TextLoader />
                    }
                    {
                        this.state.fightData
                        ?
                        <div className="chart-item">
                            <div className="title">
                                <div className="title-content">
                                    <i className="icon fight"></i>
                                    <h4> 好斗性</h4>
                                </div>
                                <p className="desc"> 好斗性包括狗狗对主人的好斗性、陌

                            </div>
                            <Radar
                                data={chartData(this.state.fightData, 'data')}
                                options={option}
                            />
                        </div>
                        :
                        <ChartLoader />
                    }
                    {
                        this.state.fearData
                        ?
                        <div className="chart-item">
                            <div className="title">
                                <div className="title-content">
                                    <i className="icon fear"></i>
                                    <h4> 恐惧感</h4>
                                </div>
<p className="desc"> 恐惧感包括狗狗对陌生人的恐惧感、对陌生犬的恐惧感、对陌生环境的恐惧

                            </div>
                            <Radar
                                data={chartData(this.state.fearData, 'data')}
                                options={option}
                            />
                        </div>
                        :
                        <ChartLoader />
                    }
                    {
                        this.state.totalData
                        ?
                        <div className="chart-item">
                            <div className="title">
                                <div className="title-content">
                                    <i className="icon comprehensive"></i>
                                    <h4> 综合行为</h4>
                                    <Icon type="question-circle-o" className="icon" o
                                </div>
                            </div>
                            <Radar
                                data={chartData(this.state.totalData, 'data')}
                                options={option}
                            />

                        </div>
                        :
                        <ChartLoader />
                    }
```

```
                </div>
                <ShareHtml
                    dog_name={this.state.dog_name}
                    pet_type={this.state.pet_type}
                    character={this.state.character}
                />
                <div className="share-btn" onClick={this.shareBtn}>
                    <img src={shareIcon}/>
                </div>
                {
                    this.state.render
                    ?
                    <Share
                        visible={this.state.shareVisible}
                        hideModal={this.hideModal}
                    />
                    :
                    ""
                }
                <Modal
                    title={`${this.state.title} 说明 `}
                    visible={this.state.visible}
                    onOk={this.handleOk}
                    onCancel={this.handleCancel}
                    footer={null}
                    width={600}
                >
                    <div className="desc">
                        <ul>
                           {this.state.content}
                        </ul>
                    </div>
                </Modal>
            </div>
        );
    }
}

export default ActionView;
```

breed-detailed.js 血统详情页

```
import React, { Component } from 'react';
// import './detail.css';
import Header from '../../Header/header';
import {Link} from 'react-router-dom';
import _ from 'lodash';
import axios from 'axios';

class BreedDetailedView extends Component {
    /**
     * 这是报告详情页面
     *
    */
    state = {
        showDog: window.location.pathname.split('/')[4],
    }

    componentDidMount() {
        let name = this.props.location.state.dog_breed;
        axios({
            method: 'get',
            url: `../data/db/${this.props.location.state.pet_type}类品系.json`,
            headers: {
                'Content-Type': 'application/json',
                'Cache-Control': 'max-age=86400'
            }
        })
        .then(response => {
            // console.log(name,response.data[name])
            let content = null;
            let result = _.map(response.data[name], (item, index) => {
                // console.log('检查',name,index,text)
                let display = {display: 'block'};
                return (
                    <li key={index}>
                        <span style={display}>{index}</span>
                        <div
                            style={display}
                        >
                        {item}
                        </div>
                    </li>
                )
            })

            if (response.data[name]) {
                content = <ul className="list">{result}</ul>
            } else {
                content = <ul className="list"><li><span>没有该品种的相关信息</span></
            }

            this.setState({
                content
            })
            // console.log(content)
        })
        .catch(error => {
            // console.log(' 错误',error)
        });
    }

    render() {
        return (
            <div className="breed-detail">
                <Header
                    tab="breed"
                    dog_name={this.props.location.state.dog_name}
                    showAction={this.props.location.state.showAction}
                    showBreed={this.props.location.state.showBreed}
                    showHealth={this.props.location.state.showHealth}
                />
                <div className='condition-wapper health-warpper'>
                    <div className='condition-header'>
                        <Link to={`/${this.props.match.params.type}/breed`}>品种</Lin
                        <i>></i>
                        <span>{this.props.location.state.dog_breed}</span>
                    </div>
                    <div className="detail-content">
```

```
                        {this.state.content}
                    </div>
                    <ul className="render-list">
                    </ul>
                </div>
            </div>
        );
    }
}

export default BreedDetailedView;
```

breedTab.js 血统占比

```
import React, { Component } from 'react';
import {Link} from 'react-router-dom';
import ImgDogLooker from '../../../img/infoDog.jpg'
import _ from 'lodash';
import { List } from 'react-content-loader';

class BreedTabView extends Component {
    constructor(props) {
        super(props);
        this.state = {
            current: 0,
            type: window.location.pathname.split('/')[2],
            showMenu: false,
            color: 0
        };
    }

    render(){
        // console.log(' 检查 data',JSONDATA);
        let { color } = this.state;

        let mixContent = _.map(this.props.mix, (item, index) => {
            // console.log(index,item)
            let textColor = '';
            let barColor = '';
            color ++;
            switch (color) {
                case 1:
                    textColor = 'medium-text text-1';
                    barColor = 'breed-color-bg yellow';
                    break;
                case 2:
                    textColor = 'medium-text text-2';
                    barColor = 'breed-color-bg blue';
                    break;
                case 3:
                    textColor = 'medium-text text-3';
                    barColor = 'breed-color-bg red';
                    break;
                default:
                    break;
            }
            return (
                <div key={index}>
                    <Link to={{
                            pathname: `/${this.props.pathParams}/breed_detailed`,
                            state: {
                                dog_name: this.props.dog_name,
                                showAction: this.props.showAction,
                                showBreed: this.props.showBreed,
                                showHealth: this.props.showHealth,
                                dog_breed: index,
                                pet_type: this.props.pet_type
                            }
                        }}
                        className={textColor}
                    >
                        {`${(item*100).toFixed(1)}%${index}`}
                    </Link>
                    <div className={barColor} style={{width: `${item*100}%`}}></div>
                </div>
            )
        });

        return(
            <div className='breed-tab'>
                <div className='dog-information'>
                    <h3 className='dog-name-md'>{this.props.dog_name}</h3>
                    <div className='head-portrait'><img src={ImgDogLooker} alt="" /><
                </div>
                <section className='stats'>
                    <div className='stats-left breed-img'>
                        <h4> 品种混合</h4>
                    </div>
                    {
                        this.props.mix
                        ?
                        <div className='stats-right'>
                            <div className='medium'>
                                {mixContent}
                            </div>
                        </div>
                        :
                        <div className='stats-right'>
                            <List />
                        </div>
                    }
                </section>
            </div>
        )
    }
}
export default BreedTabView;
```

breed.js 血统页

```
import React, { Component } from 'react';
import './breed.css';
import BreedTabView from './Breed-tab/breedTab';
import Header from '../Header/header';
import {Link} from 'react-router-dom';
import axios from 'axios';
import Share from '../Share/share';
import ShareHtml from '../Share/shareHtml';
import shareIcon from '../../img/btn_share.png';

class BreedView extends Component {
    constructor(props) {
        super(props);
        this.state = {
            type: window.location.pathname.split('/')[1],
            current: 0,
            dog_name: null,
            dog_image: null,
            mix: null,
            shareVisible: false,
            render: 0
        };
    }

    componentDidMount () {
        // 获取报告的相应json数据
        let pathParams = this.props.match.params.type;
        axios({
            method: 'get',
            url: `../../../data/json/${pathParams}.json`,
            headers: {
                'Content-Type': 'application/json',
                'Cache-Control': 'max-age=86400'
            }
        })
        .then(response => {
            if (response.status === 200) {
                this.setState({
                    dog_name: response.data.header.宠物名,
                    dog_image: response.data.header.宠物照片,
                    pet_type: response.data.header.物种,
                    character: response.data.行为.header.summary.性格标签,
                    mix: response.data.品系.body.品系组成,
                    pedigree: response.data.品系.body.家族树,
                    pathParams,
                    showAction: response.data.行为.visible,
                    showBreed: response.data.品系.visible,
                    showHealth: response.data.健康.visible,
                    render: 1
                })
            }
        })
        .catch(error => {
            this.setState({
                dog_name: null,
                dog_image: null,
                mix: null,
                pathParams
            })
        });
    }

    hideModal = () => {
        this.setState({
            visible: false,
            shareVisible: false,
        });

    }

    shareBtn = () => {
        if (this.state.render) {
            this.setState({
                shareVisible: true
            });
        }
    }

    render(){
        return(
            <div className='breed'>
                <Header
                    tab="breed"
                    dog_name={this.state.dog_name}
                    showAction={this.state.showAction}
                    showBreed={this.state.showBreed}
                    showHealth={this.state.showHealth}
                />
                <div className='breed-warpper health-warpper'>
                    <div className='breed-tab'>
                        <BreedTabView
                            dog_name={this.state.dog_name}
                            dog_image={this.state.dog_image}
                            mix={this.state.mix}
                            pathParams={this.state.pathParams}
                            showAction={this.state.showAction}
                            showBreed={this.state.showBreed}
                            showHealth={this.state.showHealth}
                            pet_type={this.state.pet_type}
                        />
                        <div className="genealogical-tree">
                            <div className="left"><i className="icon"></i> 家族树</di
                            <div className="right">
                                <Link to={{
                                    pathname: `/${this.state.pathParams}/genealogical
                                    state: {
                                        dog_name: this.state.dog_name,
                                        showAction: this.state.showAction,
                                        showBreed: this.state.showBreed,
                                        showHealth: this.state.showHealth,
                                        pedigree: this.state.pedigree
                                    }
                                }}>查看详情</Link>
                                <i className="fas fa-angle-right"></i>
                            </div>
                        </div>
                    </div>
                </div>
                <ShareHtml
                    dog_name={this.state.dog_name}
                    pet_type={this.state.pet_type}
                    character={this.state.character}
                />
                <div className="share-btn" onClick={this.shareBtn}>
                    <img src={shareIcon}/>
                </div>
                {
                    this.state.render
                    ?
                    <Share
                        visible={this.state.shareVisible}
                        hideModal={this.hideModal}
                    />
                    :
                    ""
                }
            </div>
        )
    }
}
export default BreedView;
```

genealogical-tree.js 家族树页

```
import React, { Component } from 'react';
import Header from '../Header/header';
import {Link} from 'react-router-dom';

class GenealogicalTreeView extends Component {
    /**
     * 这是报告详情页面
     *
    */
    state = {
        showDog: window.location.pathname.split('/')[4],
    }

    render() {
        return (
            <div className="breed-detail">
                <Header
                    tab="breed"
                    dog_name={this.props.location.state.dog_name}
                    showAction={this.props.location.state.showAction}
                    showBreed={this.props.location.state.showBreed}
                    showHealth={this.props.location.state.showHealth}
                />
                <div className='condition-wapper health-warpper'>
                    <div className='condition-header'>
                        <Link to={`/${this.props.match.params.type}/breed`}> 品种</Li
                        <i>></i>
                        <span> 家族树</span>
                    </div>
                    <div className="detail-content">
                        <img src={`/${this.props.location.state.pedigree}`} width="10
                    </div>
                    <ul className="render-list">
                    </ul>
                </div>
            </div>
        );
    }
}

export default GenealogicalTreeView;
```

header.js 导航栏

```
import React, { Component } from 'react';
import {Link} from 'react-router-dom';
import './header.css';
import Logo from '../../img/logo.jpeg';

class Header extends Component {
    constructor(props) {
        super(props);
        this.state = {
            type: window.location.pathname.split('/')[1],
            showMenu: false,
            dog_name: null,
            action: null,
            health: null,
            breed: null
        };
    }
    switchMenu = () => {
        let showMenu = this.state.showMenu;
        this.setState({showMenu: !showMenu});
    }
    handleTab = () => {
        localStorage.setItem('key',1);
    }
    render(){
        const tab = this.props.tab;
        let dog_name = '';
        let action = '';
        let health = '';
        let breed = '';
        if (this.props.dog_name) {
            dog_name = this.props.dog_name;
            action = '行为';
            health = '健康';
            breed = '血统';
        }
        return(
            <div id='nav'>
                <header>
                    <nav className="navbar navbar-expand-lg navbar-light bg-light">
                        <a className="navbar-brand" href="javascript:;"><img classNam
                        <button className="navbar-toggler" type="button" data-toggle=
                            <span className="navbar-toggler-icon"></span>
                        </button>
                        <div className="collapse navbar-collapse" id="navbarNav">
                            <ul className="navbar-nav">
                                <li className={tab === 'genome' ? 'nav-item active ac
                                    <Link to={`/${this.state.type}`} className="nav-l
                                </li>
                                {
                                    this.props.showBreed
                                    ?
                                    <li className={tab === 'breed' ? 'nav-item active
                                        <Link to={`/${this.state.type}/breed`} classN
                                    </li>
                                    :
                                    ''
                                }
                                {
                                    this.props.showHealth
                                    ?
                                    <li className={tab === 'health' ? 'nav-item activ
                                        <Link to={`/${this.state.type}/health`} class
                                    </li>
                                    :
                                    ''
                                }
                                {
                                    this.props.showAction
                                    ?
                                    <li className={tab === 'action' ? 'nav-item activ
                                        <Link to={`/${this.state.type}/action`} class
                                    </li>
                                    :
                                    ''
                                }
                            </ul>
                        </div>
                    </nav>
                </header>
            </div>
        )
    }
}
export default Header;
```

health-condition.js 健康详情页

```
import React, { Component } from 'react';
import './health-condition.css';
import Header from '../Header/header';
import {Link} from 'react-router-dom';

class HealthCondition extends Component {
    constructor(props) {
        super(props);
        this.state = {

        };
    }

    render() {
        const data = this.props.location.state.data;
        const dog_name = this.props.location.state.dog_name;
        const showAction = this.props.location.state.showAction;
        const showBreed = this.props.location.state.showBreed;
        const showHealth = this.props.location.state.showHealth;
        // console.log('疾病详情',dog_name);
        return (
            <div className="condition">
                <Header
                    tab="health"
                    dog_name={dog_name}
                    showAction={showAction}
                    showBreed={showBreed}
                    showHealth={showHealth}
                />
                <div className='condition-wapper health-warpper'>
                    <div className='condition-header'>
                        <Link to={{pathname: `/${this.props.match.params.type}/health
                        <i>></i>
                        <span>
                            {data.name}
                        </span>
                    </div>
                    <section className='condition-describe'>
                        <h6><span style={{color: '#8ec856'}}>{dog_name}</span>的遗传病
                        <h3 style={{color: '#8ec856'}}>{data.body.检测结果}</h3>
                        <div className='columns'>
                            <p>致病机理</p>
                            <div className='text-health'>
                                {data.describe}
                            </div>
                        </div>
                        <div className='health-condition-wapper'>
                            {
                                data.body. 基因名
                                ?
                                <div className='small-box'>
                                    <p> 基因名</p>
                                    <span className='conditions'>{data.body. 基因名}<
                                </div>
                                :
                                ""
                            }
                            {
                                data.body. 染色体
                                ?
                                <div className='small-box'>
                                    <p>染色体</p>
                                    <span className='conditions'>{data.body. 染色体}<
                                </div>
                                :
                                ""
                            }
                            {
                                data.body. 宠物基因型
                                ?
                                <div className='small-box'>
                                    <p> 基因型</p>
                                    <span className='conditions'>{data.body. 宠物基因
                                </div>
                                :
                                ""
                            }
                            {
                                data.body. 正常位点基因型
                                ?
                                <div className='small-box'>
                                    <p>正常位点</p>
                                    <span className='conditions'>{data.body. 正常位点
                                </div>
                                :
                                ""
                            }
                            {
                                data.body.致病位点基因型
                                ?
                                <div className='small-box'>
                                    <p> 致病位点</p>
                                    <span className='conditions'>{data.body.致病位点
                                </div>
                                :
                                ""
                            }
                            {
                                data.body.遗传模式
                                ?
                                <div className='small-box'>
                                    <p>遗传模式</p>
                                    <span className='conditions'>{data.body.遗传模式
                                </div>
                                :
                                ""
                            }
                            {
                                data.body.致病突变遗传可能性
                                ?
                                <div className='small-box'>
                                    <p> 遗传可能性</p>
                                    <span className='conditions'>{data.body.致病突变
                                </div>
                                :
                                ""
                            }
                        </div>
                        {/*<div className='columns'>
                            <p>codominant</p>
                            <div className='health-condition-label'>Inheritance type<
                        </div>*/}
                        {
                            data.type
                            ?
                            <div className='columns'>
                                <p>所属类别</p>
                                <div className='text-health text-detailed'>
                                    {data.type}
                                </div>
                            </div>
                            :
                            ""
                        }
                        {
                            data.body.风险解读
                            ?
                            <div className='columns'>
                                <p>风险解读</p>
                                <div className='text-health text-detailed'>
                                    {data.body.风险解读}
                                </div>
                            </div>
                            :
                            ""
                        }
                        {
                            data.symptoms
                            ?
                            <div className='columns'>
                                <p>临床症状</p>
                                <div className='text-health text-detailed'>
                                    {data.symptoms}
                                </div>
                            </div>
                            :
                            ""
                        }
                        {
                            data.refer
                            ?
                            <div>
                                {
                                    typeof(data.refer) !== 'string'
                                    ?
                                    <div>
                                        {
                                            data.refer[data.site]
                                            ?
                                            <div className='columns'>
                                                <p>参考文献</p>
                                                <div className='text-health text-deta
                                                    <h5>{data.site}:</h5>
                                                    {data.refer[data.site]}
                                                </div>
                                            </div>
                                            :
                                            ''
                                        }
                                    </div>
                                    :
                                    <div className='columns'>
                                        <p> 参考文献</p>
                                        <div className='text-health text-detailed'>
                                            {data.refer}
                                        </div>
                                    </div>
                                }
                            </div>
                            :
                            ''
                        }
                        {
                            data.advice
                            ?
                            <div className='columns'>
                                <p> 诊疗建议</p>
                                <div className='text-health text-detailed'>
                                    {data.advice}
                                </div>
                            </div>
                            :
                            ""
                        }
                        {
                            data.raise
                            ?
                            <div className='columns'>
                                <p>养育建议</p>
                                <div className='text-health text-detailed'>
                                    {data.raise}
                                </div>
                            </div>
                            :
                            ""
                        }
                    </section>
                </div>
            </div>
        );
    }
}

export default HealthCondition;
```

health.js 健康页

```
import React, { Component } from 'react';
import './health.css';
import Header from '../Header/header';
import {Link} from 'react-router-dom';
import HealthAtrisk from '../../img/atriskLarge.png';
import HealthCarrierge from '../../img/carrierLarge.png';
import Modal from 'antd/lib/modal';
import Tabs from 'antd/lib/tabs';
import { HEALTHSUMMARY } from '../../Api/data';
import PetHospitalView from './petHospital';
import _ from 'lodash';
import axios from 'axios';
import { List } from 'react-content-loader';
import Share from '../Share/share';
import ShareHtml from '../Share/shareHtml';
import shareIcon from '../../img/btn_share.png';

const TabPane = Tabs.TabPane;
```

```
class Health extends Component {
    constructor(props) {
        super(props);
        this.state = {
            type: window.location.pathname.split('/')[1],
            advice:false,
            petHospitalView:false,
            dog_name: null,
            healthData: null,
            describe: null,
            shareVisible: false,
            key: localStorage.getItem('key') ? localStorage.getItem('key') :'1',
            render: 0
        };
    }

    componentDidMount () {
        // 获取报告的相应 json 数据
        let pathParams = this.props.match.params.type;
        axios({
            method: 'get',
            url: `../data/json/${pathParams}.json`,
            headers: {
                'Content-Type': 'application/json',
                'Cache-Control': 'max-age=86400'
            }
        })
        .then(response => {
            let type = response.data.header.物种;
            function getSigleGene () {
                return axios({
                    method: 'get',
                    url: `../../data/db/${type}类单基因疾病.json`,
                    headers: {
                        'Content-Type': 'application/json',
                        'Cache-Control': 'max-age=86400'
                    }
                })
            }
            function getComplexGene () {
                return axios({
                    method: 'get',
                    url: `../../data/db/${type}类复杂疾病.json`,
                    headers: {
                        'Content-Type': 'application/json',
                        'Cache-Control': 'max-age=86400'
                    }
                })
            }
            if (response.status === 200) {
                axios.all([getSigleGene(), getComplexGene()])
                .then(axios.spread((resSingleGene, resComplexGene) => {
                    //当这两个请求都完成的时候会触发这个函数，两个参数分别代表返回的结
                    // console.log(' 报告数据', response, ' 单基因数据', resSingleGe
                    if (resSingleGene.status === 200 || resComplexGene.status === 200
                        const healthData = () => {
                            let healthData = [];
                            _.map(response.data. 健康.body. 单基因疾病.body, (item, i
                                if (item.visible) {
                                    item.name = index;
                                    item.indexName = index.substring(0,index.indexOf(
                                    item.site = index.substring(index.indexOf("(") +
                                    item.advice = '';
                                    item.describe = '';
                                    item.symptoms = '';
                                    item.raise = '';
                                    if (resSingleGene.data[item.indexName]) {
                                        item.advice = resSingleGene.data[item.indexNa
                                        item.describe = resSingleGene.data[item.index
                                        item.symptoms = resSingleGene.data[item.index
                                        item.raise = resSingleGene.data[item.indexNam
                                        item.type = resSingleGene.data[item.indexName
                                        item.refer = resSingleGene.data[item.indexNam
                                    }
                                    healthData.push(item);
                                }
                            });
                            return healthData;
                        }
                        const complexHealthData = () => {
```

```
              let complexHealthData = [];
              _.map(response.data.健康.body.复杂疾病性状.body, (item,
                if (item.visible) {
                  item.name = index;
                  item.advice = '';
                  item.describe = '';
                  item.symptoms = '';
                  item.raise = '';
                  if (resComplexGene.data[index]) {
                    item.advice = resComplexGene.data[index].诊疗
                    item.describe = resComplexGene.data[index].致
                    item.symptoms = resComplexGene.data[index].临
                    item.raise = resComplexGene.data[index].养育
                    item.type = resComplexGene.data[index].所属类
                    item.refer = resComplexGene.data[index].参考
                  }
                  complexHealthData.push(item);
                }
              });
              return complexHealthData;
            }
            let raise = _.map(response.data.健康.footer,(item, index) =>
              return (
                <div key={index}>{item}</div>
              )
            })
            // console.log('检查 res',response.data)
            this.setState({
              dog_name: response.data.header.宠物名,
              pet_type: response.data.header.物种,
              character: response.data.行为.header.summary.性格标签,
              healthData: healthData(),
              complexHealthData: complexHealthData(),
              raise,
              showAction: response.data.行为.visible,
              showBreed: response.data.品系.visible,
              showHealth: response.data.健康.visible,
              render: 1
            })
          }
```

```
                })
            }
        })
    }

    tabChange = (key) => {
        localStorage.setItem('key', key);
        this.setState({
            key
        })
    }

    advice = () => {
        this.setState({
            advice: true
        })
    }

    handleAdvice = (e) => {
    //   console.log(e);
      this.setState({
        advice: false,
      });
    }

    handleCancelAdvice = (e) => {
    //   console.log(e);
      this.setState({
        advice: false,
      });
    }

    petHospital = ()=> {
        this.setState({
            petHospital:true
        })
    }

    handlePetHospital = (e) => {
    //   console.log(e);
```

```
      this.setState({
        petHospital: false,
      });
    }

    handleCancelHospital = (e) => {
    //   console.log(e);
      this.setState({
        petHospital: false,
      });
    }

    hideModal = () => {
        this.setState({
            visible: false,
            shareVisible: false,
        });
    }

    shareBtn = () => {
        if (this.state.render) {
            this.setState({
                shareVisible: true
            });
        }
    }

    render() {
        let { data = [], complexData = [] } = {};
        if (this.state.healthData) {
            data = this.state.healthData;
        }
        if (this.state.complexHealthData) {
            complexData = this.state.complexHealthData;
        }
        let HEALTHILLDATA = data.filter(item => item.body. 检测结果 === ' 患病');
        let HEALTHCARRYDISEASE = data.filter(item => item.body. 检测结果 === ' 携带致
        let HEALTHTYPE = data.filter(item => item.body. 检测结果 === ' 未携带致病突变'
        // console.log(' 单基因疾病', data,' 患病', HEALTHILLDATA, ' 携带病', HEALTHCA
        let HIGHCOMPLEXDATA = complexData.filter(item => item.body. 检测结果 === ' 高
        let NORMALCOMPLEXDATA = complexData.filter(item => item.body. 检测结果 === '
        let NOCOMPLEXDATA = complexData.filter(item => item.body. 检测结果 === ' 未检
        // console.log(' 复杂基因疾病', complexData,' 高风险', HIGHCOMPLEXDATA, ' 普通

        // 单基因患病
        let illData = HEALTHILLDATA.map((item,index)=>{
            return(
                <section className='health-atrisk' key={index}>
                    <Link to={{
                        pathname: `/${this.props.match.params.type}/condition`,
                        state: {
                            data: item,
                            dog_name: this.state.dog_name,
                            showAction: this.state.showAction,
                            showBreed: this.state.showBreed,
                            showHealth: this.state.showHealth,
                            key: this.state.key
                        }
                    }} className='health-condition-item health-list'>
                        <h3 className='top-title'>
                            <span className='health-icon-small span-red'></span>
                            {`${item.indexName}: ${item.site} (${item.body. 基因名})`

                        </h3>

                        <p className='small-gray-text disclosure-red'>{item.type}</p>
                        <p>
                            {item.describe}
                        </p>
                        <p className='small-bottom-text'>
                            {item.conclusion}
                        </p>
                    </Link>
                </section>
            )
        });
        // 单基因携带
        let carryDisease = HEALTHCARRYDISEASE.map((item,index)=>{
            return(
                <section className='health-atrisk' key={index}>
```

```
                    <Link to={{
                        pathname: `/${this.props.match.params.type}/condition`,
                        state: {
                            data: item,
                            dog_name: this.state.dog_name,
                            showAction: this.state.showAction,
                            showBreed: this.state.showBreed,
                            showHealth: this.state.showHealth,
                            key: this.state.key
                        }
                    }} className='health-condition-item health-list'>
                        <h3 className='top-title'>
                            <span className='health-icon-small span-blue'></span>
                            {`${item.indexName}: ${item.site} (${item.body. 基因名})`
                        </h3>

                        <p className='small-gray-text disclosure-blue'>{item.type}</p
                        <p>
                            {item.describe}
                        </p>
                        <p className='small-bottom-text'>
                            {item.conclusion}
                        </p>
                    </Link>
                </section>
            )
        });
        // 单基因未携带
        let descriptive = HEALTHTYPE.map((item,index)=>{
            return(
                <section className='health-atrisk' key={index}>
                    <Link to={{
                        pathname: `/${this.props.match.params.type}/condition`,
                        state: {
                            data: item,
                            dog_name: this.state.dog_name,
                            showAction: this.state.showAction,
                            showBreed: this.state.showBreed,
                            showHealth: this.state.showHealth,
                            key: this.state.key
                        }
                    }} className='health-condition-item health-list'>
                        <h3 className='top-title'>
                            <span className='health-icon-small span-green'></span>
                            {`${item.indexName}: ${item.site} (${item.body. 基因名})`
                        </h3>

                        <p className='small-gray-text disclosure-green '>{item.type}<
                        <p>
                            {item.describe}
                        </p>
                        <p className='small-bottom-text'>
                            {item.conclusion}
                        </p>
                    </Link>
                </section>
            )
        });
        // 复杂高风险
        let highComplexData = HIGHCOMPLEXDATA.map((item,index)=>{
            return(
                <section className='health-atrisk' key={index}>
                    <Link to={{
                        pathname: `/${this.props.match.params.type}/condition`,
                        state: {
                            data: item,
                            dog_name: this.state.dog_name,
                            showAction: this.state.showAction,
                            showBreed: this.state.showBreed,
                            showHealth: this.state.showHealth,
                            key: this.state.key
                        }
                    }} className='health-condition-item health-list'>
                        <h3 className='top-title'>
                            <span className='health-icon-small span-red'></span>
                            {item.name}
                        </h3>

                        <p className='small-gray-text disclosure-red'>{item.type}</p>
                        <p>
```

```
                            {item.describe}
                        </p>
                        <p className='small-bottom-text'>
                            {item.conclusion}
                        </p>
                    </Link>
                </section>
            )
        });
        // 复杂普通
        let normalComplexData = NORMALCOMPLEXDATA.map((item,index)=>{
            return(
                <section className='health-atrisk' key={index}>
                    <Link to={{
                        pathname: `/${this.props.match.params.type}/condition`,
                        state: {
                            data: item,
                            dog_name: this.state.dog_name,
                            showAction: this.state.showAction,
                            showBreed: this.state.showBreed,
                            showHealth: this.state.showHealth,
                            key: this.state.key
                        }
                    }} className='health-condition-item health-list'>
                        <h3 className='top-title'>
                            <span className='health-icon-small span-red'></span>
                            {item.name}
                        </h3>

                        <p className='small-gray-text disclosure-red'>{item.type}</p>
                        <p>
                            {item.describe}
                        </p>
                        <p className='small-bottom-text'>
                            {item.conclusion}
                        </p>
                    </Link>
                </section>
            )
        });
```

```
// 复杂未检测
let noComplexData = NOCOMPLEXDATA.map((item,index)=>{
    return(
        <section className='health-atrisk' key={index}>
            <Link to={{
                pathname: `/${this.props.match.params.type}/condition`,
                state: {
                    data: item,
                    dog_name: this.state.dog_name,
                    showAction: this.state.showAction,
                    showBreed: this.state.showBreed,
                    showHealth: this.state.showHealth,
                    key: this.state.key
                }
            }} className='health-condition-item health-list'>
                <h3 className='top-title'>
                    <span className='health-icon-small span-green'></span>
                    {item.name}
                </h3>

                <p className='small-gray-text disclosure-green'>{item.type}</
                <p>
                    {item.describe}
                </p>
                <p className='small-bottom-text'>
                    {item.conclusion}
                </p>
            </Link>
        </section>
    )
});

let adviceData = [...HEALTHILLDATA, ...HEALTHCARRYDISEASE, ...HIGHCOMPLEXDATA
let petHospital = adviceData.map((item,index)=>{
    return(
        <li key={index}>
            <span>{item.name}:</span>
            {item.raise}
        </li>
    )
```

```
        })
        // console.log(this.state.advice);
        let advice = this.state.advice;
        return (
            <div className="health">
                <Header
                    tab="health"
                    dog_name={this.state.dog_name}
                    showAction={this.state.showAction}
                    showBreed={this.state.showBreed}
                    showHealth={this.state.showHealth}
                />
                <div className='health-warpper'>
                    <Tabs defaultActiveKey={this.state.key} onChange={this.tabChange}
                        <TabPane tab=" 单基因疾病" key="1">
                            <section>
                                <div className='health-summary'>
                                    <h3>单基因疾病</h3>
                                    <div className='health-summary-content'>
                                        <div className='health-summary-left'>
                                            <div className='text-red'>{HEALTHILLDATA.
                                            <div className='text-red-bottom'>{HEALTHS
                                        </div>
                                        <div className='health-summary-mid'>
                                            <div className='text-blue'>{HEALTHCARRYDI
                                            <div className='text-blue-bottom'>{HEALTH
                                        </div>
                                        <div className='health-summary-right'>
                                            <div className='text-green'>{HEALTHTYPE.l
                                            <div className='text-green-bottom'>{HEALT
                                        </div>
                                    </div>
                                    <div className='health-bottom-text'>{/*HEALTHCONC
                                </div>
                            </section>
                            <section className='health-atrisk'>
                                <img src={HealthAtrisk} alt="" className='health-icon
                                <h3 className="title">
                                    患病
                                </h3>
                            </section>
                            {
                                this.state.healthData
                                ?
                                <div style={{marginBottom: '26px'}}>{illData}</div>
                                :
                                <section className='health-atrisk'>
                                    <List />
                                </section>
                            }
                            <section className='health-atrisk text-blue '>
                                <img src={HealthCarrierge} alt="" className='health-i
                                <h3 className="title">
                                    携带致病突变
                                </h3>
                            </section>
                            {
                                this.state.healthData
                                ?
                                <div style={{marginBottom: '26px'}}>{carryDisease}</d
                                :
                                <section className='health-atrisk'>
                                    <List />
                                </section>
                            }
                            <section className='health-atrisk text-green'>
                                <span className='health-icon health-img-green'></span
                                <h3 className="title">
                                    未携带致病突变
                                </h3>
                            </section>
                            {
                                this.state.healthData
                                ?
                                <div style={{marginBottom: '26px'}}>{descriptive}</di
                                :
                                <section className='health-atrisk'>
                                    <List />
                                </section>
                            }
```

```
</TabPane>
<TabPane tab=" 复杂基因疾病" key="2">
    <section>
        <div className='health-summary'>
            <h3>复杂基因疾病</h3>
            <div className='health-summary-content'>
                <div className='health-summary-left'>
                    <div className='text-red'>{HEALTHILLDATA.
                    <div className='text-red-bottom'>高风险</
                </div>
                <div className='health-summary-mid'>
                    <div className='text-blue'>{NORMALCOMPLEX
                    <div className='text-blue-bottom'> 普通风
                </div>
                <div className='health-summary-right'>
                    <div className='text-green'>{NOCOMPLEXDAT
                    <div className='text-green-bottom'>未检测
                </div>
            </div>
            <div className='health-bottom-text'>{/*HEALTHCONC
        </div>
    </section>
    <section className='health-atrisk'>
        <img src={HealthAtrisk} alt="" className='health-icon
        <h3 className="title">
            高风险
        </h3>
    </section>
    {
        this.state.complexHealthData
        ?
        <div style={{marginBottom: '26px'}}>{highComplexData}
        :
        <section className='health-atrisk'>
            <List />
        </section>
    }
    <section className='health-atrisk text-blue '>
        <img src={HealthCarrierge} alt="" className='health-i
        <h3 className="title">
```

```
                                    普通风险
                                </h3>
                            </section>
                            {
                                this.state.complexHealthData
                                ?
                                <div style={{marginBottom: '26px'}}>{normalComplexDat
                                :
                                <section className='health-atrisk'>
                                    <List />
                                </section>
                            }
                            <section className='health-atrisk text-green'>
                                <span className='health-icon health-img-green'></span
                                <h3 className="title">
                                    未检测出已知风险突变
                                </h3>
                            </section>
                            {
                                this.state.complexHealthData
                                ?
                                <div style={{marginBottom: '26px'}}>{noComplexData}</
                                :
                                <section className='health-atrisk'>
                                    <List />
                                </section>
                            }
                        </TabPane>
                    </Tabs>
                    <div className='stats-block'>
                        <span> 点击</span>
                        <strong onClick={this.advice}>疾病的养育建议</strong>
                        <span> 了解并与您的宠物医师分享精准宠爱<sup>TM</sup>的建议。</
                    </div>
                    <Modal
                        title=" 疾病的养育建议"
                        visible={this.state.advice}
                        onOk={this.handleAdvice}
                        onCancel={this.handleCancelAdvice}
                        footer={null}
                        width={600}
                    >
                        <div className='desc'>
                            <ul>
                                {petHospital}
                            </ul>
                        </div>
                    </Modal>
                    <div className='stats-block'>
                        <span> 点击</span>
                        <strong onClick={this.petHospital}>附近可提供协助的宠物医院</s
                        <span> 查看精准宠爱<sup>TM</sup>推荐的专业机构。</span>
                    </div>
                    <Modal
                        title=" 宠物医院"
                        visible={this.state.petHospital}
                        onOk={this.handlePetHospital}
                        onCancel={this.handleCancelHospital}
                        footer={null}
                        width={700}
                    >
                        <div className='pet-hospital'>
                            <PetHospitalView />
                        </div>
                    </Modal>
                    <ShareHtml
                        dog_name={this.state.dog_name}
                        pet_type={this.state.pet_type}
                        character={this.state.character}
                    />
                    <div className="share-btn" onClick={this.shareBtn}>
                        <img src={shareIcon}/>
                    </div>
                    {
                        this.state.render
                        ?
                        <Share
                            visible={this.state.shareVisible}
                            hideModal={this.hideModal}
                        />
                        :
                        ""
                    }
                </div>
            </div>
        );
    }
}

export default Health;
```

petHospital.js 宠物医院页

```
import React, { Component } from 'react';
import './petHospital.css';
import { PETHOSPITALINTRODUCTION } from '../../Api/data';


class PetHospitalView extends Component {
    constructor(props) {
        super(props);
        this.state = {

        };
    }

    render() {
        let hospitalView = PETHOSPITALINTRODUCTION.map((item , index) =>{
            return(
                <li className='hospital-lists' key={index}>
                    <img src={item.img} alt="" />
                    <p>{item.name}</p>
                    <p className='title' title={item.content}>
                        <span>简介：</span>
                        {item.content}
                    </p>
                    <p className='address'><span>地址：</span>{item.address}</p>
                    <p className='hospital-phone'><span>医院电话：</span>{item.phone}
                    <p className='business-hours'><span>营业时间：</span>{item.busine
                </li>
            )
        });

        return (
            <div className="pet-hospital-wapper">
                <ul>
                    {hospitalView}
                </ul>
            </div>
        );
    }
}

export default PetHospitalView;
```

advice.js 养育建议页

```
import React, { Component } from 'react';
import Header from '../Header/header';
import {Link} from 'react-router-dom';
import dogadvice from '../../img/icon_introduction_green.svg';
import _ from 'lodash';
import axios from 'axios';


const SEARTHDATA = [
    {
        name: '淀粉酶',
        summary: '淀粉消化能力',
        text: '根据淀粉酶的养育建议'
    },
    {
        name: '行为',
        summary: '性格标签',
        text: '根据性格标签的养育建议'
    },
    {
        name: '儿童安全性',
        summary: '儿童安全性',
        text: '根据儿童安全性的养育建议'
    },
    {
        name: '基因年龄',
        summary: '相对于人类的基因年龄',
        text: '根据基因年龄的养育建议'
    },
    {
        name: '智商',
        summary: '智商指数',
        text: '根据智商的养育建议'
    }
]

class Advice extends Component {
    /**
     * 这是报告详情页面
     *
    */
    state = {
        showDog: window.location.pathname.split('/')[4],
    }

    componentDidMount() {
        let pathParams = this.props.match.params.type;
        let data = this.props.location.state.searchData;
        axios({
            method: 'get',
            url: `../data/db/交互式报告素材.json`,
            headers: {
                'Content-Type': 'application/json',
                'Cache-Control': 'max-age=86400'
            }
        })
        .then(response => {
            let display = {display: 'block', whiteSpace: 'pre-wrap'};
            let adviceResult = this.props.location.state.adviceData.map((item,index)=
                return(
                    <li key={index}>
                        <div className="list-title">{item.name}：</div>
                        <div>{item.raise}</div>
                    </li>
                )
            })

            let recommendFood =
                <li className='disease-advice'>
                    <p className='dog-print'><img src={dogadvice} alt="" /></p>
                    <span style={display}>{'食物推荐'}</span>
                    <div
                        style={display}
                        dangerouslySetInnerHTML={{__html: response.data['健康']['食
                    >
                    </div>
                </li>
            let linkTo = _.map(SEARTHDATA, (item, index) => {
                return (
                    <Link to={{
                            pathname: `/${pathParams}/detail`,
                            state: {
                                status: data[item.name].visible ? '详情': '未解锁功
                                name: item.name,
                                item: data[item.name].header.summary[item.summary],
                                body: data[item.name].body,
                                dog_name: this.props.location.state.dog_name,
                                showAction: this.props.location.state.showAction,
                                showBreed: this.props.location.state.showBreed,
                                showHealth: this.props.location.state.showHealth
                            }
                        }}
                        key={index}
                        className="link-to-detail"
                    >
                        {item.text}
                    </Link>
                )
            })
```

```
            let content =
                <ul className="list">
                    <li className='disease-advice'>
                        <span> 疾病养育建议</span>
                        <p className='dog-print'><img src={dogadvice} alt="" /></p>
                    </li>
                    {adviceResult}
                    <li className='disease-advice'>
                        <p className='dog-print'><img src={dogadvice} alt="" /></p>
                        <span> 其他养育建议</span>
                        {linkTo}
                    </li>
                    {recommendFood}
                </ul>
            this.setState({
                content
            })
        })
    }

    render() {
        return (
            <div className="detail">
                <Header
                    tab={this.props.location.state.status}
                    dog_name={this.props.location.state.dog_name}
                    showAction={this.props.location.state.showAction}
                    showBreed={this.props.location.state.showBreed}
                    showHealth={this.props.location.state.showHealth}
                />
                <div className='condition-wapper health-warpper'>
                    <div className='condition-header'>
                        <Link to={`/${this.props.match.params.type}`}>{this.props.loc
                        <i>></i>
                        <span> 科学的养育建议</span>
                    </div>
                    <div className="detail-content">
                        {this.state.content}
                    </div>
                    <ul className="render-list">
                    </ul>
                </div>
            </div>
        );
    }
}

export default Advice;
```

share.js 分享页

```
import React, { Component } from 'react';
import './share.css';
import Modal from 'antd/lib/modal';
import Button from 'antd/lib/button';
import html2canvas from 'html2canvas';
import Spin from 'antd/lib/spin';
import Icon from 'antd/lib/icon';

const antIcon = <Icon type="loading" style={{ fontSize: 24 }} spin />;

class Share extends Component {
    constructor(props) {
        super(props);
        this.state={
            render: 1
        }
    }

    componentDidUpdate () {
        let imgData = null;
        // 将#print的html转成图片
        let _this = this;
        if (this.state.render) {
            if(document.getElementById('print')) {
                html2canvas(document.getElementById('print')).then(function(canvas) {
                    imgData = canvas.toDataURL('image/jpeg');
                    // console.log('转成base64',imgData)
                    _this.setState({
                        imgData,
                        render: 0
                    })
                });
            }
        }
    }

    save = () => {
        const save_link = document.createElementNS('http://www.w3.org/1999/xhtml', 'a
        const event = document.createEvent('MouseEvents');
        let filename = '分享图片.jpeg'
        /**
         * 在本地进行文件保存
         * @param  {String} data     要保存到本地的图片数据
         * @param  {String} filename 文件名
         */
        save_link.href = this.state.imgData;
        save_link.download = filename;

        event.initMouseEvent('click', true, false, window, 0, 0, 0, 0, 0, false, fals
        save_link.dispatchEvent(event);
    }

    render() {
        return (
            <Modal
                wrapClassName="share-content"
                title="分享图片"
                style={{ top: 0, bottom: 0 }}
                visible={this.props.visible}
                onOk={this.handleOk}
                onCancel={this.props.hideModal}
                footer={null}
                destroyOnClose={true}
                width={360}
            >
                {
                    this.state.imgData
                    ?
                    <img src={this.state.imgData} style={{'marginBottom': '10px'}} />
                    :
                    <div className="example">
                        <Spin indicator={antIcon}/>
                    </div>
                }
                <Button className="save-btn" onClick={this.save}> 保存图片分享</Button
            </Modal>
        );
    }
}

export default Share;
```

shareHtml.js 分享 html 部分

```
import React, { Component } from 'react';
import './share.css';
import shareDog from '../../img/share_dog.jpg';
import shareCat from '../../img/share_cat.jpg';
import QRCode from 'qrcode-react';
import shareLogo from '../../img/qrcode-logo.jpg';

class ShareHtml extends Component {
    constructor(props) {
        super(props);

        this.state={

        }
    }

    render() {
        let shareImg = '';
        if (this.props.pet_type === '犬') {
            shareImg = shareDog;
        } else if (this.props.pet_type === '猫') {
            shareImg = shareCat;
        }
        const htmlDisplay = {
            'position': 'fixed',
            'top': '-9999px',
            'background': `url(${shareImg}) no-repeat`,
            'width': '750px',
            'height': '1334px',
            'backgroundSize': 'cover',
            'marginBottom': '10px'
        }
        const url = window.location.href;
        return (
            <div id="print" style={htmlDisplay}>
                <div className="name">我家主子<span>{this.props.dog_name}</span> 属于
                <p className="character">{this.props.character}</p>
                <div className="qr-code">
                    <QRCode
                        value={url}
                        size={200}
                    />
                    <img src={shareLogo} />
                </div>
            </div>
        );
    }
}

export default ShareHtml;
```

“‘