

使用如下命令安装Axios：

```
1 $ npm install axios
2 $ yarn add react-native-axios
```

安装antdUI组件库：

```
1 $ npm install antd-mobile --save
```

直接使用Axios进行请求时，为了完成请求地址拼接，参数设置，异步操作处理，JSON格式转化等等操作，将会产生许多冗余代码，为了简化代码，需要对Axios进行二次封装：

- 在src目录下创建utils目录，该目录下创建http目录
- 在http目录下创建文件 `httpBaseConfig.js` 与 `request.js`
 - `httpBaseConfig.js` 用于配置服务器域名，端口号，API地址
 - `request.js` 用于编写Axios请求逻辑

Axios基于ES6中的Promise对象进行开发，因此可以使用then链来处理同步问题，而ES7加入async函数后，可以在async函数中使用await关键词实现更方便的处理，await会阻塞后续代码直到得到返回的Promise对象，具体可以参考如下博客：

[理解 JavaScript 的 async/await - SegmentFault 思否](#)

Axios还为我们提供了方便的基础设置、拦截器等操作，通过设置回调函数可以完成发送请求前，和得到返回的数据后进行处理。

最后我们将不同类型的请求封装到一个http类中。

最后得到的 `request.js` 如下：

```
1 import axios from "axios";
2 import baseConfig from "./httpBaseConfig";
3
4 // 默认域名
5 axios.defaults.baseURL =
6     baseConfig.baseUrl + ":" + baseConfig.port + baseConfig.prefix;
7 // 默认请求头
8 axios.defaults.headers["Content-Type"] = "application/json";
9
10 // 响应时间
11 axios.defaults.timeout = 10000;
12
13 // 请求拦截器
14 axios.interceptors.request.use(
15     (config) => {
16         // TODO:在发送前做点什么
17         // showLoading(); //显示加载动画
18         return config;
19     },
20     (error) => {
21         // hideLoading(); //关闭加载动画
22         // TODO:对响应错误做点什么
23         return Promise.reject(error);
24     }
25 );
26
27 // 响应拦截器
28 axios.interceptors.response.use(
29     (response) => {
30         // TODO:请求返回数据后做点什么
31         if (response.status === "200" || response.status === 200) {
32             return response.data.data || response.data;
33         } else {
34             // TODO:请求失败后做点什么
35             throw Error(response.opt || "服务异常");
36         }
37         return response;
38     }
39 );
```

```
38     },
39     (error) => {
40         // TODO:对响应失败做点什么
41         return Promise.resolve(error.response);
42     }
43 );
44
45 // 请求类
46 export default class http {
47     // ES7异步get函数
48     static async get(url, params) {
49         try {
50             let query = await new URLSearchParams(params).toString();
51             let res = null;
52             if (!params) {
53                 res = await axios.get(url);
54             } else {
55                 res = await axios.get(url + "?" + query);
56             }
57             return res;
58         } catch (error) {
59             return error;
60         }
61     }
62
63     static async post(url, params) {
64         try {
65             let res = await axios.post(url, params);
66             return res;
67         } catch (error) {
68             return error;
69         }
70     }
71
72     static async patch(url, params) {
73         try {
74             let res = await axios.patch(url, params);
```

```

75         return res;
76     } catch (error) {
77         return error;
78     }
79 }
80
81 static async put(url, params) {
82     try {
83         let res = await axios.put(url, params);
84         return res;
85     } catch (error) {
86         return error;
87     }
88 }
89
90 static async delete(url, params) {
91     /**
92      * params默认为数组
93      */
94     try {
95         let res = await axios.post(url, params);
96         return res;
97     } catch (error) {
98         return error;
99     }
100 }
101 }

```

`httpBaseConfig.js` 中的配置如下：

```

1 export default httpBaseConfig = {
2     baseUrl: 'http://www.*****.***',
3     port: '****',
4     prefix: '/AppServer/ajax/'
5 }

```

最后使用时，调用请求后，我们得到的将是一个Promise对象，使用then链将其保存到状态中即可完成数据显示：

```

1  handleRequest() {
2      let param = {
3          userName: "mingming",
4          classTimeId: "50648",
5          type: "3",
6          // 'callback': 'ha'
7      };
8      http.get("teacherApp_lookNotice.do", param)
9          .then((res) => {
10              console.log(res);
11              let data = JSON.parse(res);
12              this.setState({
13                  message: data.message,
14              });
15              console.log(data);
16          })
17          .catch((error) => {
18              console.log(error);
19          });
20  }

```

最终打印到控制台的结果如下：

```

1  {"data": {"author": "明茗", "content": "这是一个测试通知测试通知测试通知测试通知测试通知测试通知,测试***功等等,还有***,和大***等等", "isAuthor": true, "isUpdate": false, "noReadNum": 53, "noreadList": ["索夏利", "何一繁", "段莹", "孙亮亮", "李冯石", "贺玉婷", "张立新", "龚夏萌", "刘驰誉", "王玲", "张俊", "王楠", "姜克杰", "孙丽园", "李波", "代麦玲", "李妮", "李坤江", "李杰", "黄运科", "陈雨菲", "黄萍", "王致远", "李杰", "柯团团", "陈雯慧", "彭思毅", "张昌", "段怡欣", "管雅", "严彤鑫", "徐文莉", "朱景洲", "刘乔瑞", "王子豪", "孙红", "赵美婷", "李雕坛", "黄楠", "张静静", "刘祎璠", "冯健强", "王俊杰", "张辉", "彭诗雨", "叶刚", "何萍", "何健", "王锦婷", "周骏", "杨千骏", "李娇", "郭聪聪"], "num": 60, "readList": ["李龙龙", "杨文选", "刘佳璇", "方建辉", "卢文静", "左亚东", "李盈斌"], "readNum": 7, "title": "测试通知"}, "message": "数据保存成功!", "success": true}

```