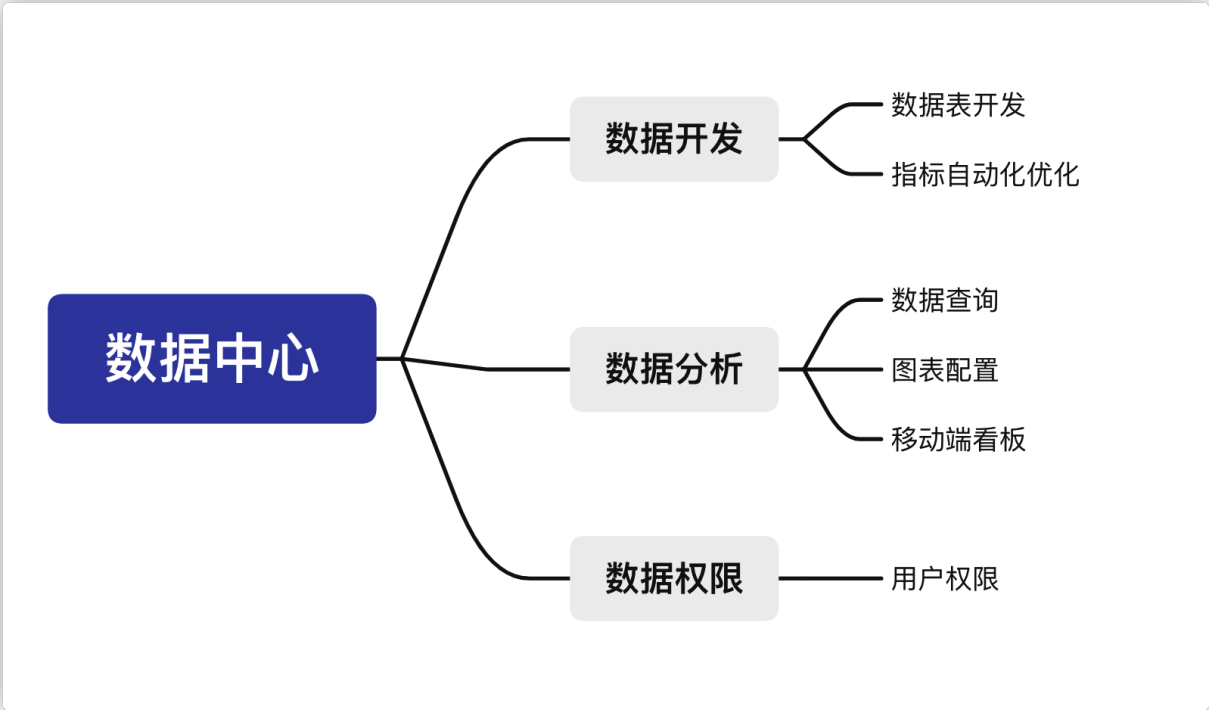


需求总览



需求名称	时间	概述
数据分析模块（移动端自定义看板）	9.03-9.29	同步pc端看板配置，在移动设备上进行筛选与横屏查看
数据开发模块（视图表&文件上传）	8.20-8.31	数据表管理与建表配置流程控制，支持使用上传文件作为数据源进行建表配置

移动端自定义看板

设计

交互流程：

看板列表：

蜂报

全部

供应链-财务

基础组件

1113333

报表负责人: yin.qi

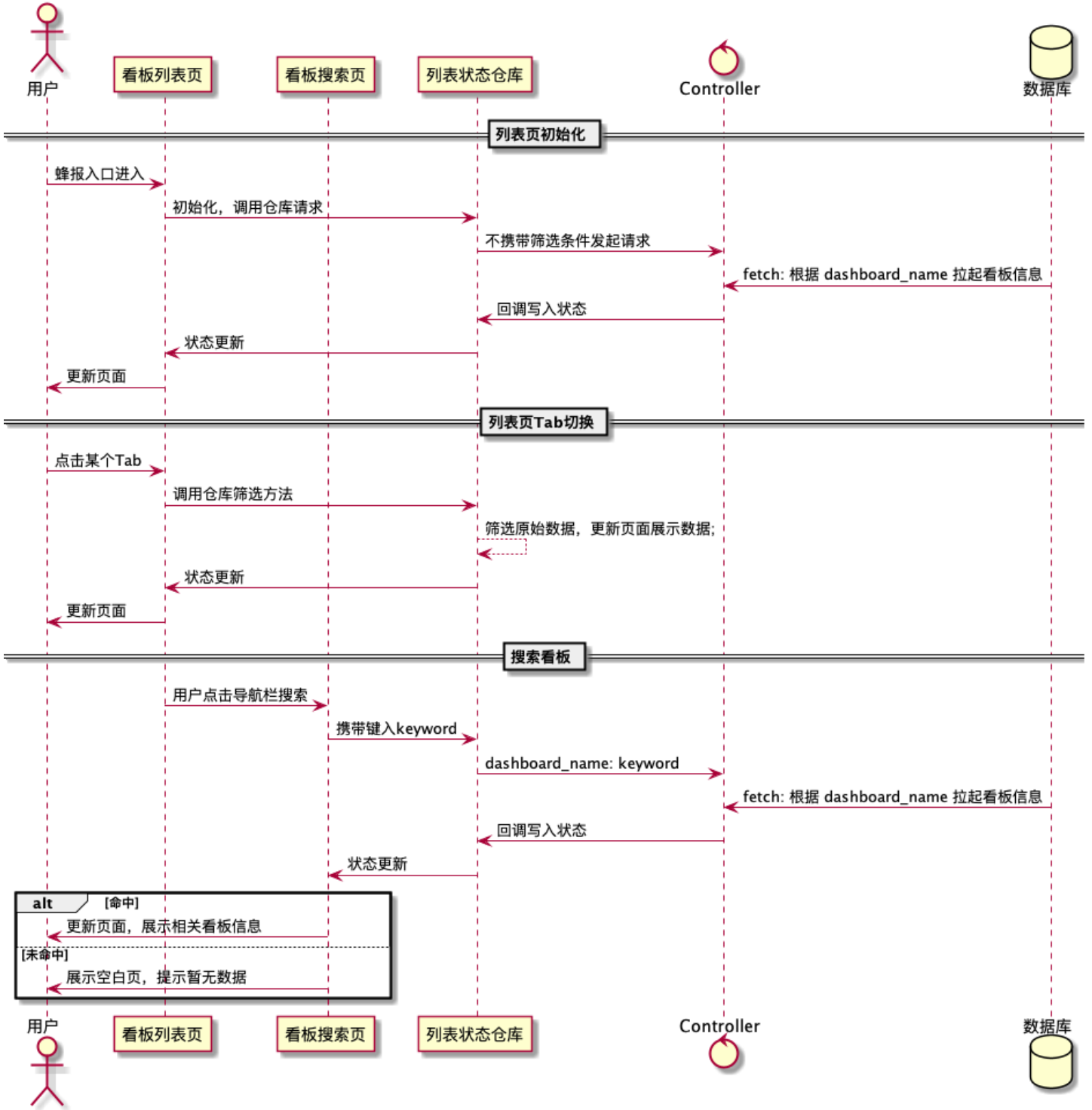
111

报表负责人: yin.qi

蜂报测试看板

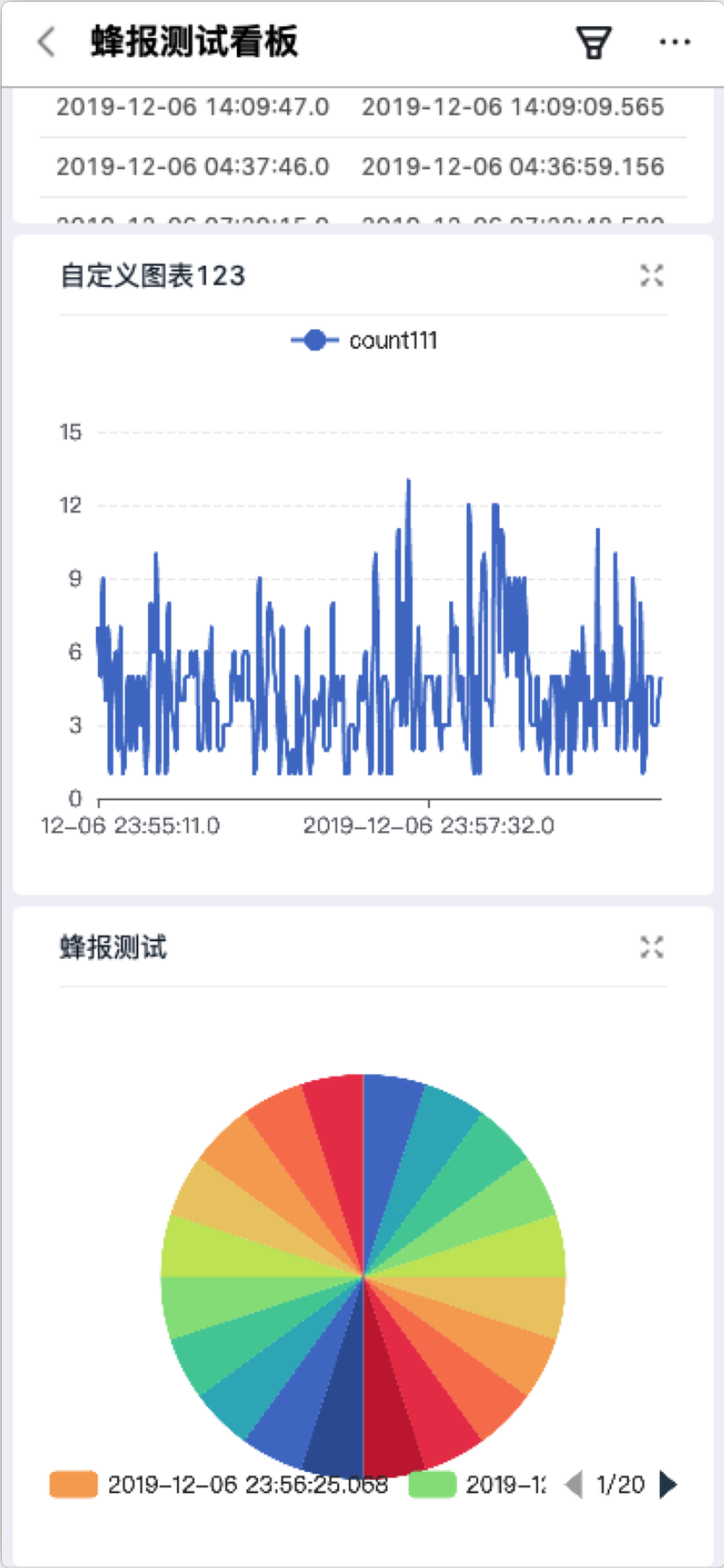
报表负责人: 111

看板列表模块



看板详情：





时间范围

开始时间

—

结束时间

取数范围

开始时间

—

结束时间

app\_version

展开

801026980100058010025

801026880100268010002

801006680100208010229

ws\_pos\_id

展开

249352493424936

230452522325347

229932437824370

count111

10

—

1000

client\_time

开始时间

—

结束时间

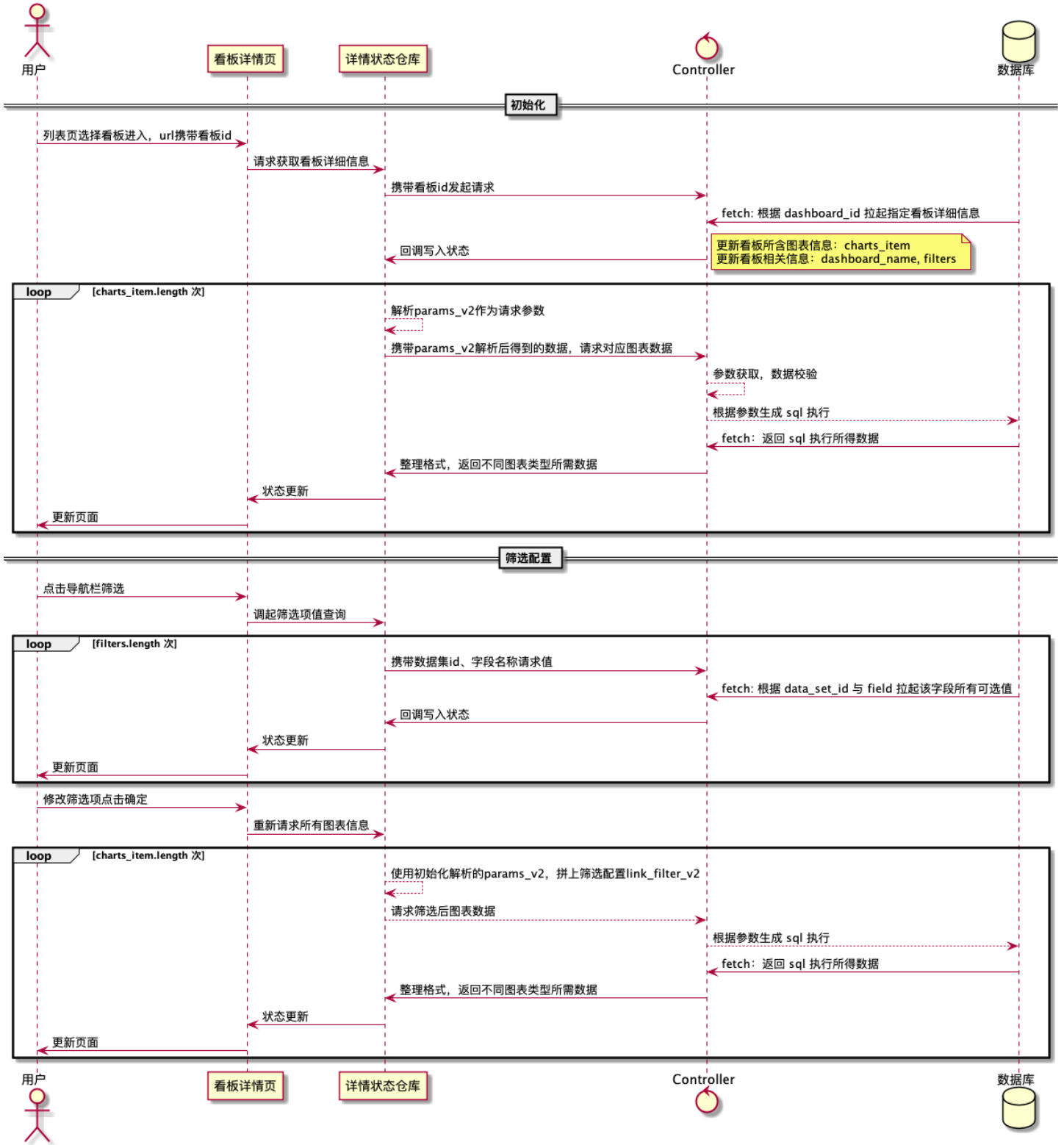
gid

展开

重置

确定

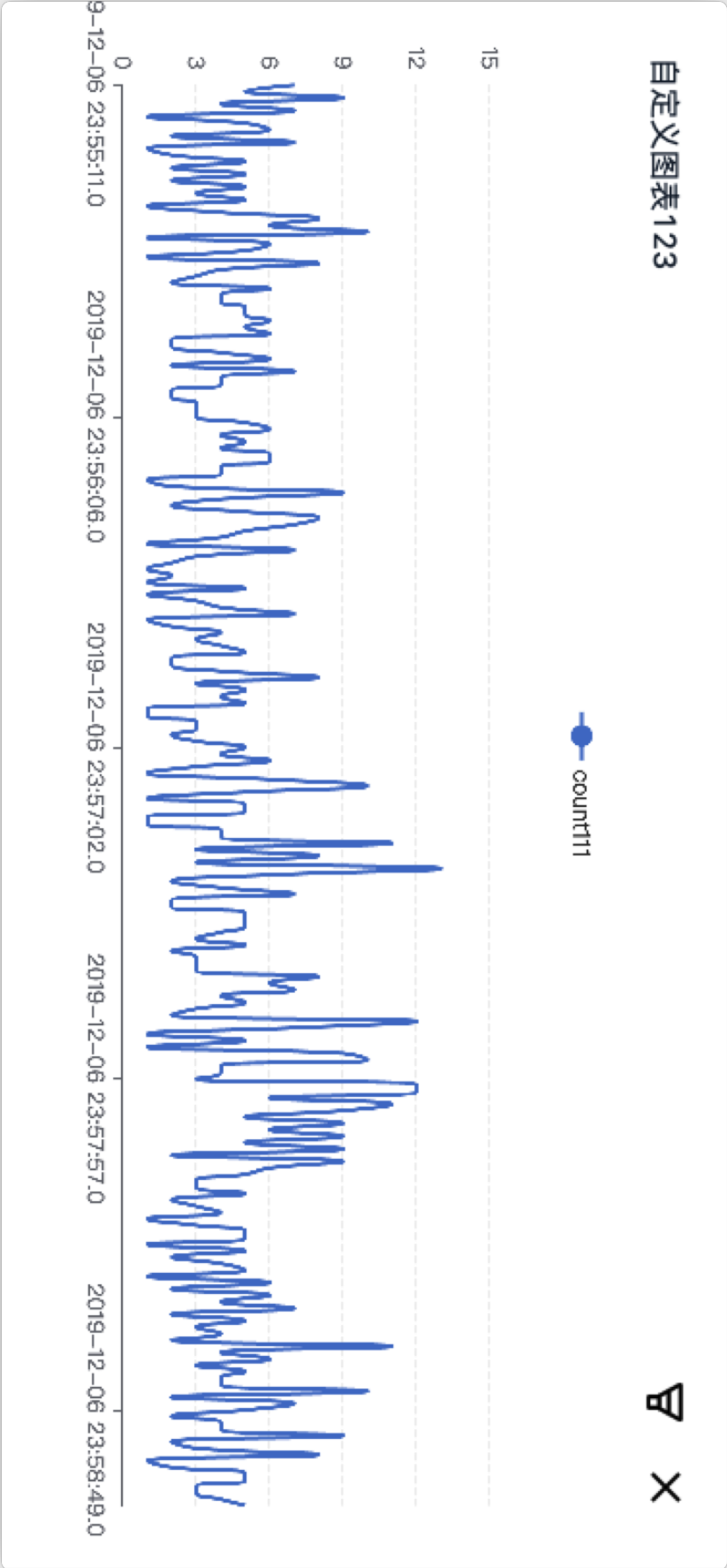
看板详情模块

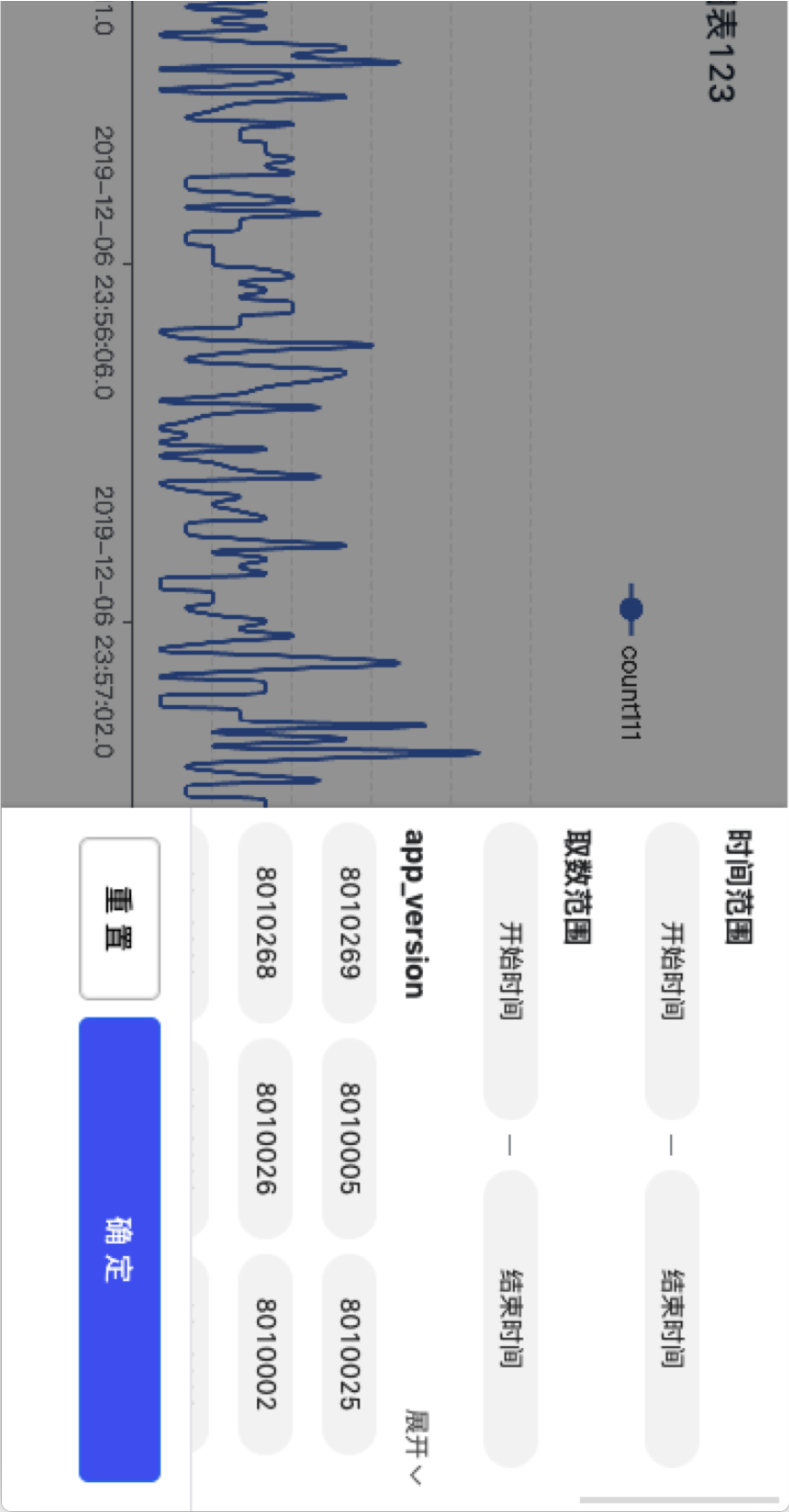


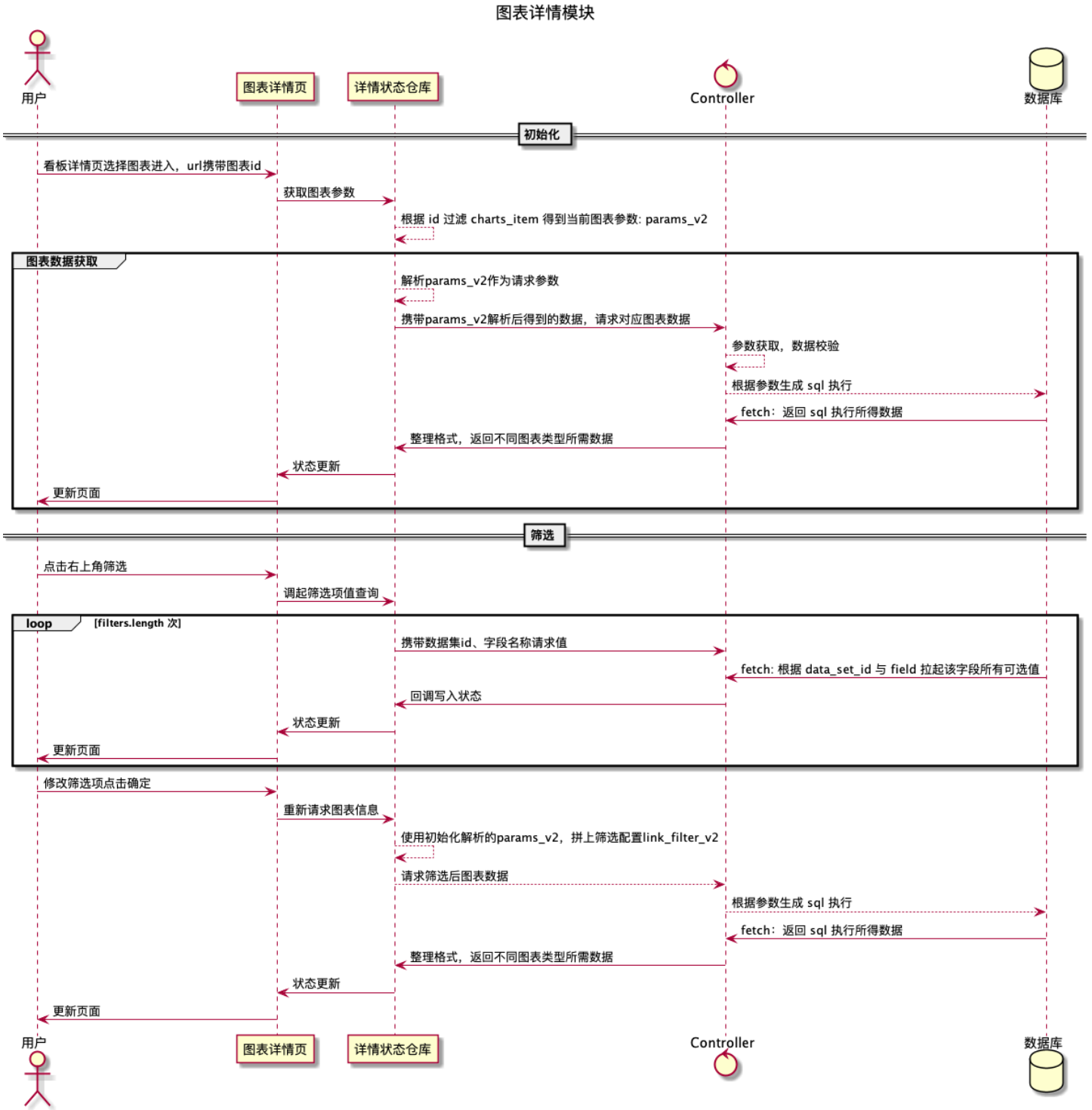
图表详情：



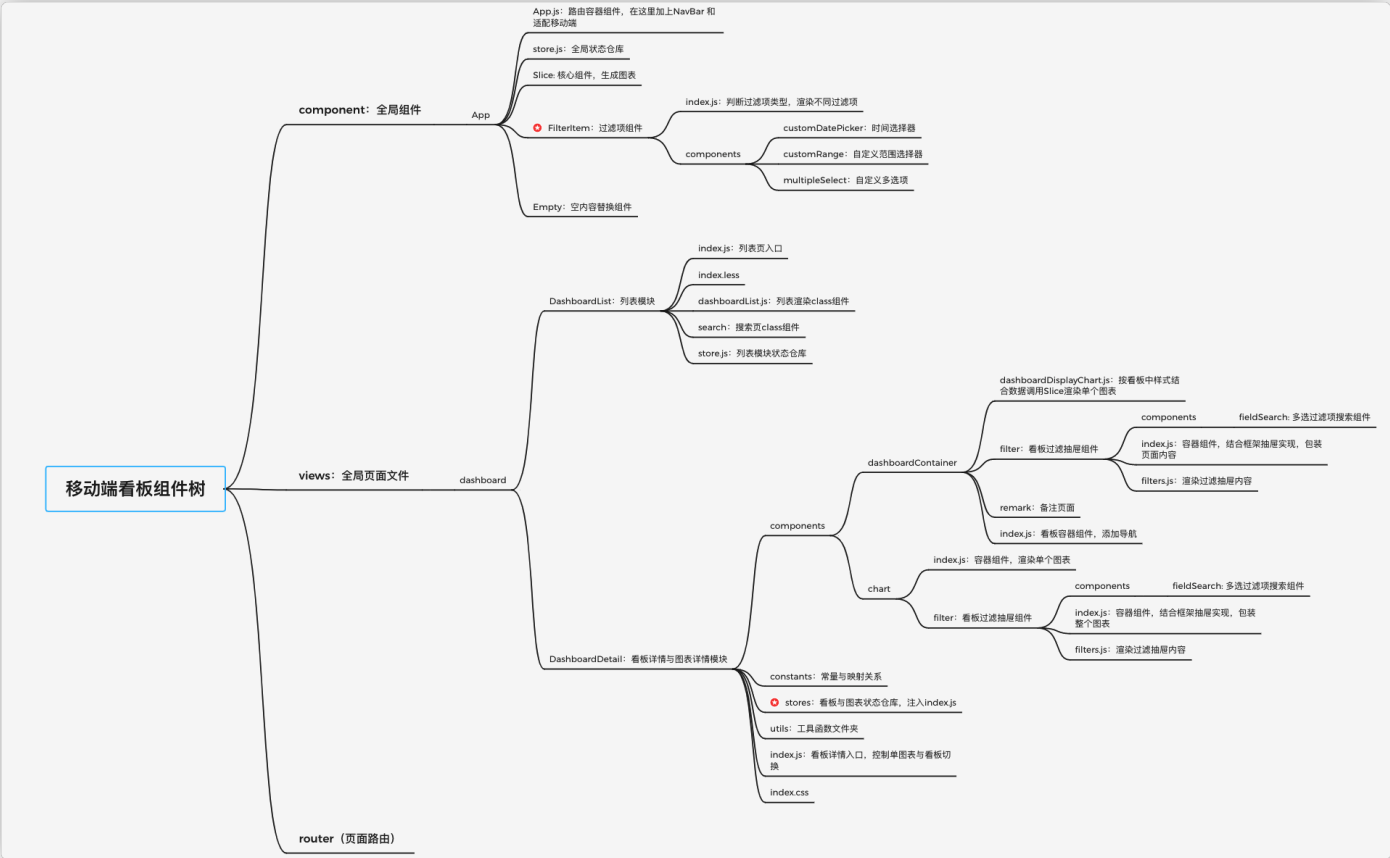








代码结构：



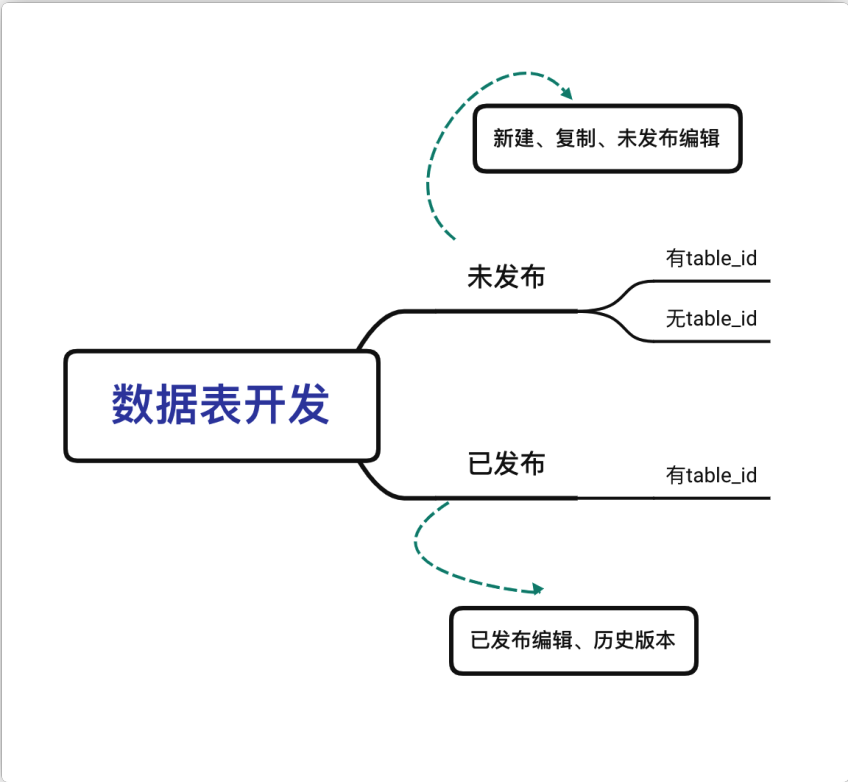
场景：  
不同类型筛选项修改，状态收集

## 视图表&文件上传

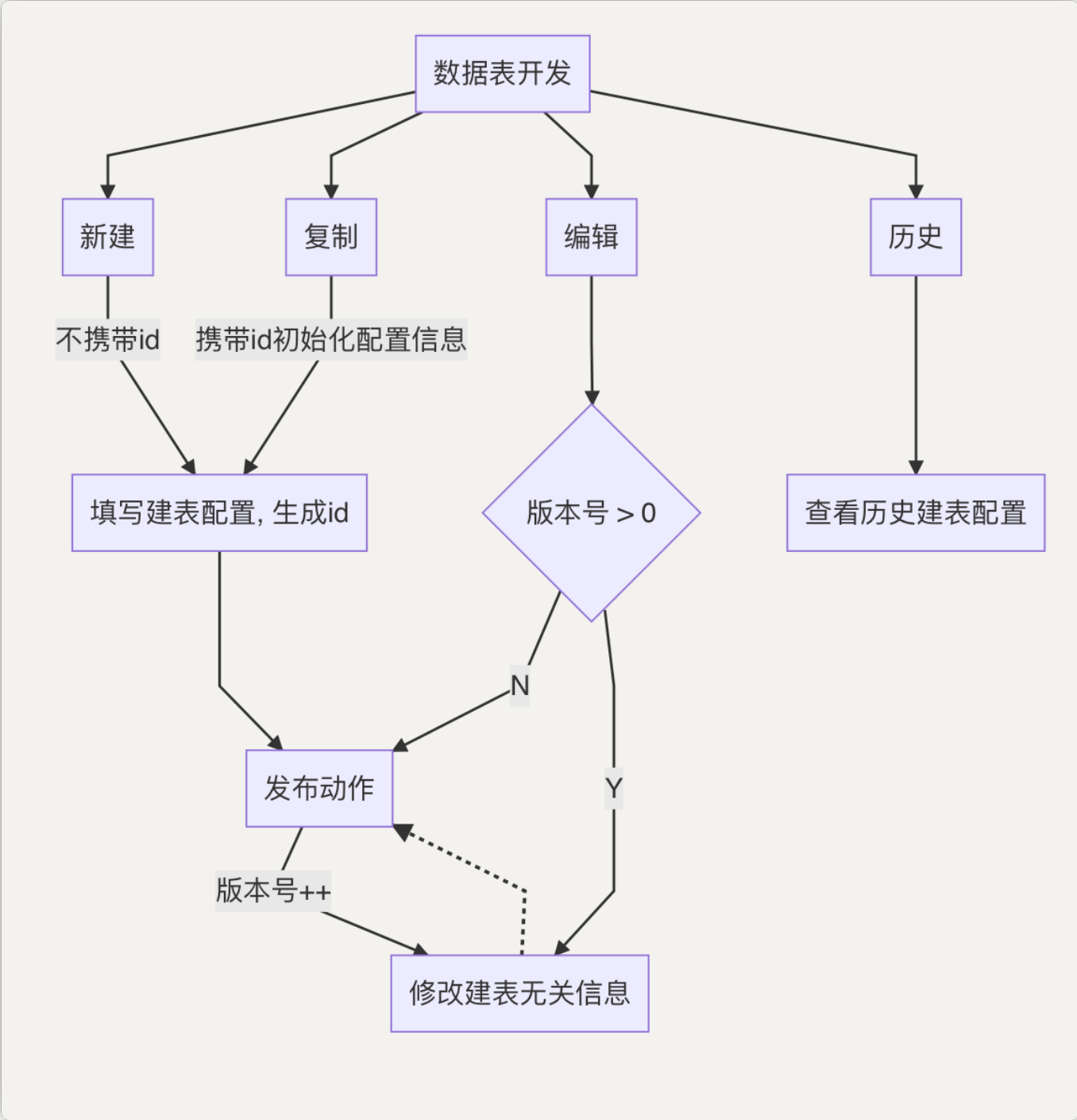
- 场景：
- 1. 状态控制
  - 2. 文件上传

## 状态控制

状态：



业务流程：



## 文件上传

失败尝试：上传控件 -> file对象 -> 二进制流, readAsBinaryString() -> body (fail)

成功尝试：使用FormData数据类型 -> append('file', file对象) -> Request设置data: { formdata: true } -> content-type: multipart/form-data

Upload组件api:

```
<Field label="选择文件" labelClass="w6em" className="w300">
  <Upload
    name="file"
    accept=".txt, .csv, .xlsx"
    beforeUpload={this.saveFile}
    onRemove={this.removeFile}
    fileList={fileList}
  >
    <A>点击选择上传文件</A>
  </Upload>
</Field>
```

Request:

```
/**
 * 请求体设置钩子
 *
 * 由于本钩子实际上最终生成了数据形式，为了方便业务操作
 * 本钩子置于所有请求中间件之后
 */
Request.unuse.contentTypeRequestHook = Request.useAfterPrivateHooks('send', function contentTypeRequestHook(next) {
  const { req } = this;

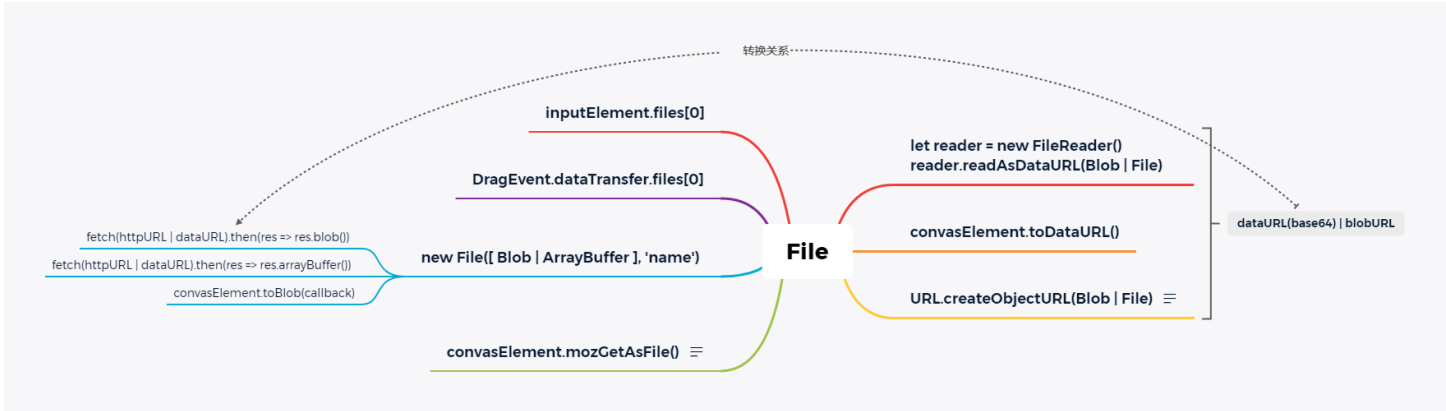
  if (req.formData) {
    req.body = req.data;
  } else if (/^get$/i.test(req.method)) {
    req.url = appendUrl(req.url, stringify(req.data));
  } else if (req.formData) {
    req.body = req.data;
  } else if (req.form) {
    req.headers.set('Content-Type', 'application/x-www-form-urlencoded;charset=UTF-8');
    req.body = stringify(req.data);
  } else {
    req.headers.set('Content-Type', 'application/json;charset=UTF-8');
    req.body = JSON.stringify(req.data);
  }

  req.credentials = req.credentials || 'same-origin';

  next();
});
```

基于xhr 2提供的FormData接口生成FormData对象，使用append (k, v) 添加表单项，此时v可以为file对象（该对象基于Blob，扩展了支持用户上传文件的功能，存储文件的相关信息，可通过使用Blob的api进行读取处理，如FileReader）。

将FormData对象作为xhr请求的body时，请求内容格式content-type会被自动设置为 multipart / form-data，此时文件随表单以数据流的格式上传。



## 总结

- 1. 合理的设计可以减少开发时的成本，设计时对业务动作的思考也可以帮助避免在实现时容易忽略的问题。
- 2. 我们所使用的框架、组件，时常不能完美契合需求场景，需要有实现自定义封装，深入底层的能力。
- 3. 一个完整的需求，需要与设计、产品、后端有充分的沟通，避免开发到后期理解错误带来的时间损失。

# 学习 & 产出

React-hooks: <https://wiki.corp.bianlifeng.com/pages/viewpage.action?pageId=683639109>

TypeScript: [https://github.com/1257668543/company\\_study/tree/master/TS](https://github.com/1257668543/company_study/tree/master/TS) (进行中)

Like Be the first to like this