



机械工业出版社
华章分社

8月27日 | 49届前端早早聊

无/低代码 资本热宠 搭建卷城

搭建原理
低代码实践
业务适配
提效流程
生产力工具



2022 全年行程 ▶

1/29 前端职业前景	2/26 TS技术专场	3/19 管理专场	4/9 面试面试专场	4/23 组件专场	5/14 监控专场	5/28 性能专场	6/18 成长与晋升	6/25 跨端专场	7/16 海外工作专场	7/17 一站式基础	8/6 工程化专场	8/27 低代码专场	9/3 低代码进阶专场	9/17 音视频专场	9/24 创业专场	10月 开发语言专场	10月 健康健康专场	11月 自由职业专场	11月 框架专场	12月 运维专场	12月 测试专场	12月 优化专场
----------------	----------------	--------------	---------------	--------------	--------------	--------------	---------------	--------------	----------------	---------------	--------------	---------------	----------------	---------------	--------------	---------------	---------------	---------------	-------------	-------------	-------------	-------------

09:00 编排:《从界面到逻辑-低代码逻辑编排的思考和实践》

姜天意

腾讯
CSIG-医疗健康事业部-前端负责人

10:00 商业:《低代码体系组合拳:最低成本实现商业化》

谭景琳

铜鹅精 CRM
业务系统组-前端负责人

11:00 应用:《低代码企业应用的建设与实践》

钟柳

兴盛优选
前端工程师

12:00 开发流:《营销场景下的低代码建设》

周頌 Jojo

自由开发者

13:00 业务:《复杂业务场景下的低代码实践》

鱼鱼

欧米云
共享平台研发团队-低代码平台前负责人

14:00 营销:《字节游戏营销平台低代码应用实践》

谢军令

字节跳动
游戏引擎-营销平台负责人

15:00 工具:《跨境 B2B 导购搭建工具》

高凯

行云集团
体验技术团队-前端专家

16:00 智能:《智能生成:CODE.FUN 一键从设计稿到源代码》

杨帆

CODE.FUN
CODE.FUN 创始人

17:00 UI:《樟卯:可扩展的低代码工具开发框架》

余彦臻

SmartX
高研中台与内部效率团队-研发负责人

18:00 中后台:《如何利用低代码快速高效搭建中后台页面》

小虫

芒果TV
运营产品经理-前端工程师

19:00 方案:《低代码:基于 React 和 Mobx 的实现》

千瑜

Akulaku
部门经理-前端专家

20:00 智能:《低码化与智能化融合》

文子祺

兴盛优选
前端运营技术部-低代码智能化负责人

21:00 UI:《基于现有 UI 框架快速落地低代码框架》


王浩

善诊
高级产品经理-低代码开发

22:00 引擎:《B 端页面的低代码引擎之路》

吕洋

腾讯
美团配送-美团配送工程师



基于现有UI框架快速落地低代码框架

王浩 @善诊

最小闭环

● DSL 设计 ——— ● **PART 1** ——— ● 交互行为

页面渲染

DSL 设计 - 基础组件

Info

Success

Warning

Error

组件库里已有的组件粒度已经足够细了，挑选常用的做好映射就足够了

```
1 // DLS
2 {
3   _tag: 'div',
4   body: [
5     { _tag: 'Button', title: 'Info', type: 'info' },
6     { _tag: 'Button', title: 'Success', type: 'success' },
7     { _tag: 'Button', title: 'Warning', type: 'warning' },
8     { _tag: 'Button', title: 'Error', type: 'error' }
9   ]
10 }
11
12 // 代码 getContent = div.body
13 <component
14   v-for="(item, index) in getContent"
15   :key="index"
16   :is="item._tag"
17   :data="item"
18 ></component>
```

DSL 设计 - 高阶组件



通常为业务组件、不同于基础组件，业务组件内部的交互会更丰富和复杂一些

```
1 // DSL
2 {
3   _tag: 'div',
4   body: [
5     {
6       _tag: 'info-card',
7       ajax: '/mock/info-card',
8       content: [
9         {
10          tips: '这个字段表示历史用户总量',
11          label: '历史用户总量',
12          key: 'total',
13          chart: 'totalChart'
14        },
15        {
16          label: '今日健康小管家添加',
17          key: 'TodayHousekeeperCount',
18          chart: 'todayHousekeeperCountChart'
19        }
20      ]
21    }
22  ]
23 }
```

页面渲染 - 组件逐级渲染

客户端

```
1 <template>
2   <Page :data="pageConfig"></Page>
3 </template>
4
5 // pageConfig = {
6 //   _tag: 'page',
7 //   body: [
8 //     { _tag: 'txt', content: 'Page 1 txt 1' },
9 //     { _tag: 'txt', content: 'Page 1 txt 2' },
10 //   ],
11 //   _tag: 'page',
12 //   body: [ { _tag: 'txt', content: 'Page 2 txt 1' } ]
13 // }
14 // |
15 // }
```

渲染页面

_tag: page

_tag:txt

Page 1 txt 1

_tag:txt

Page 1 txt 2

_tag: page

_tag:txt

Page 2 txt 1

Page 组件

```
1 <template>
2   <div class="page">
3     <component
4       v-for="(item, index) in data.body"
5       :data="item"
6       :is="item.tag"
7       :key="index"
8     ></component>
9   </div>
10 </template>
```

Txt 组件

```
1 <template>
2   <div class="txt">[[data.content]]</div>
3 </template>
```

交互行为

DSL

```
1 {
2   _tag: 'page',
3   content: [
4     {
5       tag: 'Button',
6       title: '下一页(抽屉形式)',
7       onEvent: {
8         click: {
9           action: {
10            type: 'page',
11            name: 'show',
12            pageName: 'drawPage'
13          }
14        }
15      ]
16    }
17  ]
18  childPageList: [
19    {
20      name: 'drawPage',
21      // ...
22    }
23  ]
24 }
```



行为：点击按钮打开 名称为“drawPage”的页面

Button.vue

```
1 <template>
2   <Button
3     v-bind="data"
4     @click="$emit('on-action', data?.onEvent?.click?.action)"
5   >{{data.title}}</Button>
6 </template>
7
8 <script>
9   export default {
10     name: 'button',
11     props: {
12       // 组件配置
13     },
14     data: {
15       type: Object,
16       default () => {}
17     },
18   },
19 </script>
```

回到过去

—— PART 2 ——

遇到了什么问题

- 探索型业务、功能点多而杂，常伴随着更多的后台需求
- 研发资源紧张、但仍浪费了很多时间在重复劳动上
- 交互没有规范，产品肆意妄为



不是现在不是现在，两边大佬请无视



一个周五的下午

写下了第一个版本，主要思路是一个页面拆分为不同的区块，对应不同的组件，算是一个典型的 CRUD 页面



1、表单区域

2、操作区域

3、表格区域

当前状态	用户数量	创建人	备注	创建时间	操作
正常		testersif	新增8-12	2019-05-12 17:16:46	新增 删除
正常		testersif	正常已支付已付款	2019-05-10 18:57:27	新增 删除
正常		testersif	正常已支付未付款	2019-05-10 18:58:06	新增 删除
正常		testersif	正常已支付已付款	2019-05-10 18:52:51	新增 删除
正常		testersif	正常已支付未付款	2019-05-10 18:50:39	新增 删除
正常		testersif	已支付已付款	2019-05-10 18:48:49	新增 删除
正常		testersif	未支付已付款	2019-05-10 18:47:33	新增 删除

4、页码区域

5、其它区域 (通常为弹窗, 默认不展示)

BizTable - 详情页示例

由于缺陷比较多，我们就挑一个组件来讲下那些让产品经理抓狂的设计理念

- 页面层级只有两级
- 详情页的内容只允许查看，想要修改必须关闭详情页打开编辑页
- 详情页的表格没有操作栏，不允许对表格的内容进行增、删、改
- 详情页只支持这两个组件
- ...



举步维艰

由于最初设定的限制规则太多，但产品需求本身是多变的，为了适应更多场景，只好不断的打补丁，BizTable 内部也变得越来越混乱



濒临失控

就这样坚持了一段时间后，BizTable 本身的问题愈发明显

- 一部分已使用 BizTable 实现的模块，由于添加新的功能而被迫必须使用代码重新实现
- 另外 BizTable 本身为了兼容更多场景也被改的面目全非，维护工作变得很困难

此时的 BizTable 基本处于失控状态，研发和维护成本非常高，逐步被弃用

再出发

—文档先行、**PART3**谋而后动—

用户访谈



一线用户

新人不知道怎么用，提示也看不懂，最好能支持在手机上操作，然后...

能实现我的需求就行，不要到时候这也实现不了那也实现不了



产品经理



UI / UE

希望各业务线交互统一，避免用户使用时付出额外的学习成本

用户访谈 - 确定初期研发思路

一、文档需先行

- 视觉交互、接口调用等形成规范
- 各功能模块设计思路沉淀文档

二、功能要完善

- 更快上手、更简单的复用
- 可扩展、结合编码实现定制化需求


注意：初期最核心的还是提升研发的效率，用户访谈的作用，是为了在设计阶段可以覆盖更多的业务场景，同时兼顾后面的一些需求，留好口子

方案设计与实施 - 文档先行

设计思想

设计理念

SuperTable的核心思想是，
举一个栗子，这个页面



Super Table

Tip: 根据接口返回的两种格式，
+ { "code": 0, "data": { "list": [...], "total": 100 }, "errorMsg": "" },
+ { "errorCode": 1, "errorMsg": "..." }

以上两种格式只保留内部格式，
一致。

1、表格类接口规范

请求参数
重点关注: currentPage, pageSize


```
1 {  
2   "currentPage": 1,  
3   "pageSize": 20,  
4   ... // 其他参数  
5 }
```

返回参数
接口返回: page (currentPage, pageSize, totalData), resultList

SuperTable 1.0

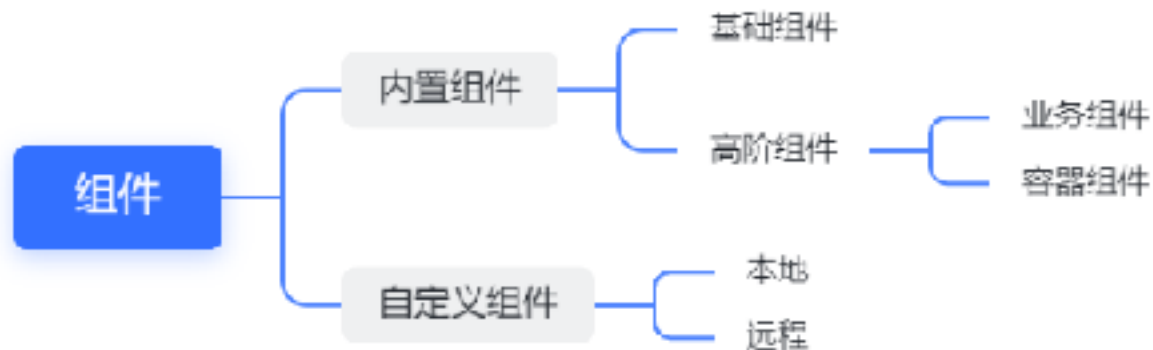
- Release Note
- Super Table 产品使用指南
- 快速开始
- 设计原理
- API 接口文档
- 行为文档
- 组件文档
- 设计思路
- 存在问题
- 辅助工具
- 常见问题

SuperTable 1.0



- 基础
 - 设计、交互 (用户界面)
 - 接口 (前后端交互)
- 核心
 - 模块化
 - 自定义行为
 - 组件系统
 - 数据源统一管理 (方便数据共享)
- 高级应用
 - 功能完善, 文档完善
 - 应用简单方便
 - 降低了部署的难度
 - 在线部署

方案设计与实施 - 组件规范



- 基础组件：与组件库中现有组件一一对应
- 业务组件：基于现有业务封装且高频使用的组件
- 容器组件：处理组件间行为交互、数据传输等内容
- 自定义组件：满足用户多元化需求

方案设计与实施 - 组件规范

自定义组件 - 本地加载示例



```
1 import example from '...'
2 const pageConfig = {
3   _tag: 'page',
4   body: [
5     // ... 省略部分代码
6     {
7       tag: 'custom',
8       remote: false,
9       component: example
10    },
11    // ... 省略部分代码
12  ]
13 }
```

```
1 <!-- example.vue -->
2 <template>
3   <div>自定义组件</div>
4 </template>
5
6 <!-- custom.vue -->
7 <template>
8   <component :is="component"></component>
9 </template>
10
11 <script>
12 export default {
13   props: { data: Object },
14   data: { component: 'div' },
15   created () {
16     if (this.data.remote) {
17       this.loadRemoteComponent()
18     } else {
19       this.component = this.data.component
20     }
21   },
22   methods: { loadRemoteComponent () { /* ... */ } }
23 }
24 </script>
```

方案设计与实施 - 组件规范

自定义组件 - 远程加载示例

_tag: page

tag: txt

Page 1 txt 1

tag: txt

Page 1 txt 2

_tag: custom

远程加载组件

_tag: page

```
1 ▾ const pageConfig = {
2   _tag: 'page',
3   body: [
4     // ... 省略部分代码
5     [
6       _tag: 'custom',
7       remote: true,
8       remoteUrl: 'http://.../myLib.umd.js'
9     ],
10    // ... 省略部分代码
11  ]
12 }
```

```
1  <!-- custom.vue -->
2  <template>
3    <component :is="component"></component>
4  </template>
5
6  <script>
7    import requirejs from '...'
8    export default {
9      props: { data: Object },
10     data: { component: 'div' },
11     created () {
12       if (this.data.remote) {
13         this.loadRemoteComponent()
14       } else {
15         this.component = this.data.component
16       }
17     },
18     methods: [
19       loadRemoteComponent () {
20         requirejs([this.data.remoteUrl], (component) => [
21           this.component = component
22         ])
23       }
24     ]
25   }
26 </script>
```

方案设计与实施 - 行为规范

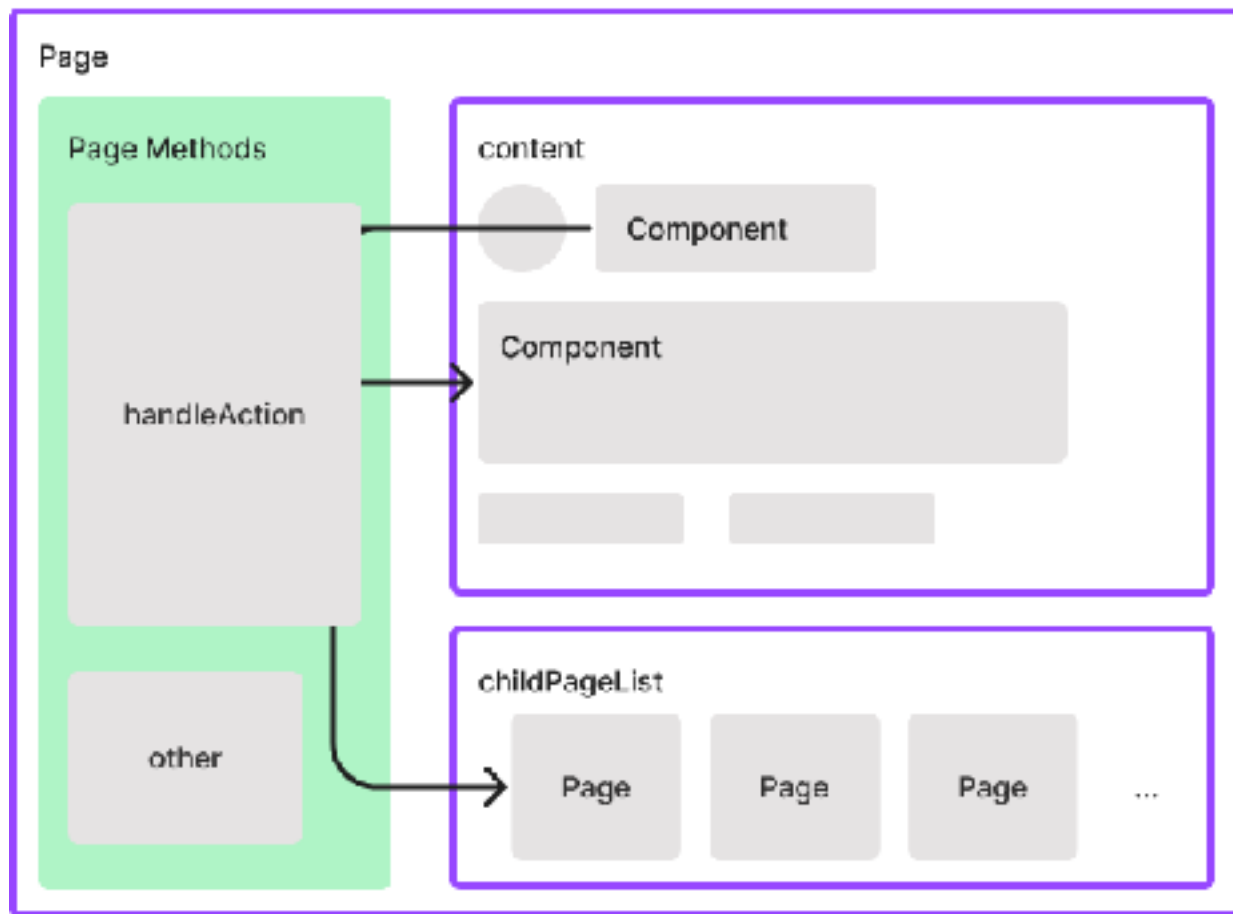
Page 为容器组件

1. 子组件触发的事件，将在 Page 组件中完成操作

2. 根据结果做出反馈

- 调用一个接口
- 打开一个页面
- ...

当然也支持链式调用



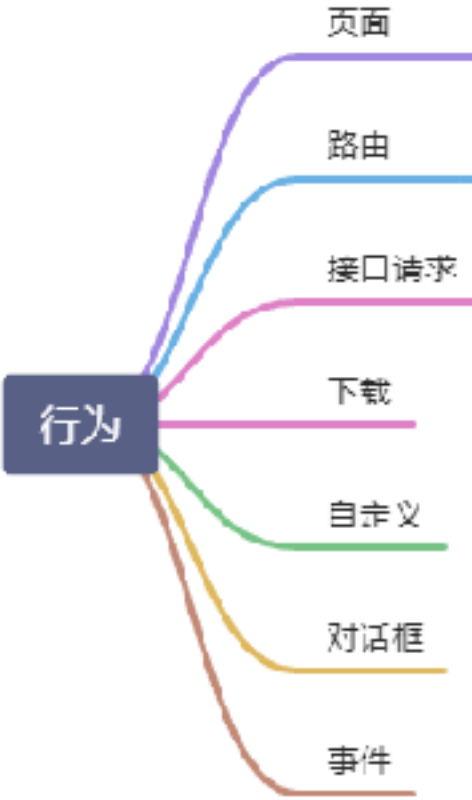
方案设计与实施 - 行为示例

场景：点击“删除”按钮，弹窗提示窗，用户确认后删除数据，同时刷新页面



```
{  
  "title": "删除",  
  "type": "error",  
  "action": [  
    { "type": "modal", "name": "confirm", "config": { "title": "确认删除?" } },  
    { "type": "ajax", "showMessage": "删除成功", "url": "/xxxxxx" },  
    { "type": "page", "name": "refresh" }  
  ]  
}
```

方案设计与实施 - 行为示例



分类	行为	类型 (type)	名称 (name)	其它参数
页面	重置页面数据	page	reload	
	刷新页面数据	page	refresh	
	页面展示	page	show	pageName
	关闭当前页面	page	close	
路由	打开指定路由 (打开新页面)	router	push	routeName extraData
	打开指定路由 (替换当前页)	router	replace	routeName extraData
	打开新窗口	router	open	url

方案设计与实施 - 成果演示

...

配套设施 - 示例站点

快捷切换开发环境地址

输入子域名

输入验证码

获取验证码

登录

首页

模板管理

示例 (11)

Form 表单

InfoCard 信息卡片

Page 页面

1. 简单页面

2. 子页面

3. 子页面操作按钮

Pageinfo 信息展示

Table 表格

Tab 标签页

输入验证码查看验证码

查看完整配置

快捷文档

历史用户数量

2708

今日新增用户数量

123

今日新增用户数量

422

活动名称	活动名称
活动名称078	活动名称078
活动名称042	活动名称042
活动名称339	活动名称339
活动名称419	活动名称419
活动名称009	活动名称009

共 166 条

<

1

>

3

...

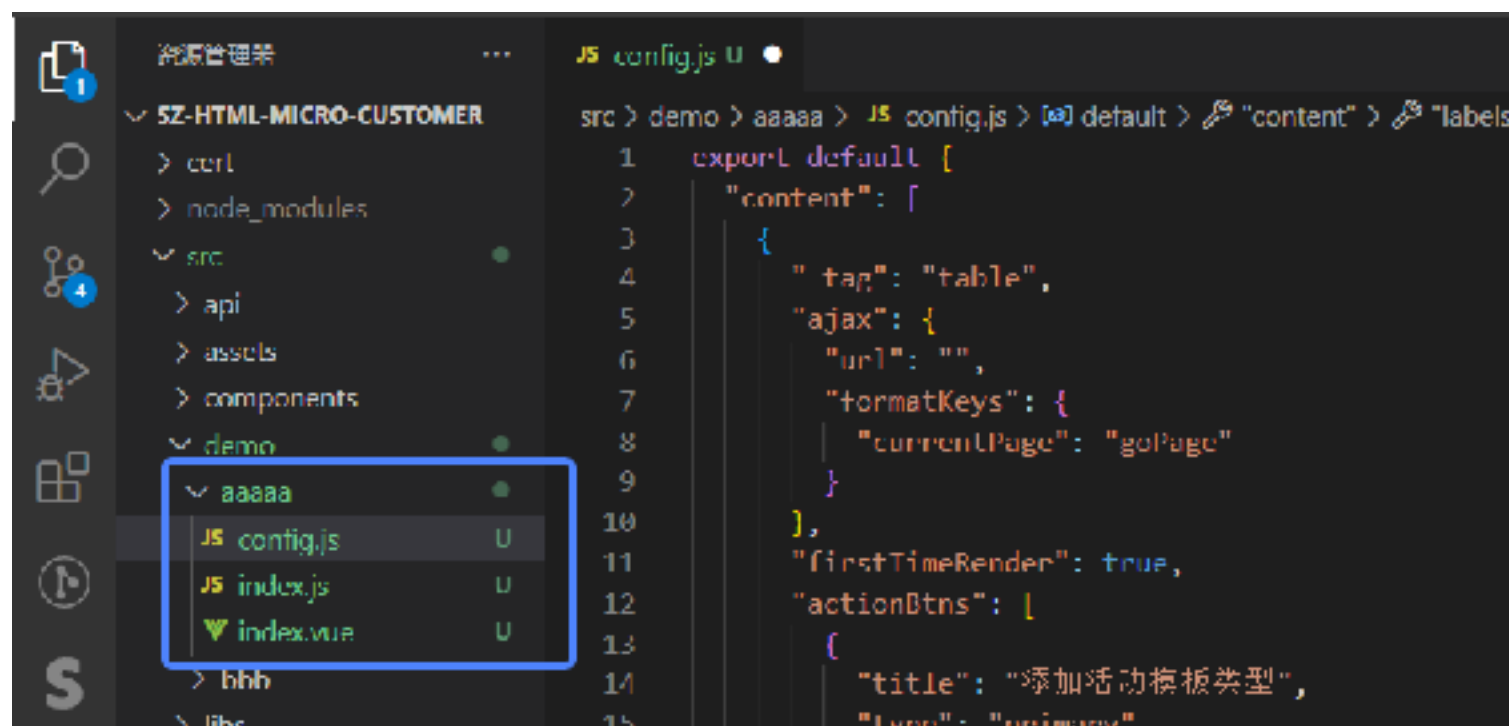
17

>

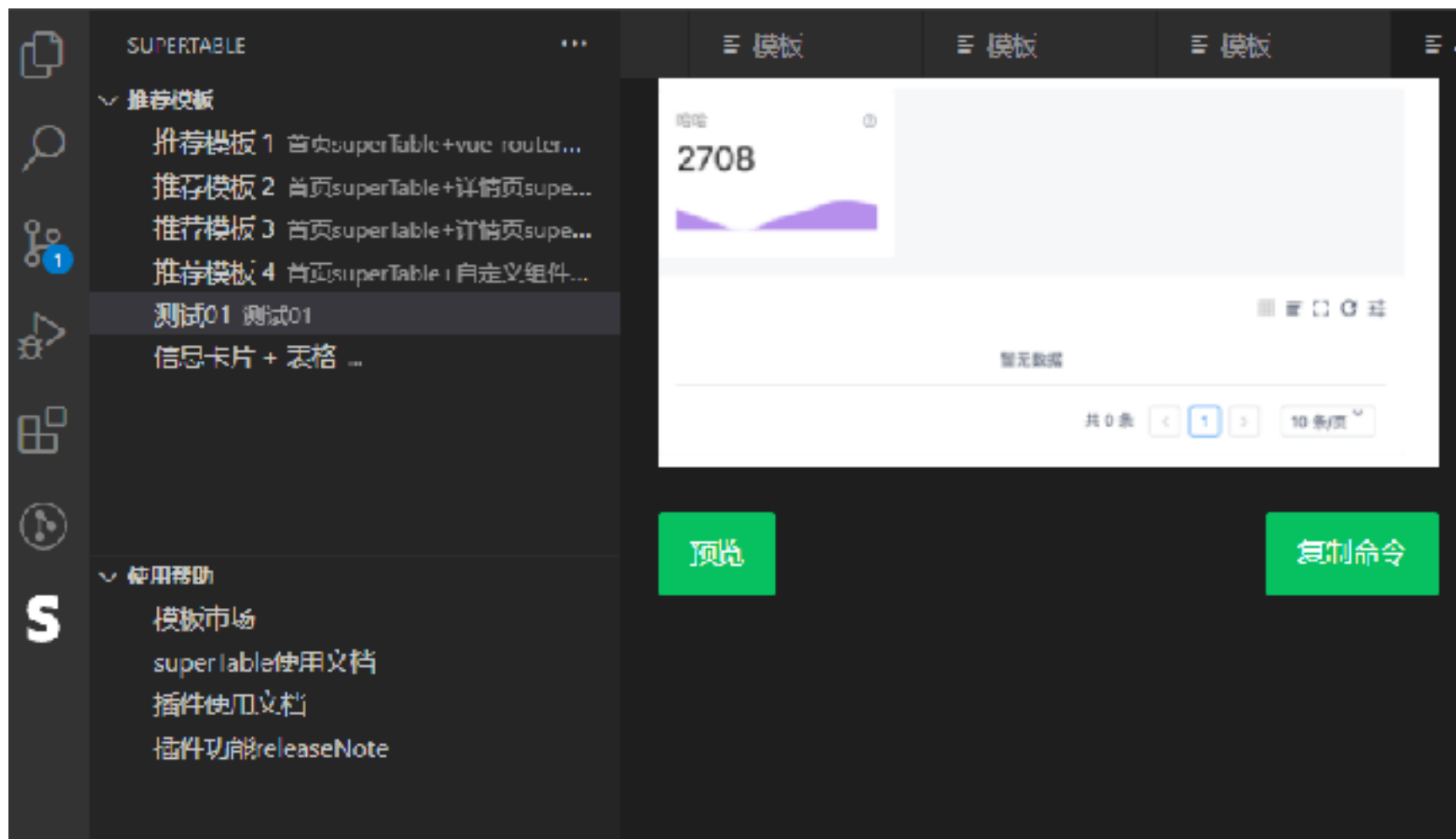
10 条/页

配套设施 - 脚手架

```
PS C:\workspace\sz-html-micro-customer\src\demo> szcli supertempl  
ate flIQqMQBsJ9TG1ZCqzc8w33  
id flIQqMQBsJ9TG1ZCqzc8w33  
? 请输入项目名: aaaaa  
模板文件aaaaa建立成功!
```



配套设施 - VS Code 插件



一些思考

—— PART 4 ——

感谢聆听

Q&A - 交个朋友



炫尘



内蒙古 包头



最好备注名字和公司，感谢

扫一扫上面的二维码图案，加我微信



一起走满5年

早早聊天使票预售

扫描右侧二维码

抢前 1000 分红名额：



早

虎年跳个好团队

早早聊 跳槽营



职业路线辅导
跳槽套路指导
模拟面试摸底
题库讲解直播
面经直播分享
优质团队内推
优质简历分享
职业成长视频

八大服务 破解 七大跳槽难题

学历硬伤
跳槽高频
项目简单
底子薄弱
深度不足
业务不精
管理不通



跟我一起学习吧!

每月挑一个周六，一起充电成长!



快扫码上车早早聊
400个录播等你看

2020 全年行程 ▶

1/11 前端转管理
2/29 前端搞基建
3/28 前端搞基建
4/11 前端搞规划
4/25 前端搞验收
5/16 Serverless
5/30 玩转微前端
5/31 前端的方方面面
6/13 在线文档专场
6/20 前端的跨端栈
6/27 女生职业专场
7/18 前端搞可视化
8/15 前端搞构建
8/29 职业成长与晋升
9/26 前端搞报表
10/17 组件化资产
11/21 前端搞框架专场
12/5 脚手架工具链
12/26 前端的性能专场

2021 全年行程 ▶

1/9 前端自由职业
1/23 前端团队管理
2/6 小程序组件化
2/27 页面搭建专场
3/20 大厂招聘面试
4/10 前端的CI/CD
4/24 前端的转行方法
5/9 前端的进阶提升
5/15 前端的玩转互动
6/5 跨端搞Flutter
6/12 前端搞动画专场
6/26 前端搞WebGL
7/17 前端搞可视化
8/14 前端的玩转BFF
8/21 GraphQL专场
8/28 前端的框架专场
9/11 前端的网络安全专场
10/23 前端的Vue
11/20 前端的IDE

2022 全年行程 ▶

1/29 前端的职业进阶
2/26 TS技术专场
3/19 管理专场
4/9 跳槽面试专场
4/23 组件专场
5/14 监控专场
5/28 性能专场
6/18 成长与晋升
6/25 跨端专场
7/16 海外工作专场
7/17 一站式基建
8/6 工程化专场
8/27 低代码专场
9/3 低代码微前端
9/17 音视频专场
9/24 创业专场
10月 开发语言专场
10月 保持健康专场
11月 自由职业专场
11月 框架专场
12月 运维专场
12月 鸿蒙专场
12月 可视化专场