

# 问卷管理系统

代码的复杂性不在于其本身，而在于应对多样化的业务场景，系统需要更深入的设计和沉淀。作为专家工程师，不仅要具备架构能力，还需深入理解业务逻辑。我们不能单纯地以工具应对问题，而是要提升全局视角，从行业和场景的角度结合技术深度来架构系统。

本章将全面探讨企业级问卷系统的构建，涵盖从业务思考到技术架构的全套实践方案，并分析智能化场景的思路。

环境：Node 版本 > 18.x

技术栈：Node.js + Nest.js + MongoDB + Vue3 + Docker

## 业务模型

问卷系统分为问卷管理端和问卷投放端。

问卷管理端：面向问卷管理者，提供问卷创建、设计、多人协作及数据分析功能。管理端支持定制问卷皮肤、设置逻辑规则和响应机制，生成分发链接或二维码，并通过短信、公众号或邮件分发问卷，实时监控填写进度和响应情况。

问卷投放端（答题端）：面向受访者，提供直观且友好的填写体验。投放端界面简洁，支持多种输入方式如选择题、填空题等，兼容不同设备和浏览器，确保填写便捷。系统自动保存填写进度，支持断点续填，提高用户填写的连续性和准确性。

实体名称	描述	示例
用户	用户账号的一种类型，注册成功即为个人用户，需通过认证或付费后才能升级为企业账号	个人用户
问卷空间	承载问卷的实体，每一份问卷都是在一个空间中。空间与空间之间有层级关系	个人空间
问卷	问卷平台最重要的实体，包含众多属性，且面向不同用户，不同端都具备不一样的形态	问卷管理端
题目	问卷中包含的实体，包含众多属性，且因为题型分类较多，属性之间既有通用属性，也有各自题型适用的特殊属性	单选题
子标题	题目中包含的实体，通常矩阵类题型和组合类题型才有	
选项	题目中包含的实体，所有题目都具有选项，可能为系统提供的值，也可能支持用户自定义输入任意值	
答卷	问卷在经过问卷投放端用户填写后提交上来的实体，不可变更	

# 产品架构

在调研领域，「创、投、收、分析、报告」是问卷系统核心的节点：

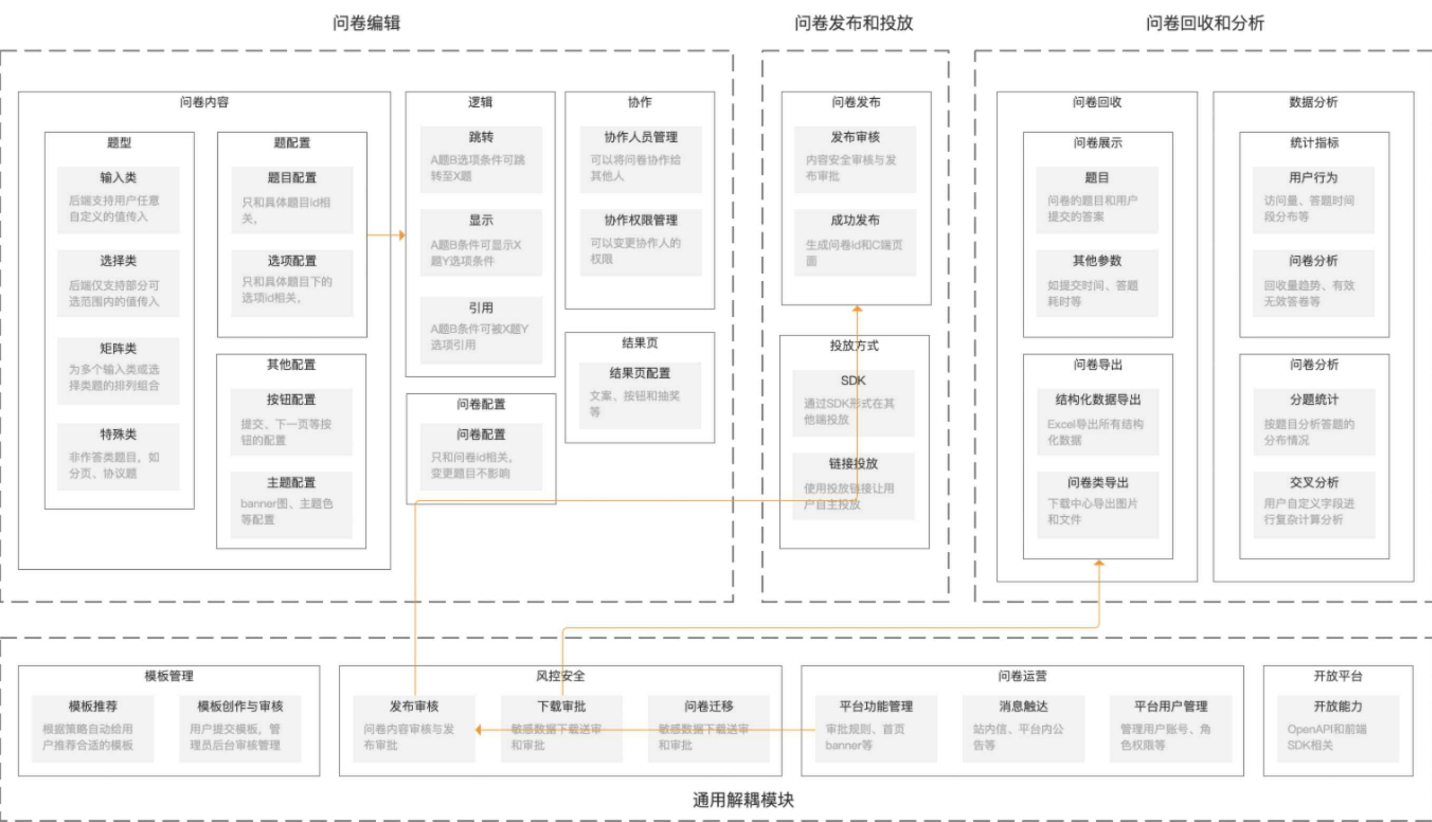
「创」阶段是问卷设计和创建的过程，根据调研需求设计问卷。

「投」阶段涉及到问卷的分发，通过多种渠道有效触达目标受众。

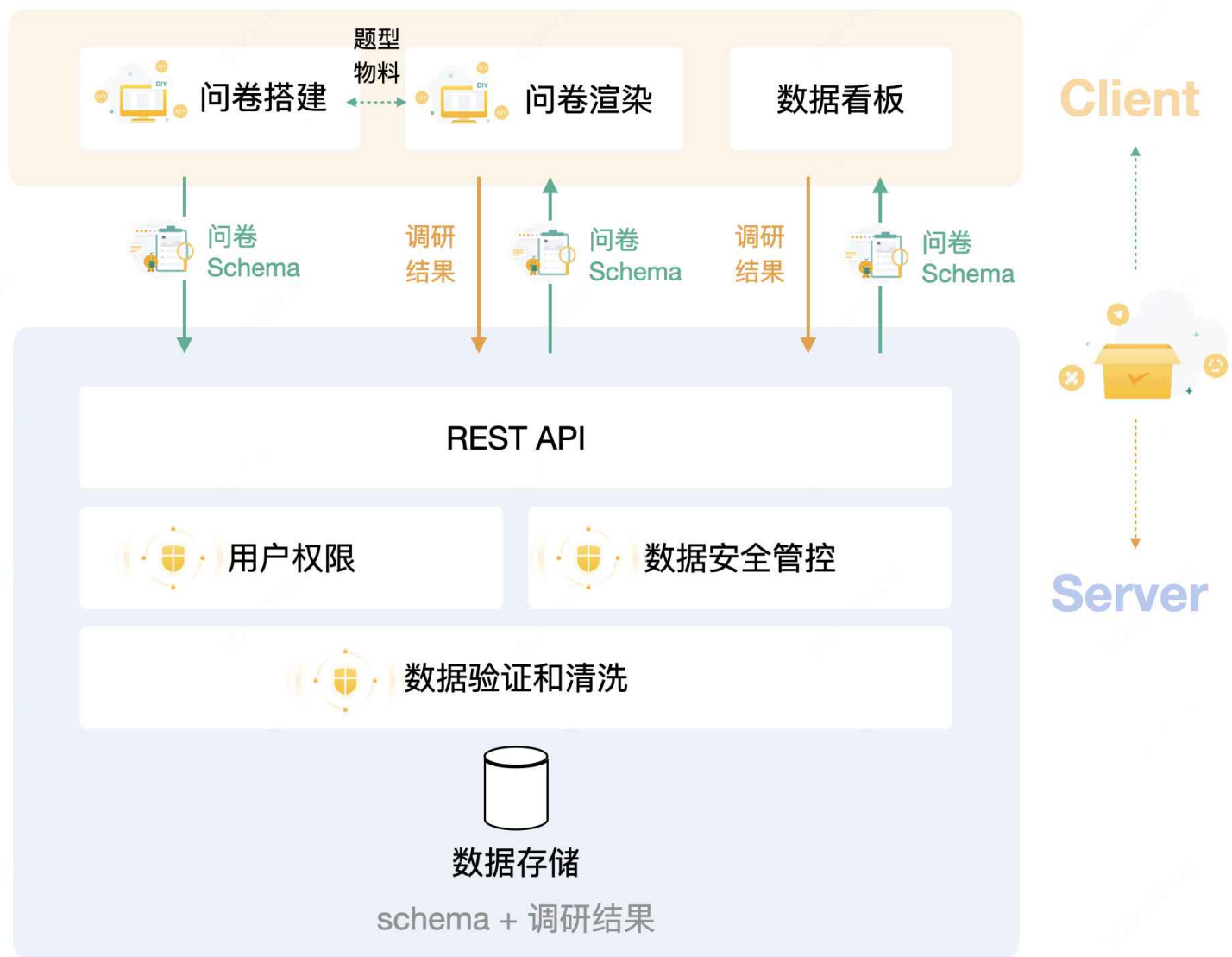
「收」阶段是问卷收集的过程，发放的问卷被目标受众填写并收回。

「分析」阶段是对收集到的数据进行深入分析，从中提取出有效的信息。

「报告」阶段则是将分析结果整理成报告，提供有效的建议和指导方案。



# 技术架构

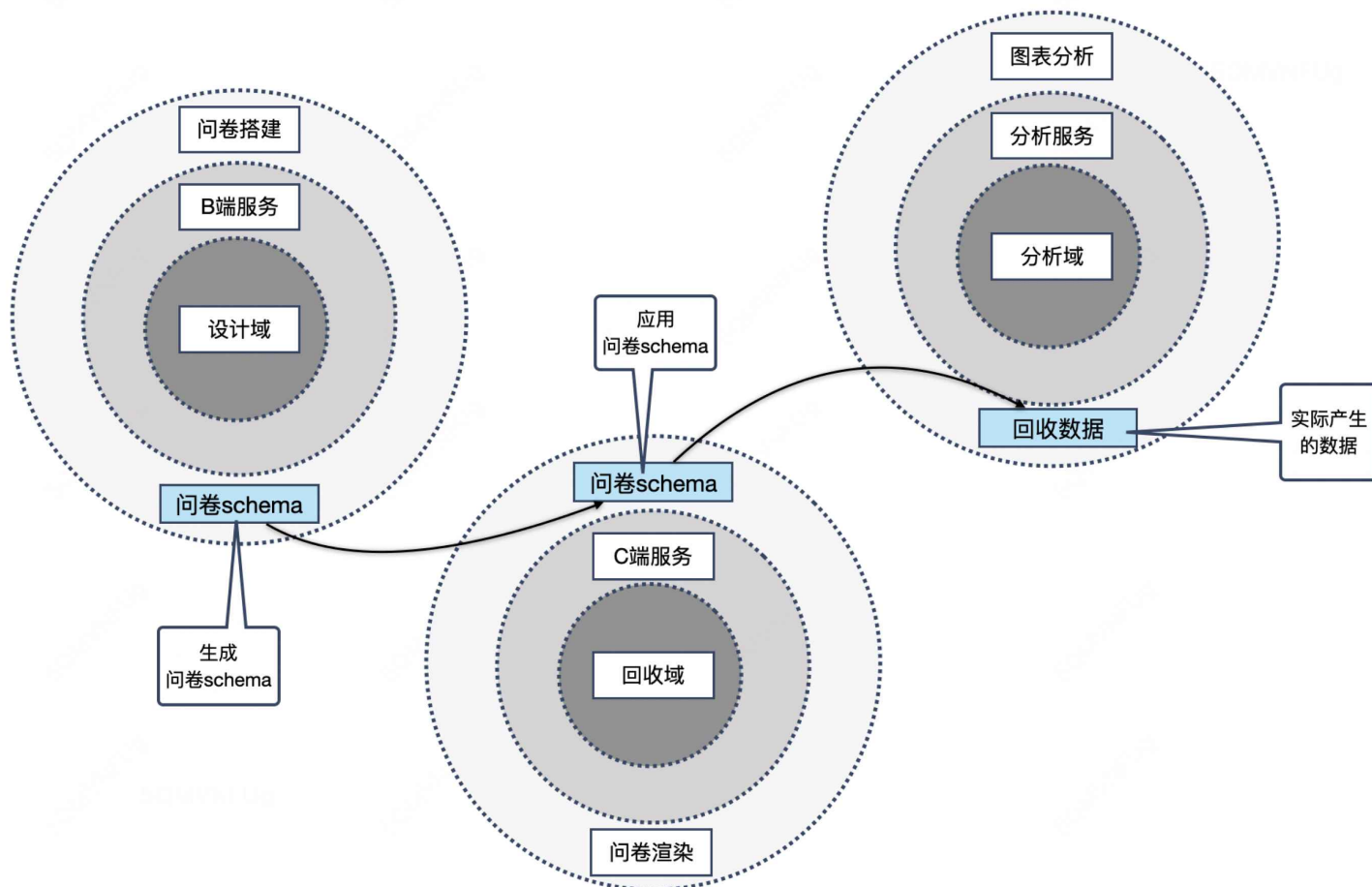


理解了架构全景后，可以进一步深入技术原理。

## 领域模型

问卷的本质是收集数据与洞察数据，数据贯穿于问卷的全生命周期。从创建开始的数据定义，到回收时的数据加工，再到分析的数据处理，以及数据血缘关系的追踪，在这个过程中，数据不断地发生变化。通过 Schema 来表达领域知识和数据定义，以及数据之间的关联，使得问卷能够支持多样化的应用场景。

管理端问卷设计实际是在设置问卷 schema 的值，投放端基于 schema 生成问卷内容，进而产生回收数据。



## 领域协议

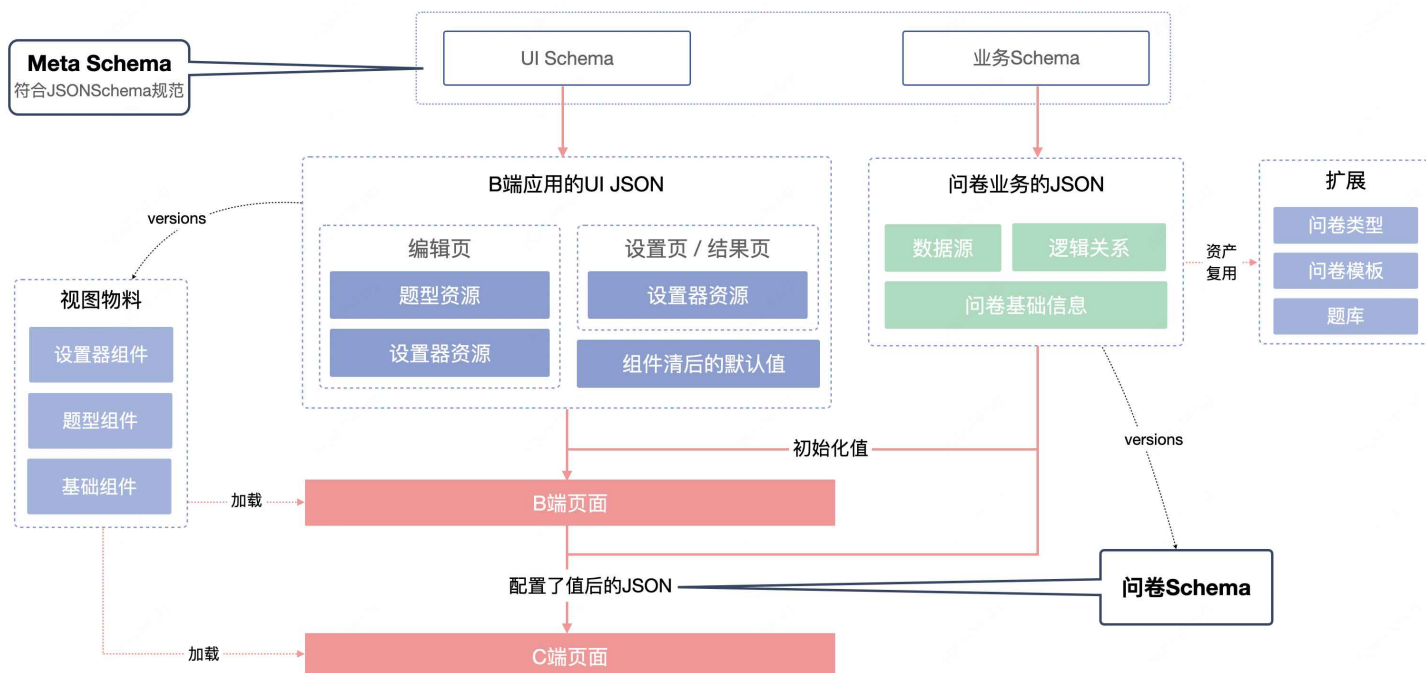
题型是问卷的核心基础元素，随着业务场景的复杂度提升，除了数据的管理，我们还需要对题型组件进行协议约定以便更好地复用和管理。所以我们把领域协议拆分成两部分，问卷业务协议 + 物料协议：

### 1、问卷业务协议，即 问卷 Schema

业务描述：问卷协议、题型协议。

### 2、物料协议，即 UI Schema

物料描述：题型物料协议。



问卷 Schema 可以分为两部分：问卷基础配置如有效期、主题配置等、不同题型有不同的配置。

### 1、基础信息 Schema 结构

问卷基础信息的 Schema 模板维护在服务端：

server/src/modules/survey/template/surveyTemplate/templateBase.json

```
{
  "bannerConf": { ...
  },
  "submitConf": {
    "submitTitle": "提交",
    "confirmAgain": {
      "is_again": true,
      "again_text": "确认要提交吗? "
    },
  },
  "msgContent": { ...
  },
},
"bottomConf": { ...
},
"baseConf": {
  "begTime": "2024-01-01 00:00:00",
  "endTime": "2034-01-01 00:00:00",
  "tLimit": 0,
  "language": "chinese",
  "answerBegTime": "00:00:00",
  "answerEndTime": "23:59:59"
},
"skinConf": { ...
},
"logicConf": { ...
}
}
```

**问卷基础配置**

**主题配置**

## 2、题型信息 Schema 结构

前端物料库维护题型字段，不同的题型有不同的配置文件 `meta.js`。

如 `web/src/materials/questions/widgets/InputModule/meta.js`

## 物料体系

为了保障预览实时性和渲染一致性，提升题型的多状态管理能力，并降低题型设计的复杂性和成本、提升题型扩展的灵活性，采用题型物料化的设计思路：

### 1、基础物料

将多题型的通用能力抽象为可复用的组件，如：输入类、选择类、评分类等。

### 2、题型物料

基于基础物料增加每个题型单独的配置和行为，扩展出包含业务特性的题型，如：单选题、评分题等。

## 源码分析

## 工程结构

根路径下，有 `server` 和 `web` 两个文件夹，分别是后端和前端的项目（以下是核心部分，完整内容查看工程源码）。

### server

服务端核心子模块：

#### survey

- 用于提供问卷管理端的问卷管理功能，为编辑问卷，投放问卷，问卷回收查看等提供接口服务。

#### surveyResponse

- 用于提供问卷投放端的问卷填写功能，主要功能为问卷投放端用户填写问卷提供接口服务。

#### auth

- 用户的功能服务，主要为用户相关操作接口，为登陆，注册等用户相关操作提供接口服务。

`survey`、`surveyResponse`、`auth` 都是 `api` 服务，主要用于提供服务接口。

```
1 .
2 |— README.md # 说明
3 |— build # 构建产物
4 |— package.json # 包说明
5 |— src # 源码目录
6 |   |— index.ts # 服务入口文件
```

```

7 |   └─ modules
8 |       └─ survey # 问卷管理接口层
9 |       └─ surveyResponse # 问卷发布接口层
10 |   └─ auth # 登陆注册模块
11 | └─ tsconfig.json

```

## web

前端核心包含物料库(materials)、问卷管理 - 问卷管理端(management)、问卷投放 - 问卷投放端(render):

```

1 | .
2 | └─ README.md # 说明
3 | └─ build # 构建产物
4 | └─ package.json # 包说明
5 | └─ vite.config.js
6 | └─ public # 静态资源文件
7 | └─ src # 源码目录
8 |   └─ management # 问卷管理端工程
9 |       └─ main.js # 问卷管理端入口文件
10 |      └─ App.vue # 问卷管理端入口vue文件
11 |      └─ api # 接口请求文件夹
12 |      └─ components # 组件
13 |          └─ empty.vue # 通用的空页面
14 |          └─ leftMenu.vue # 编辑页、投放页、数据回收页通用的菜单
15 |          └─ logo.vue # logo组件
16 |      └─ config # 配置
17 |          └─ questionConfig.js # 题目的基础字段配置
18 |          └─ questionMenuConfig.js # 控制编辑页左侧的题型菜单的题型显示顺序
19 |          └─ setterConfig # 内置组件设置器的配置
20 |              └─ bannerConfig.js # banner的设置器配置
21 |              └─ logoConfig.js # logo的设置器配置
22 |              └─ submitConfig.js # 提交按钮的设置器配置
23 |      └─ pages # 问卷管理端页面
24 |          └─ analysis # 回收数据页
25 |          └─ create # 创建页
26 |          └─ list # 列表页
27 |          └─ login # 登陆页
28 |          └─ publishResult # 发布结果页
29 |          └─ edit # 编辑页
30 |              └─ index.vue # 编辑页入口
31 |              └─ pages # 编辑页子页面
32 |              └─ components # 编辑页组件
33 |              └─ modules # 微模块文件夹
34 |                  └─ contentModule # 内容管理微模块

```



```
35 | | | generalModule # 通用功能微模块
36 | | | permissionModule # 权限微模块
37 | | | questionModule # 题内容微模块
38 | | | logicModule # 题状态微模块
39 | | | settingModule # 设置微模块
40 | | | index.html # 问卷投放端html模板
41 | | | router # 问卷管理端路由配置
42 | | | store # 问卷管理端store
43 | | | styles # 样式
```

## 服务启动

启动服务时使用以下命令：

```
1 cd server
2 npm run install
3 npm run local
4 cd web
5 npm run install
6 npm run dev
```

当看到以下页面时，说明启动完成：

## MongoDB

数据库使用 [mongodb-memory-server](#) 这个库：

- 1、数据保存在内存中，重启服务会更新数据。
- 2、启动内存服务器新实例时，如果找不到 MongoDB 二进制文件会自动下载，因此首次可能需要一些时间。

## 进阶

可以自行配置数据库，具体方式可以：[查看文档](#)。

## Docker部署

基于 Dockerfile 构建一个名为 xiaoju-survey-app 的 Docker 镜像。

```
1 cd xiaoju-survey
2 docker build -t xiaoju-survey-app .
```



镜像构建完成后，可以运行 Docker 来启动应用。

```
docker run --name running-xiaoju-survey-app -p 8080:80 xiaoju-survey-app
```

## 练习

学习完原理后，可以启动源码进行进一步学习。