

Web开发专题（综合练习）外贸发货看板开发实验讲义

一、实验目标

- 掌握 RooCode 四个角色的协作开发流程 @<https://docs.roocode.com/basic-usage/using-modes>
- 学会使用 Jekyll 构建数据驱动的静态网站 @<https://jekyllrb.com/docs/step-by-step/01-setup/>
- 实现外贸运单数据的动态展示与管理

二、实验环境准备

```
# 安装 Jekyll 环境
gem install bundler jekyll
# 创建项目目录结构
jekyll new shipment-dashboard
cd shipment-dashboard
```

三、核心知识点

1. RooCode 角色分工

角色	职责	本实验对应任务
战略协调者	任务拆解与资源分配	制定开发阶段计划
技术领导者	系统设计与技术方案制定	设计数据结构和页面模板
技术助手	解决技术细节问题	解答 Jekyll/Liquid 相关问题
软件工程师	代码实现与调试	编写数据文件和模板代码

2. Jekyll 数据驱动开发

- `_data` 目录支持 YAML/JSON/CSV 格式
- 数据访问语法: `site.data.filename`
- Liquid 模板控制流: `for / if / case`

四、实验步骤详解

步骤 1: 需求拆解（战略协调者）

要求： 将整体需求拆解为可执行子任务

实现方法：

1. 创建任务清单：

- [] 设计数据文件结构
- [] 创建客户数据文件
- [] 创建运单数据文件
- [] 开发运单列表页面
- [] 开发运单详情页面
- [] 实现状态显示逻辑

步骤 2：数据结构设计（技术领导者）

要求： 设计符合需求的 YAML 数据结构

实现方法：

```
# _data/clients.yml 示例
- code: NTS-2304
  name: Ruixin
  timezone_offset: +3
  language: ar
  country: SA
  contact: "+966 551234567"
```

```
# _data/shipments.yml 示例
- tracking_number: NTS-2304_ELECT_50_20250507T093000
  client_code: NTS-2304
  transport_mode: Sea
  status: 已发货
  created_at: 2025-05-07T09:30:00
  shipped_date: 2025-05-08
```

步骤 3：页面模板开发（软件工程师）

要求： 实现运单列表展示功能

实现方法：

```
<!-- shipments.html -->
---
layout: default
---
<ul>
  {% for shipment in site.data.shipments %}
  <li>
```

```
<h3>{{ shipment.tracking_number }}</h3>
<p>状态:
  {% case shipment.status %}
    {% when '未发货' %} 🟡 等待发货
    {% when '已发货' %} 🚢 运输中
    {% when '已到达' %} ✅ 抵达目的港
  {% endcase %}
</p>
</li>
{% endfor %}
</ul>
```

步骤 4：数据关联查询（技术助手）

要求：实现运单与客户信息关联显示

实现方法：

```
{% assign client = site.data.clients | where: "code", shipment.client_code | first %}
<p>客户时区: UTC{{ client.timezone_offset }}</p>
```

步骤 5：日期格式处理（软件工程师）

要求：显示友好日期格式

实现方法：

```
{% assign shipped_date = shipment.shipped_date | date: "%Y年%m月%d日" %}
<p>发货日期: {{ shipped_date }}</p>
```

步骤 6：状态过滤功能（技术领导者）

要求：实现按状态筛选运单

实现方法：

```
{% assign pending_shipments = site.data.shipments | where: "status", "未发货" %}
<h4>待处理运单 ({{ pending_shipments.size }})</h4>
```

五、实验效果验证

1. 启动本地服务器

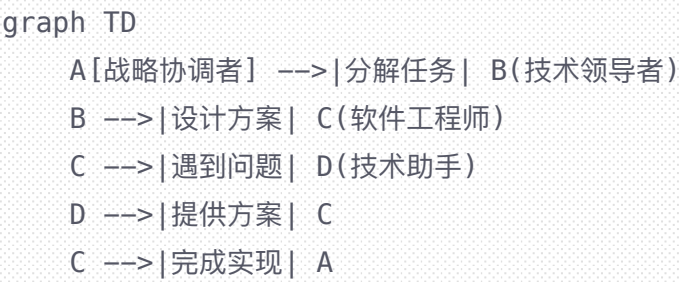
```
bundle exec jekyll serve
```

- 2. 访问 `http://localhost:4000/shipments.html`
- 3. 修改 `_data/shipments.yml` 后刷新页面观察变化

六、扩展思考

- 1. 如何添加自动计算预计到达时间？
- 2. 怎样实现不同时区的时间转换？
- 3. 如何通过 GitHub Actions 实现自动部署？

七、RooCode 角色协作流程图



通过本实验，学生将掌握多角色协作开发模式，并学会使用 Jekyll 构建数据驱动型网站的核心技能。