

JY1.0 立库详细界面功能说明书

1. 文档目的

本文档用于整理 JY1.0 项目中“立库查询/立库可视化”组件目前已经实现的界面结构、后端数据来源、交互逻辑和现有功能范围，便于后续培训、验收、维护和功能扩展。

本文档依据当前前端实现整理，核心代码位置如下：

- src/views/WarehouseManagement/WarehouseVisualization/index.vue
- static/config.js

2. 页面定位

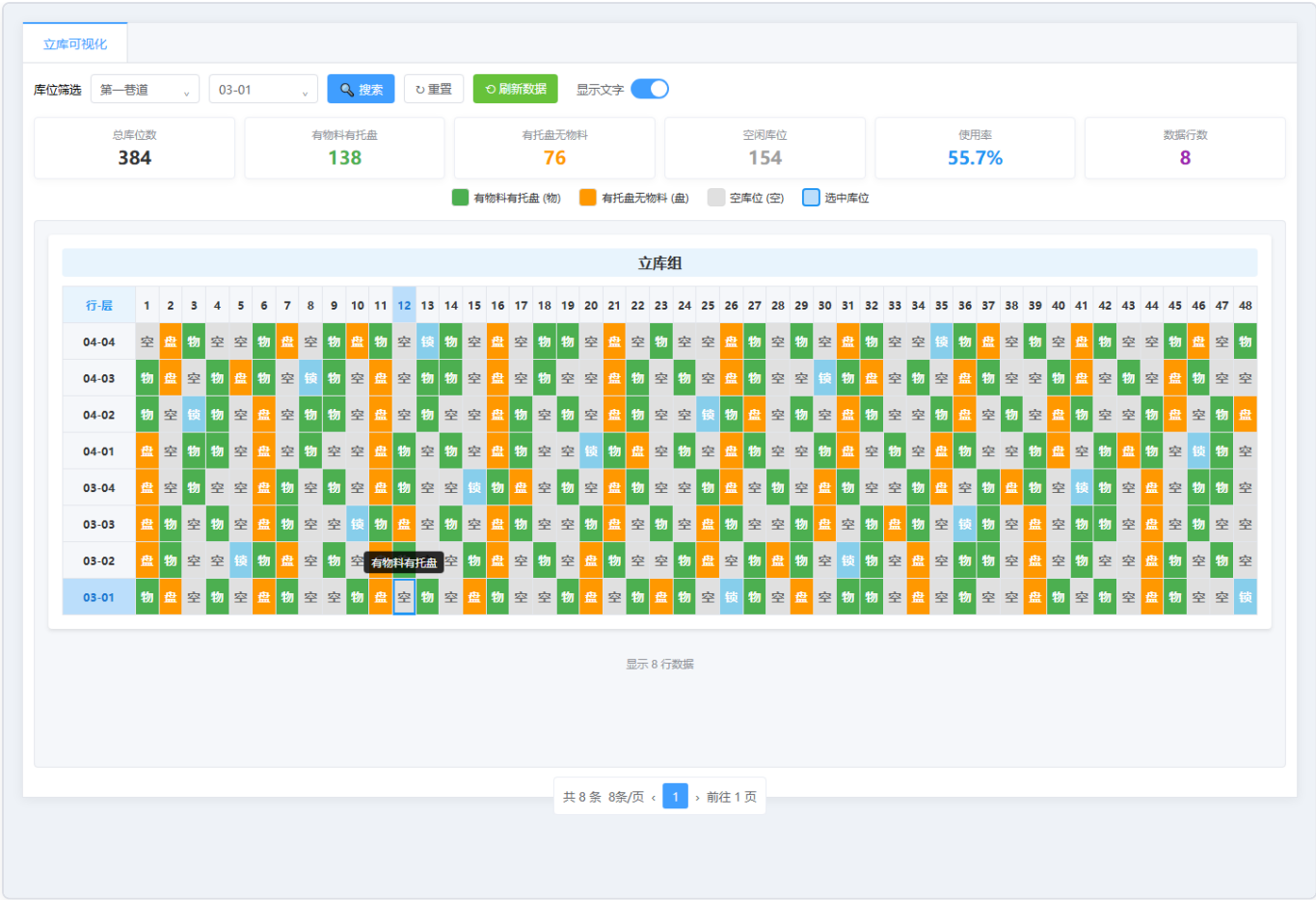
立库详细界面当前对应的页面名称为：

- 页面组件名：WarehouseVisualizationSystem
- 页面标签名：立库可视化
- 目录位置：src/views/WarehouseManagement/WarehouseVisualization/index.vue

该页面用于展示立体仓库库位占用情况，并支持按巷道查看、按行层筛选、单击库位查看库位详情或物料明细。

3. 页面总体结构

图 1：立库可视化主界面



页面整体采用标准的 `div.app-container + el-card` 布局，主体包含以下区域：

1. 顶部筛选与操作区
2. 统计信息卡片区
3. 状态图例区
4. 立库可视化表格区
5. 分页区
6. 库位详情弹窗
7. 物料信息弹窗

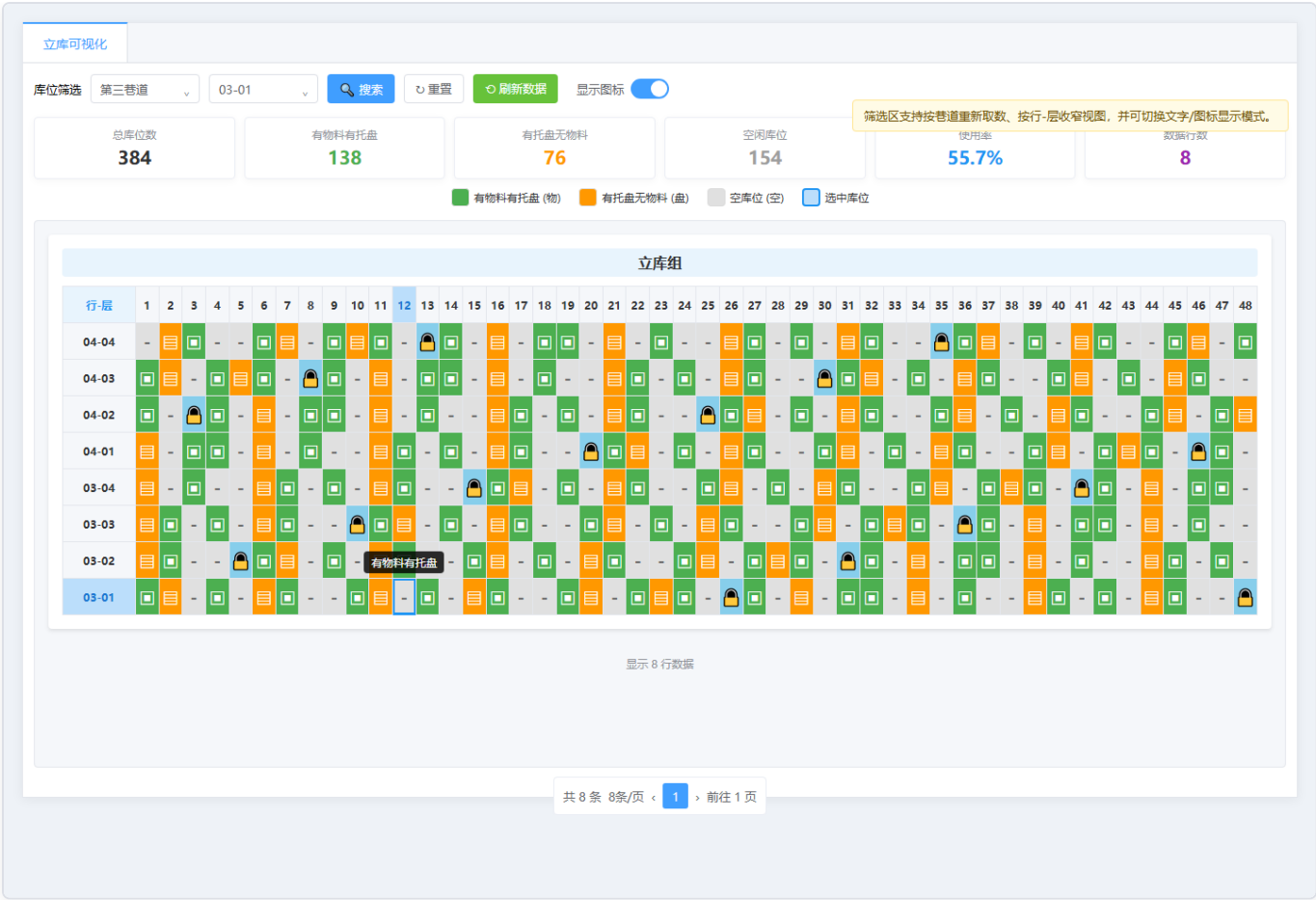
当前页面只有一个主标签页：

- 立库可视化

代码中保留了图表分析相关逻辑，但当前模板中未实际启用“分析”页签。

4. 顶部筛选与操作区说明

图 2：筛选区与显示模式切换示意



顶部区域已经实现以下控件：

4.1 巷道筛选

控件类型： el-select

可选项固定为：

- 第一巷道
- 第二巷道
- 第三巷道
- 第四巷道

对应值为：

- 1
- 2
- 3
- 4

用途：

- 用户选择巷道后，点击“搜索”，页面会调用立库接口重新获取该巷道数据。

4.2 行层筛选

控件类型： el-select

用途：

- 根据当前已加载的立库数据，自动生成“行-层”下拉选项。
- 用户可筛选某一行层，只显示对应行数据。

特点：

- 选项从接口返回数据动态提取。
- 会自动去重。
- 按数值倒序排列。

4.3 搜索按钮

按钮文字：搜索

作用：

- 重置当前页码为第一页。
- 先执行前端筛选。
- 再根据当前巷道值调用 `GetPosition` 获取最新立库数据。

说明：

- 当前“搜索”动作的核心是按巷道重新取数。
- 代码里存在 `searchKeyword` 变量，但页面上没有对应输入框，因此当前没有“按列关键字搜索”的实际入口。

4.4 重置按钮

按钮文字：重置

作用：

- 清空状态筛选值
- 清空行层筛选值
- 清空搜索关键字
- 页码回到第一页
- 重新执行前端筛选

页面提示：

- 执行后弹出“筛选条件已重置”。

注意：

- 当前重置不会主动重新按巷道取数，只是重置前端筛选状态。

4.5 刷新数据按钮

按钮文字：刷新数据

作用：

- 触发加载动画

- 500ms 后重新装载当前 `defaultWarehouseData`
- 更新时间戳
- 弹出 “数据刷新成功”

说明：

- 刷新逻辑最终仍依赖最近一次 `GetPosition` 返回并写入的 `defaultWarehouseData` 。
- 如果前面已经切换过巷道，刷新展示的是当前缓存的该次结果。

4.6 显示模式切换开关

控件类型： `el-switch`

两种模式：

- 显示文字： `text`
- 显示图标： `icon`

作用：

- 控制库位单元格显示简称文字还是图标。

切换效果：

- 文字模式显示： `物 / 盘 / 空 / 锁`
- 图标模式显示：

- 锁库： `el-icon-lock` - 有物料有托盘： `el-icon-box` - 有托盘无物料： `el-icon-tickets` - 空库位： `el-icon-minus`

页面提示：

- 切换后会提示当前已切换到文字模式或图标模式。

5. 统计信息卡片区说明

页面中部已实现 6 个统计卡片：

5.1 总库位数

计算方式：

- `立库行数 * 列数`

5.2 有物料有托盘

统计状态值为 `2` 的库位数量。

5.3 有托盘无物料

统计状态值为 `1` 的库位数量。

5.4 空闲库位

统计状态值为 `0` 的库位数量。

5.5 使用率

计算方式：

- $(\text{有物料有托盘} + \text{有托盘无物料}) / \text{总库位数} * 100\%$

保留 1 位小数。

5.6 数据行数

显示当前 `warehouseData.length`，即当前已加载的立库行数。

6. 状态图例说明

页面已实现状态图例，用于帮助用户理解库位颜色含义：

- 绿色：有物料有托盘
- 橙色：有托盘无物料
- 灰色：空库位
- 蓝色边框：当前选中库位

另外，代码中还支持：

- 蓝色底：锁库（状态值 9）

该状态在单元格式度和状态文字中已实现，但图例区未单独展示“锁库”说明。

7. 立库可视化表格区说明

这是页面的核心区域。

7.1 展示形式

页面使用原生 `table` 绘制立库矩阵，不是 `el-table`。

结构包含：

- 表头列号
- 左侧行层标识
- 中间库位矩阵单元格

7.2 行列组织方式

行

每一行对应一个“行-层”值，例如：

- 01-01
- 03-12

页面左侧固定显示该行层编号。

列

每列按照数字列号展示，从 1 到 columnCount 。

columnCount 的计算规则为：

- 读取首行对象字段
- 排除 行-层
- 其余字段个数即为列数

7.3 单元格状态值

当前库位单元格支持以下状态：

状态值	含义	前端显示简称	颜色
0	空库位	空	灰色
1	有托盘无物料	盘	橙色
2	有物料有托盘	物	绿色
9	锁库/在存取	锁	蓝色

7.4 单元格交互效果

当前已实现以下交互：

- 鼠标悬停单元格时放大
- 鼠标悬停显示状态浮层
- 点击后高亮选中库位
- 点击后高亮整行对应的行层
- 点击某列后，对应列表头也会高亮

7.5 空数据处理

当 warehouseData.length === 0 时，页面显示空白占位区域。

说明：

- 代码里曾预留 el-empty 结构，但目前已注释，没有实际 “暂无数据” 文案和 “加载示例数据” 按钮。

8. 分页功能说明

页面底部已实现分页控件，条件为：

- totalPages > 1 时显示

分页能力包括：

- 查看总数
- 切换每页条数

- 上一页/下一页
- 页码跳转
- 直接输入页码跳转

每页可选条数：

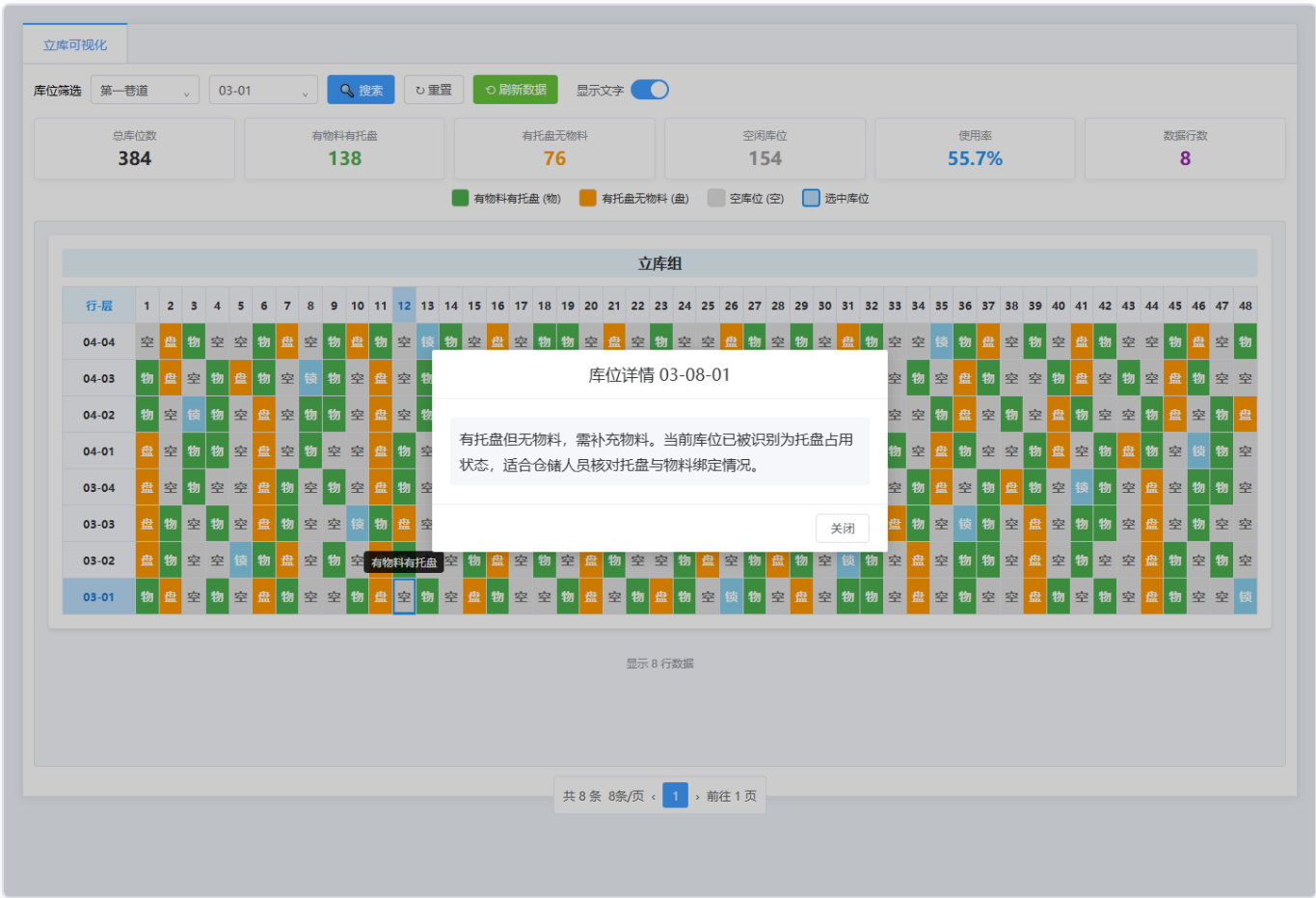
- 5
- 8
- 10
- 15

说明：

- 分页针对的是“行数据”，不是单元格。
- 也就是一次显示若干个“行-层”。

9. 库位详情弹窗说明

图 3：库位详情弹窗示意



点击非物料状态库位时，会弹出“库位详情”窗口。

触发条件：

- 单元格状态不等于 2

9.1 弹窗标题

格式：

- 库位详情 + 库位编号

例如：

- 库位详情 03-12-01

9.2 弹窗内容

当前弹窗内容为库位状态说明文本，即 `selectedCellDetail.description` 。

不同状态说明如下：

- 0：空闲库位，可存放物料
- 1：有托盘但无物料，需补充物料
- 2：有托盘且有物料，正常使用中
- 9：在存取，库位已锁

9.3 当前实现特点

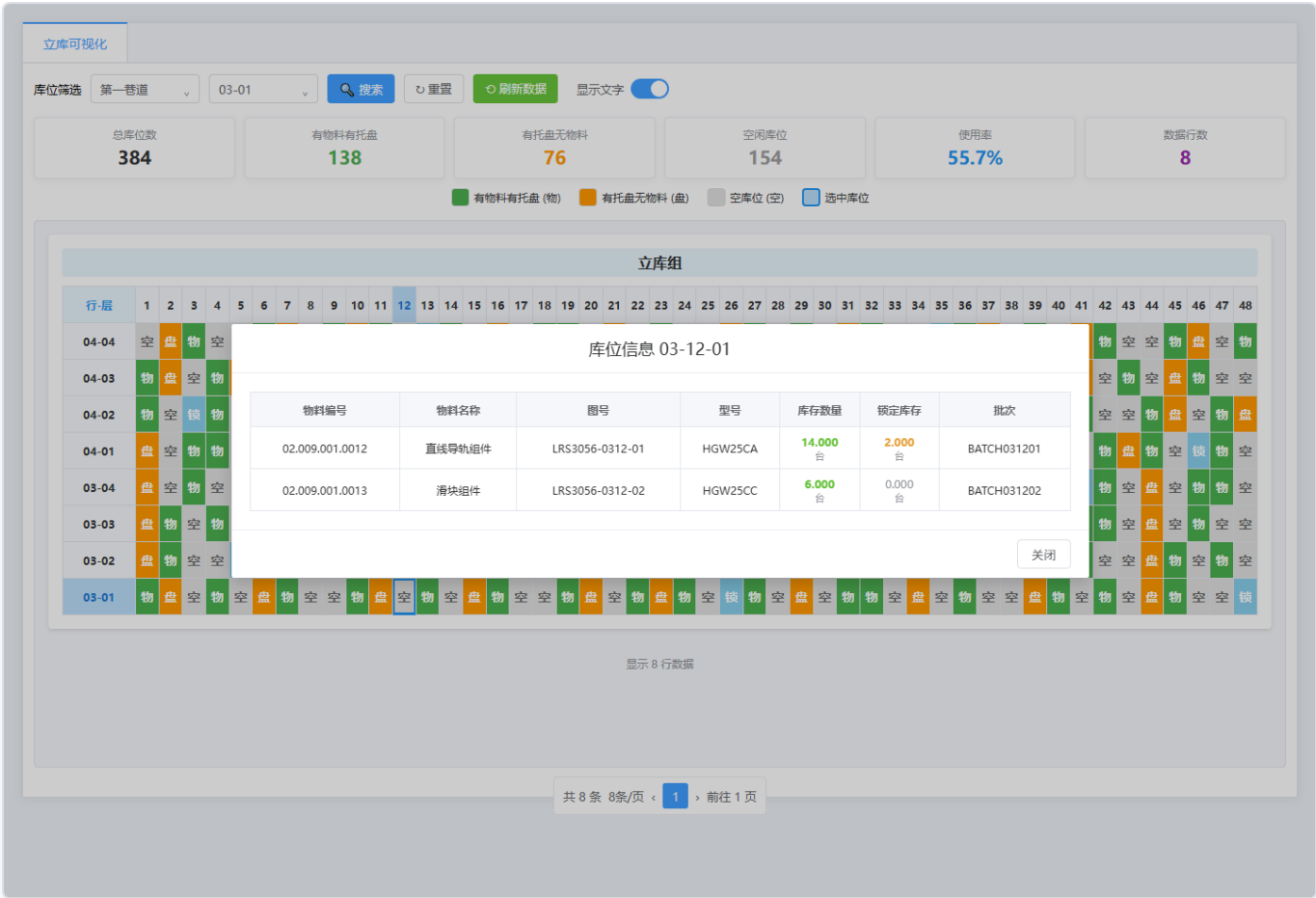
目前该弹窗属于轻量说明弹窗，展示内容较简洁，没有进一步展示：

- 仓库编号
- 创建时间
- 最近操作人
- 任务锁定来源

这些字段在前端对象构造里有部分预留，但未在弹窗模板中完整展示。

10. 物料信息弹窗说明

图 4：物料信息弹窗示意



点击状态为 2 的库位时，会弹出“库位信息”窗口。

触发条件：

- 单元格状态等于 2

10.1 弹窗标题

格式：

- 库位信息 + pno

即使用返回数据第一条的库位编号作为标题显示。

10.2 数据来源

点击单元格后会调用 `StockDetail(rowLayer, column)`，查询数据库视图：

- `StockDetailView`

查询方式为手写 SQL：

- `select ... from StockDetailView where pno='xx-xx-xx'`

10.3 库位编号拼接规则

点击单元格时，前端会把界面上的 行-层 和列号转换为后端查询用的 pno：

1. 将列号补足 2 位
2. 将 行-层 以 - 拆分

3. 按 行-列-层 重新拼装

例如：

- 页面行层： 03-01
- 点击列： 12
- 后端查询编码： 03-12-01

10.4 弹窗字段

当前弹窗表格已展示以下字段：

- 物料编号 mno
- 物料名称 mname
- 图号 spec
- 型号 thao
- 库存数量 qty
- 锁定库存 lockqty
- 批次 batch

其中数量字段已实现：

- 保留三位小数格式化
- 大于 0 时显示强调色
- 在数量下方显示单位 munit

10.5 返回数据兼容处理

前端对接口返回做了多种兼容判断，支持以下几类格式：

1. response.data 直接就是数组
2. response.data.data 是数组
3. response.data.rows 是数组
4. response.data 是单对象，且对象里包含 mno 或 pno

说明：

- 这意味着当前页面对后端返回结构的容错较强。

10.6 无数据提示

如果接口无返回数据，则提示：

- 未查询到数据

11. 后端接口与数据来源说明

当前页面实际对接了两类立库相关数据。

11.1 立库布局/状态数据

请求方法：

- `GetPosition(hang)`

请求构造:

- `this.CreateData('11', 'GetPositionView', param)`

参数内容:

- `hang` : 巷道编号
- `potvar` : 空字符串
- `tt` : 固定值 1

调用方式:

- `this.ExecDatabase1(Data)`

返回结果用途:

- 写入 `defaultWarehouseData`
- 再触发 `refreshData()`
- 最终更新主界面矩阵

11.2 库位物料明细数据

请求方法:

- `StockDetail(rowLayer, column)`

请求方式:

- 直接构造 SQL 查询 `StockDetailView`

调用方式:

- `this.ExecDatabase1(Data)`

返回结果用途:

- 写入 `materialInfoList`
- 打开物料信息弹窗

11.3 接口地址配置

立库接口使用独立配置地址:

- `static/config.js`
- `url1 = http://192.168.1.27:10006`
- `request_config1 = url1 + '/submit/MESCommonBase.ashx'`

说明:

- 这表明立库数据并未走默认 MES 主接口, 而是走单独的立库数据库服务入口。

12. 已实现功能清单

基于当前页面代码, 已经实现并可确认的功能如下。

12.1 页面展示类功能

- 立库矩阵可视化展示
- 行层与列号展示
- 库位状态颜色区分
- 文字/图标两种展示模式切换
- 统计卡片展示
- 状态图例展示
- 分页展示

12.2 数据查询类功能

- 默认加载立库数据
- 按巷道重新查询立库数据
- 点击状态为 2 的库位查询物料明细
- 点击非 2 状态库位查看库位状态说明

12.3 筛选与交互类功能

- 按行层筛选
- 点击搜索按钮重新按巷道加载
- 重置筛选条件
- 刷新当前缓存数据
- 鼠标悬停显示状态提示
- 单元格选中高亮
- 行高亮
- 列高亮

12.4 数据展示增强功能

- 数量保留 3 位小数
- 锁定库存单独显示
- 单位显示
- 物料列表支持多条记录显示
- 接口返回结构兼容处理

13. 代码中存在但当前未完整启用的能力

以下能力在代码中已有部分实现或预留，但从当前模板和实际入口看，尚未完整开放为可见功能。

13.1 状态筛选

页面 `data` 中存在：

- `statusFilter`

筛选逻辑中也保留了状态筛选分支，但顶部没有对应状态下拉框，因此用户当前无法直接按状态筛选。

13.2 关键字搜索

页面 `data` 中存在：

- `searchKeyword`

并保留了按列搜索的逻辑入口，但模板中没有对应输入框，因此当前不构成实际功能。

13.3 图表分析功能

代码中已实现：

- 状态分布饼图
- 行列使用率图
- ECharts 初始化与销毁

但模板中没有对应图表容器，也没有实际启用 `warehouse-analysis` 页签，因此当前属于预留代码。

13.4 数据导出功能

代码中已实现 `exportWarehouseData()`，可以把当前 `warehouseData` 导出为 JSON 文件。

但当前页面模板没有挂出“导出”按钮，因此用户在界面上无法直接使用。

13.5 打印报表功能

代码中存在 `printWarehouseReport()`，但当前只是提示“功能开发中”，没有实际打印逻辑。

13.6 物料标签打印

代码中存在 `printMaterialInfo()`，当前同样仅提示“功能开发中”。

13.7 焦点定位功能

代码中存在 `focusOnCell(rowLayer, column)`，可自动滚动到指定单元格位置。

但当前页面没有表格明细区或其他入口去调用它，因此属于预留能力。

14. 现有业务规则总结

从当前实现可归纳出以下业务规则。

14.1 单元格状态规则

- `2` 代表该库位已有物料，应进入物料明细查询流程
- 非 `2` 状态不查物料，进入库位说明流程
- `9` 表示库位正在存取或已锁定

14.2 巷道数据规则

- 页面默认在 `mounted` 时调用 `GetPosition(1)`，默认先加载第一巷道
- 用户再通过顶部选择器切换其他巷道

14.3 编码转换规则

界面展示编码与后端查询编码不是同一个结构：

- 界面主要使用 行-层
- 后端明细查询使用 行-列-层

因此点击库位时必须做一次编号重组。

15. 使用流程说明

典型操作流程如下：

1. 打开立库可视化页面
2. 系统默认加载第一巷道数据
3. 页面显示统计信息、图例和库位矩阵
4. 用户可切换巷道并点击“搜索”
5. 用户可按“行-层”筛选关注区域
6. 用户可切换文字/图标显示模式
7. 鼠标悬停可查看库位状态提示
8. 点击空库位/托盘位/锁定位，查看库位说明
9. 点击有物料库位，查看该库位物料明细
10. 需要恢复视图时可点击“重置”或“刷新数据”

16. 当前界面优点

从现有实现看，该页面已经具备以下优点：

- 视觉上能直接反映立库占用状态
- 巷道切换清晰
- 物料库位点击查询路径明确
- 状态颜色和图标表达直观
- 对后端返回结构适配较灵活
- 前端交互响应比较轻量

17. 当前界面限制

从代码现状看，也存在一些限制：

- 状态筛选未开放界面入口
- 关键字搜索未开放界面入口
- 导出功能未开放按钮
- 图表分析功能未在页面中启用
- 空数据时缺少明确提示文案
- 锁库状态没有放入图例区
- 库位详情弹窗信息量偏少
- 刷新逻辑更多是前端缓存刷新，不是强制重新拉取所有条件数据

18. 维护与扩展建议

如果后续继续完善该页面，建议优先考虑以下方向：

1. 增加状态筛选下拉框，补齐 `statusFilter` 实际入口
2. 增加库位搜索输入框，支持直接定位 行-层-列
3. 在界面上开放“导出 JSON”按钮
4. 补上锁库状态图例
5. 完善空数据占位提示
6. 把图表分析页签正式接入模板
7. 在库位详情弹窗中增加更多业务字段
8. 将明细查询 SQL 封装为标准后端接口名，减少前端直接拼 SQL 的耦合

19. 结论

当前“立库详细界面”已经不是简单静态展示，而是一个具备实际业务查询能力的立库可视化页面。它已经实现了：

- 按巷道加载立库状态
- 立库矩阵可视化展示
- 库位状态识别
- 库位详情查看
- 库位物料明细查询
- 筛选、分页、刷新、显示模式切换等基础交互

同时，代码中还保留了导出、图表分析、状态筛选、定位等扩展能力，说明该页面已经具备进一步升级为完整“立库监控与查询中心”的基础。