## Home.vue

### fetchHotBooks(limit = 27)

**位置**

* <script setup> 区块，第19 行

**作用**

* 向后端接口拉取热门小说列表，用于首页 <RankingCarousel> 组件展示。

**调用时机**

* 组件挂载时由 onMounted() 自动触发。

**HTTP 请求**

bash

复制编辑

GET /api/books/hot?limit={limit}

Host: <后端域名>

**请求参数**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| limit | Number | 否 | 要获取的小说数量，默认 27 |

**成功响应示例**

* 状态码：200 OK
* Content-Type：application/json

[

{

"id": 1,

"rank": "01",

"title": "惜花芷",

"category": "宫斗",

"cover": "src/assets/covers/book1.jpeg"

},

{

"id": 2,

"rank": "02",

"title": "我不是戏神",

"category": "都市",

"cover": "src/assets/covers/book2.jpeg"

},

…

]

**失败响应示例**

* 状态码：4xx／5xx
* 响应体：

{ "message": "错误描述" }

**前端处理**

1. books.value = data；
2. <RankingCarousel :books="books" /> 自动更新渲染。

## NovelDetail.vue

### loadNovelDetail()

**作用**

* 从后端获取该小说的基本信息（标题、作者、封面、简介、更新时间等），用于页面顶部展示。

**调用时机**

* 组件挂载时由 onMounted() 自动调用一次。

**HTTP 请求**

GET /api/novel/{novelId}

Host: <后端域名>

Authorization: Bearer <token> // 如需鉴权

**成功响应示例**

* 状态码：200 OK

{

"id": 123,

"title": "惜花芷",

"author": "空留",

"category": "宫斗",

"updateTime": "2025-04-15 11:45",

"cover": "/assets/covers/book1.jpeg",

"description": "双生妹妹嫁入皇宫前夕…"

}

**失败响应示例**

{ "message": "小说不存在或服务器错误" }

**前端处理**

1. novel.value = data；
2. 捕获错误后 console.error，可加 UI 提示或占位数据。

### loadChaptersList()

**作用**

* 获取该小说的章节列表，用于“目录”版块渲染。

**调用时机**

* 与 loadNovelDetail() 并行，在 onMounted() 中调用。

**HTTP 请求**

GET /api/novel/{novelId}/chapters

Host: <后端域名>

Authorization: Bearer <token>

**成功响应示例**

* 状态码：200 OK

[

{ "id": 1, "title": "第一章 替嫁" },

{ "id": 2, "title": "第二章 拿惯银枪的手" },

…

]

**失败响应示例**

{ "message": "获取章节列表失败" }

**前端处理**

1. chaptersList.value = data；
2. 目录由 <router-link> + v-for 自动渲染。

### fetchComments()

**作用**

* 拉取该小说的所有评论，用于“评论区”展示。

**调用时机**

* 组件挂载时 onMounted() 调用一次；
* 用户提交新评论后再次调用以刷新列表。

**HTTP 请求**

GET /api/novel/{novelId}/comments

Host: <后端域名>

Authorization: Bearer <token>

**成功响应示例**

* 状态码：200 OK

[

{

"id": 1001,

"author": "Alice",

"content": "太好看了！",

"createdAt": "2025-04-26T10:30:00Z",

"likes": 5

},

…

]

**失败响应示例**

{ "message": "加载评论失败" }

**前端处理**

1. comments.value = data；
2. 通过计算属性 sortedComments 排序并渲染。

### submitComment()

**作用**

* 向后端提交用户新评论，提交成功后刷新评论列表。

**触发方式**

* 用户点击“发表评论”按钮。

**HTTP 请求**

POST /api/novel/{novelId}/comments

Host: <后端域名>

Authorization: Bearer <token>

Content-Type: application/json

{

"content": "写下你的评论…"

}

**成功响应示例**

* 状态码：201 Created

{

"id": 1007,

"author": "当前用户",

"content": "写下你的评论…",

"createdAt": "2025-04-27T13:00:00Z",

"likes": 0

}

**失败响应示例**

{ "message": "发布失败，请重试" }

**前端处理**

1. newComment.value = ''；
2. 调用 fetchComments() 刷新列表；
3. 捕获错误后 console.error 并可提示用户。

### toggleFavorite()

**作用**

* 切换用户对该小说的“收藏”状态，并同步到后端。

**触发方式**

* 用户点击“收藏”按钮时调用。

**HTTP 请求**

POST /api/novel/{novelId}/favorite

Host: <后端域名>

Authorization: Bearer <token>

Content-Type: application/json

{

"favorite": true // 或 false

}

**成功响应示例**

* 状态码：200 OK

{ "favorite": true }

**失败响应示例**

{ "message": "同步收藏状态失败" }

**前端处理**

1. 立即切换 isFavorited.value；
2. 后端失败时回滚并可提示用户。

## NovelContent.vue

### loadNovelMeta()

**作用**

* 从后端获取小说元信息（主要是标题），用于页面顶部展示。

**触发方式**

* 组件挂载时 onMounted() 自动调用一次。

**HTTP 请求**

GET /api/novel/{novelId}

Host: <后端域名>

Authorization: Bearer <token>

**成功响应示例**

* 状态码：200 OK

{

"id": 123,

"title": "惜花芷",

"author": "空留",

"updateTime": "2025-04-15T11:45:00Z",

…

}

**失败响应示例**

{ "message": "加载小说信息失败" }

**前端处理**

* novelTitle.value = data.title；
* 错误时 console.error 并可提示用户。

### loadChaptersList()

**作用**

* 获取小说的章节列表，用于“上一章/下一章”导航。

**触发方式**

* 组件挂载时 onMounted() 自动调用一次。

**HTTP 请求**

GET /api/novel/{novelId}/chapters

Host: <后端域名>

Authorization: Bearer <token>

**成功响应示例**

* 状态码：200 OK

[

{ "id": 1, "title": "第一章 替嫁" },

…

]

**失败响应示例**

{ "message": "加载章节列表失败" }

**前端处理**

* chapters.value = data；
* 计算 currentChapter、nextChapter、prevChapter。

### loadChapterContent()

**作用**

* 拉取当前章节的完整文本内容，用于正文按行渲染。

**触发方式**

* 组件挂载时调用；
* 路由参数变化时再次调用。

**HTTP 请求**

GET /api/novel/{novelId}/chapters/{chapterId}/content

Host: <后端域名>

Authorization: Bearer <token>

**成功响应示例**

* 状态码：200 OK

{

"id": 2,

"title": "第二章 拿惯银枪的手",

"content": "第一段…\n第二段…\n…"

}

**失败响应示例**

{ "message": "加载章节内容失败" }

**前端处理**

* currentChapter.value = data；
* 由 contentLines 计算属性拆行渲染；
* 错误时 console.error 并可提示用户。

## Search.vue

### searchNovels(q)

**作用**

* 向后端发起搜索请求，根据关键词获取匹配的小说列表。

**触发方式**

* doSearch() 点击搜索按钮或回车时调用；
* 页面加载或路由 ?q=xxx 变化时调用。

**HTTP 请求**

GET /api/novel/search?q={q}

Host: <后端域名>

Authorization: Bearer <token>

**请求参数**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| q | String | 是 | 搜索关键词 |

**成功响应示例**

* 状态码：200 OK

[

{

"id": 101,

"title": "惜花芷",

"author": "空留",

"status": "连载中",

…

},

…

]

**失败响应示例**

{ "message": "搜索失败，请稍后重试" }

**前端处理**

1. searchResults.value = data；
2. hasSearched.value = true；
3. 错误时 console.error 并可提示用户；
4. 结果由 filtered 计算属性渲染。