最重要的规范：

1.api调用时，禁止硬编码

Vue调用c#：

禁止：axios.get('http://localhost:5000/api/xxx')

应用：src/modules/book/api.js

import http from '@/services/http'

http.get(`/book/${id}`)

C#调用Oracle：

第一部分readme规范

1.按照前端/后端/数据库 需要导入的包列出

例如 npm install \*\*\*\*

第二部分 Database规范

1. data/、functions/、triggers/、views/中，所有表/函数/触发器加前缀，避免命名冲突

| **模块** | **表/函数/触发器前缀** |
| --- | --- |
| 读者组 | reader\_ |
| 书籍组 | book\_ |
| 管理员组 | admin\_ |

2.使用CREATE OR REPLACE而不是CREATE FUNCTION

举例：



正确做法：

| **类型** | **推荐语法** |
| --- | --- |
| 函数 | CREATE OR REPLACE FUNCTION ... |
| 触发器 | CREATE OR REPLACE TRIGGER ... |
| 过程 | CREATE OR REPLACE PROCEDURE ... |
| 视图 | CREATE OR REPLACE VIEW ... |

第三部分Vue规范

1.pages下的vue负责组织各component下的组件，。

尽量把渲染的元素抽象出来，放置在/reader（替换为自己小组）/components/\*\*\*.vue中，保证pages下vue的整洁。

2.可能共用的组件放在/frontend/src/shared下

例如：现在已经在shared中的顶部导航栏