### JavaScript 技术管理文档

#### a. 强制（必须遵守）

1. 必须使用 ES6 及以上语法，禁止使用 var 定义变量，统一使用 let 和 const。
2. 所有函数必须使用箭头函数（除非有特殊 this 绑定需求）。
3. 每个文件只包含一个默认导出。
4. 禁止使用未定义的变量。
5. 使用严格等于 (===) 和不等于 (!==)，禁止使用 == 和 !=。
6. 函数和类必须有明确的导出或注释说明其用途。
7. 类名必须使用大驼峰命名法（如：UserProfile）。
8. 变量名必须使用小驼峰命名法（如：userName）。
9. 文件命名必须使用小写字母和中划线分隔（如：user-service.js）。
10. 必须开启 ESLint 并配置团队统一规范。
11. 禁止在业务逻辑中出现 console.log，只允许在调试阶段使用。
12. 必须使用模板字符串（``）拼接字符串，禁止使用 + 拼接。
13. 每行代码长度不得超过 100 字符。
14. 语句必须以分号结束。
15. 不得在循环中定义函数或使用 await。
16. 必须使用 async/await 处理异步，禁止混用回调和 promise。
17. 禁止使用魔法数字，必须用常量或枚举代替。
18. 所有接口调用必须封装为 API 模块，禁止直接调用 fetch/axios。

#### **b. 推荐（强烈建议遵守）**

1. 组件文件应尽量控制在 300 行以内。
2. 优先使用函数式组件而非类组件（React 环境下）。
3. 业务代码应划分模块，每个模块应有独立目录结构。
4. 常量统一存放在 constants.js 文件中管理。
5. 注释应使用英文，遵循 JSDoc 风格进行函数和类注释。
6. 项目应使用统一的 UI 框架（如 Element Plus、Ant Design）。
7. 尽量避免使用 any 类型（TypeScript 项目中）。
8. 避免嵌套超过 3 层的 if/else 结构，应使用提前 return 减少嵌套。
9. 组件 props 应定义默认值和类型检查。
10. 项目结构应遵循约定式目录命名，如 components/, views/, services/。
11. 每次提交代码前建议执行代码格式化和单元测试。
12. 使用解构赋值简化代码结构。
13. 尽量避免直接修改 DOM，优先使用框架内置操作。
14. 每个模块（类或文件）应写单元测试（覆盖率建议 > 80%）。

#### **c. 允许（可视情况使用）**

1. 可根据实际需要使用类或对象工厂模式组织逻辑。
2. 可以在合适场景使用 forEach、map、reduce 等函数式 API。
3. 开发阶段可使用浏览器插件如 Redux DevTools 协助调试。
4. 可在单一文件中编写样式（如 Vue 的 <style scoped>）。
5. 可引入 lodash 等工具库提高开发效率，但应控制体积。
6. 可在项目中使用 TypeScript 增强类型安全。
7. 允许使用 JS Proxy、Reflect 等高级特性，但需注释说明使用理由。
8. 部分模块允许使用动态 import 实现懒加载优化性能。