目录

核心原则：STAR 法则 + “我”的主体性

在描述任何项目或问题时，遵循 **STAR 法则（Situation, Task, Action, Result）**，并确保每个环节都以“我”为主导。

- **S（情景）：** “**我**当时所在的项目是…，它的主要目标是…”
- **T（任务）：** “**我**在这个项目中的主要职责是…，**我**需要解决的具体问题是…”
- **A（行动）：** “**我**是这么思考和行动的：第一…，第二…”（**这是重点，要详细展开**）
- **R（结果）：** “最终，通过**我**的方案，达到了…效果，数据上有…提升，并且**我**总结了…经验”

一、“我”的项目与模块（突出个人贡献）

不要说： “我们团队做了一个电商项目，用了 Vue 和 React。”
要这样说： “**我**作为前端核心开发，主导了『购物车』和『商品详情页』这两个核心模块的重构。**我**的技术选型是 Vue 3 + Pinia，因为看中了其 Composition API 带来的逻辑复用优势。”

需要强调的细节：

1. **我的角色：**是主导、负责、还是参与？是从0到1，还是重构优化？
2. **我的技术选型：** “**我**选择了 `Webpack/Vite` 是因为…”、“**我**引入了 `Tailwind CSS` 是为了解决…效率问题”。
3. **我的架构设计：** “**我**设计了项目的状态管理结构，将用户数据与UI状态分离…” “**我**封装了高复用的 `Table` 组件和 `useRequest` Hook，被全组使用…”

遗漏点： 量化你的影响。例如：“**我**封装的这个组件，让同类需求的开发时间减少了 **30%**”、“**我**优化的首屏加载方案，让 LCP 时间从 **1.5s 降低到了 0.8s**”。

二、“我”遇到的问题与挑战（展现排查和定义问题的能力）

不要说： “项目遇到过性能问题，后来解决了。”
要这样说： “在开发过程中，**我**遇到了一个非常棘手的问题：在生产环境下，列表页在低端手机上滚动时会出现明显卡顿。**我**的第一步是使用 `Chrome Performance` 面板和 `Lighthouse` 进行定位，**我**发现是因为…”

需要强调的细节：

1. **我如何定位：**明确说出你用的工具（Performance、Lighthouse、Sourcemap、Sentry）和方法。
2. **我如何分析：**展现你的逻辑思维。“**我**首先排除了网络问题，然后排除了接口问题，最终将问题锁定在前端代码的…上。”
3. **问题的本质：** “**我**发现问题的根本原因不是代码写得不好，而是由于**我**在 `useEffect` 里错误地依赖了一个引用类型变量，导致了无限重渲染。”

遗漏点： 可以准备一些有深度的、能体现你综合能力的问题，如：

- **跨域/部署问题：** “**我**在部署时遇到了 `CORS` 问题，**我**是通过配置 `Nginx` 反向代理来解决的，而不是简单让后端设置 `Access-Control-Allow-Origin: *`。”
- **兼容性/样式问题：** “**我**解决了某个组件在 Safari 上的样式异常，原因是**我**发现 `flexbox` 和 `min-height` 在 Safari 下的渲染机制不同。”

三、“我”的思考与解决方案（体现技术深度和决策能力）

这是面试的核心，要详细说明你的思维过程。

不要说： “我用节流优化了一下。”

要这样说： “针对这个滚动性能问题，**我**思考了三种方案：

1. **方案A（节流）：**简单，但不能从根本上解决大量 DOM 渲染的压力。
2. **方案B（虚拟列表）：**效果最好，但当时工期紧张，实现复杂。
3. **方案C（懒加载 + 分页）：**能较快见效，且符合业务场景。
   我的决策是：先采用**方案C**快速上线缓解问题，同时在技术债清单中创建了『虚拟列表』的优化项，并在后续迭代中由**我**主导完成了方案B的落地。”

需要强调的细节：

1. **我的权衡：**展现你思考的全面性（性能、成本、业务、可维护性）。
2. **我的技术细节：** “**我**没有直接用现成的 `lodash.throttle`，而是自己实现了一个，因为需要处理特定的 leading 和 trailing 边缘情况…” “**我**在实现虚拟列表时，关键点是**我**计算了…”
3. **我的创新/优化：** “**我**在 Webpack 配置中独创了一个 `splitChunks` 策略，将…单独打包。”

四、 “我”的总结与复盘（体现学习能力和成长性）

不要说： “解决了，没问题了。”
要这样说： “事后**我**对整个解决过程做了复盘，并总结了三条经验：

1. **技术层面：****我**意识到以后写 `useEffect` 的依赖数组时要格外小心引用类型。
2. **流程层面：****我**向团队提议并将 `Performance` 评测加入了代码合并前的 Checklist，防止同类问题再次发生。
3. **个人成长：**通过这次经历，**我**对浏览器渲染原理和 React 更新机制的理解更深了。**我**还把研究过程写成了一篇技术博客分享给了团队。”

遗漏点：

- **沉淀与分享：**强调你是否将经验文档化、工具化、流程化，这体现了你的团队贡献度和工程师思维。
- **主动改进：** “**我**因此为公司前端 Wiki 贡献了一篇《前端性能优化指南》。”

五、 容易遗漏的“我”的亮点

1. **“我”的主动性：**
   - “在完成本职工作后，**我**主动调研并引入了 `eslint-plugin-security`，帮助团队规避了常见的安全风险。”
   - “**我**发现团队的构建速度很慢，主动去学习了 `Vite`，并搭建了一个原型向团队演示，最终说服大家成功迁移。”
2. **“我”的软技能：**
   - **沟通协作：** “**我**和后端同学一起制定了新的 `RESTful` 接口规范…” “当和产品经理对需求有分歧时，**我**是通过画技术原型和列举数据来说服对方的…”
   - **项目管理：** “**我**不仅负责开发，还**我**用 `Jira` 拆分了任务并估算了工时，最终这个模块**我**提前了两天完成。”
3. **“我”的深度：**
   - 在回答原理题时，也要用“我”。“对于 `Vue` 的响应式原理，**我**的理解是…，**我**还专门写过一版简化的实现来加深理解。”
   - “**我**不仅会用 `Webpack`，**我**还学习过它的插件机制，并写过一个简单的 `BundleAnalyzerPlugin` 来分析产物。”

总结一句话： 面试就是讲一个关于“你”的故事，一个关于“你”如何思考、如何行动、如何解决问题、如何成长的故事。把“我们”变成“我”，把你的价值和能力清晰地“喊”出来。