

1. 怎样去管理框架默认与webpack配置.

$\left\{ \begin{array}{l} we: we.config.js \\ create-react-app: eject \end{array} \right\}$  ① configure Webpack: 声明式  
② 链式操作: 命令式

长: 对于脚手架下项目, 如果配置(webpack.config.js)直接复制到项目本地, 存在不能及时更新最新特性的弊端, 需要直接复制, 而是使用脚手架(npm包)

在部署脚手架时, 需要多次模板中引入依赖包, 一层, 要版本与依赖包  
的兼容(让模板依赖包不冲突)

2. 多页应用打包场景.

1) HS: 前端技术在移动端的应用.

2) SPA  $\left\{ \begin{array}{l} 1 \times \text{级应用} \\ \text{关联性强, 体验好.} \end{array} \right.$

3) MPA: 页面, 去跟页: 一般由: 替换一次

4) 多页面多时, 内存不足  $\rightarrow$  需要打包  
① 同时换空间.

$\left. \begin{array}{l} \text{SPA} \\ \text{MPA} \end{array} \right\} \rightarrow \text{取决于业务场景.}$

5) 一个bundle 包含多个chunk.

3. 项目多环境(开发, 测试, 部署, 生产)怎样配置环境变量.

1) 在编译阶段设置不同值.

2) 使用 process.env.xxx

配置:  $\left. \begin{array}{l} .env.development \\ .env.production \end{array} \right\}$

$\left. \begin{array}{l} \text{we中} \rightarrow \text{react 兼容性} \end{array} \right\}$

$\rightarrow$  其他 --mode 指定其他模式!  
与 default plugin 一致!

② 配置时, 使用 dotenv 来配置!

4. webpack打包, 按文件类型输出到不同的文件夹.

1) 不要输出到相同文件夹(如果非要, 要说明)

2) 在相同的 loader 中配置

- js: output
- css: url-loader

5. Webpack 面试问题.

- 1. Webpack 的价值体现 / 你对 Webpack 的看法
- 2. Webpack 与 Gulp、Grunt 之类工具的差异
- 3. Loader 与 Plugin 之间的差异
- 4. 用过的 Loader 和 Plugin (以此来判断你遇到过的问题)
- 5. Tree-shaking、sideEffects
- 6. Source Map 的最佳实践
- 7. Webpack 的核心工作过程
- 8. 文件名 hash (文件指纹) 控制缓存

(sentry) 监控平台

文件指纹

- noSource-Source-map +
- noEmit-Source-map

面试技巧

- 1. 自我介绍
- 2. 实操
- 3. 问题
- 4. 升华

6. 问题补充

1) rules: → rule → test:

- use
- options
- include
- exclude
- enforce

2)

