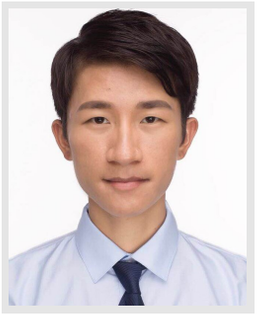
陈旭东



3年前端及nodejs全栈开发经验，能独立完成项目的前端开发、搭建webpack前端工程化、前端性能优化、nginx部署、nodejs后端开发、数据库设计、docker部署等。

基本信息

男 | 26岁 | 2年经验 | 15767973362 | 248200851@qq.com |

(惠州学院-本科) 计算机科学与技术专业 | 前端以及nodejs后端

个人作品

http://www.cxdsimple.com/cv/
(个人线上简历，本简历采用技术：webpack，ts，node koa2，nginx，docker)

https://simplecodecx.github.io/archives/
(个人技术博客)

https://github.com/SimpleCodeCX/koa-ts-cli
koa-ts-cli：实现了一个创建 node koa2 + ts 工程的脚手架（提供了三个模板）

工作经验

2017.3-至今 **惠州市德赛西威汽车电子股份有限公司** **前端开发（兼nodejs后端）**
>2000人

所在部门：德赛西威技术研究院
主要工作内容：
一、车联网数据云平台 (前端)
本项目实现基于车联网数据的中后台系统
采用技术栈：angular、typescript、scss、webpack、nginx、docker 等
1、负责业务模块的开发
2、设计并实现了统一封装 api 数据接口的解决方案：将后端提供的 api 数据接口统一封装成单独的模块，而上层的各个组件只需通过依赖注入的方式，注入接口模块中的api service进行数据获取即可。
为团队带来的效益如下：
1）使接口模块实现多系统复用
2）解决团队成员代码风格问题，减少接口代码统一优化和升级成本
3）降低接口维护成本，规范开发流程，显著提高了开发效率
3、负责对项目进行性能优化,参考 pagespeed 以及配置 nginx 性能优化策略如下:
1) 结合 webpack 的分块打包和 hashChunk 打包机制，以及 http 的强缓存原理对 css、js 以及图片等资源设置强缓存
2) 配置 nginx 对 css、js 以及 json 等文本资源进行 gzip 压缩，并结合 http1.1 的分块传输机制开启 http 持久连接并边压缩边输出，减少 TTFB 时间
3) 对于图片资源，通过 nginx 检测浏览器是否支持 webp 格式图片，并结合 nginx rewrite 指令实现 webp 格式和其他图片格式的自动切换，实现零成本兼容 webp（无需修改项目代码）
优化效果: 项目的首屏时间从4秒降低至1秒
难点：熟悉 http 协议相关缓存机制，熟悉 nginx 相关指令配置及原理，以及熟悉各种不同图片格式的应用场景
4、参与编写公共组件库
5、参与设计项目部署策略，编写 dockerfile 并通过 docker + nginx 进行部署, 并将镜像备份于 nexus docker 私服中,从而提高了部署效率、方便版本恢复以及部署迁移

二、负责 node 微服务的开发

1、负责搭建 node koa2 + typescript 工程，集成了 jest、apidoc、docker、eslint、husky 等功能
并产出了脚手架：koa-ts-cli，该脚手架用来初始化 koa2 + ts 工程，节省了每次创建新项目的大量配置时间。
koa-ts-cli 的 npm 地址如下: https://www.npmjs.com/package/koa-ts-cli

2、负责对 node 后端的技术选型 (koa2,阅读过源码), 数据库设计 (mariadb), 项目架构设计 (route层 + controller层 + bll层 + dal层 + model层)

3、负责设计规范的 resful api 接口、设计 koa2 中间件（比如跨域中间件、日志采集上报中间件 log4js + logstash、全局错误处理 try catch 中间件等）、设计单元测试用例（supertest）

4、定制统一的接口响应规范、统一接口出错处理以及全局错误码等

个人项目

2019.3-至今　　个人线上简历（前端+node后端）　　独立完成

实现一个能展现自己个人技术能力的线上简历，本项目涉及到的技术有：前端、性能优化、webpack、node 后端、resful api、数据库设计、nginx 部署、docker 部署。

前端通过搭建 webpack4 工程，采用轻量级前端模板引擎 art-template 以及 typescript 实现，通过 nginx + docker 进行部署，并配置了 nginx 的性能优化策略，部署于自己的 centos 服务器，后端采用 nodejs koa2，设计相关的中间件、以及数据库，为前端提供 resful api 接口。

线上简历访问地址：http://www.cxdsimple.com/cv/

项目 github 地址：(前端) https://github.com/SimpleCodeCX/cv
(后端) https://github.com/SimpleCodeCX/cv-server

具体如下：

1、webpack：搭建 webpack4 前端工程,实现编译打包 scss（并通过 postcss-loader 自动处理 css 前缀）、typescript、图片、字体、并实现页面实时刷新等；实现代码模块分割，将第三方库、css、异步模块分别打包成单独的 chunk 并以 chunkhash 命名,从而有利于缓存优化。

在搭建此项目的过程中，我写了一套关于 webpack4 系列教程：http://www.cxdsimple.com/blog/20190325/720ca7f0.html

2、性能优化：配置 nginx 性能优化策略如下:

1) 结合 webpack 的分块打包和 hashChunk 打包机制，以及 http 的强缓存原理对 css、js 以及图片等资源设置强缓存

2) 配置 nginx 对 css、js 以及 json 等文本资源进行 gzip 压缩

3) 对于图片资源，通过 nginx 检测浏览器是否支持 webp 格式图片，并实现 webp 格式和其他图片格式的自动切换

3、后端：采用 nodejs koa2，实现 resful api 接口设计、koa2 中间件设计 (比如跨域中间件、全局错误 try catch 中间件等)，定制统一的接口规范，统一的出错处理，单元测试 (supertest) 等

4、docker 部署：编写相应的 Dockerfile 文件，前后端都进行 docker 部署

个人技能

1、扎实的 javascript 基础

2、擅长前端 react 框架（熟悉 angular、了解vue），阅读并理解过 react 的相关源码，比如：redux、react-redux、react-router、dom diff 等。

3、擅长 nodejs，有后端开发和部署经验，阅读过 nodejs koa2 源码，目前负责了公司 node 微服务的开发，负责脚手架开发，接口设计、数据库设计、koa2 中间件开发、单元测试用例的编写等。

4、熟悉webpack，阅读过webpack的部分源码，比如LoaderRunner、基于Tapable的webpack插件机制，能基于webpack搭建一套完整的前端工程化。

5、熟悉 web 性能优化、能够基于 webpack + nginx 配置 web 性能优化策略

6、熟悉 typescript，我整合了一个 node koa2 + typescript 的工程方案，并产出了一个脚手架：koa-ts-cli
https://www.npmjs.com/package/koa-ts-cli

7、熟悉 scss，并利用 scss 编写可复用 css，并运用于项目中

8、熟悉 git，有 gitLab 项目管理经验

9、有 docker、nginx 部署经验