前端项目全流程所需技术栈

1. Github action
2. 书写你对下面代码的理解 可在后面更上相应的注释

1. 如果此时 我想在合并到main分支上时去触发下列行为的时候，应该如何修改

将branches 修改为main

# work flow name

name: Docker Image Dev V1 Env

# when to trage this work flow

# 监听v1-dev,当v1-dev分支push commit后，将执行ci

on:

  push:

    branches:

      - "v1-dev"

    # tags:

    #   - "v1.[0-9]+.[0-9]+-dev"

# setup default working dir

defaults:

  run:

shell: bash

# 工作区域路径

    working-directory: ./App

# the jobs to run this ci

jobs:

  build:

    runs-on: ubuntu-latest

    steps:

      # pull all branch and tag data, 拉取项目代码到当前工作区

      - name: Check out the repo

        uses: actions/checkout@v3

      # get the tag name

      # - name: Get the Tag Name

      #   id: tagName

      #   run: echo "::set-output name=tag::${GITHUB\_REF#refs/tags/}"

      # publish the docker to github package，运行docker action，将会运行配置dokcer的文件，将生成的内容发布到ghcr.io

      - name: publish to github package

        uses: elgohr/Publish-Docker-Github-Action@v4

        with:

          name: datareachable/dr\_Front\_surveyDistribution

          username: ${{ github.actor }}

          password: ${{ secrets.FRONT\_CI\_SECRETS }}

          registry: ghcr.io

          dockerfile: Dockerfile\_v1\_dev

          tags: "v1-dev"

1. 学习如何在github action中获取git commit中的信息
2. Github Package

该模块还没学习

1. 什么是npm

Sadada

1. 发布一个npm包的命令是什么？
2. 如何将包发布到github 上
3. 如何发布一个1.0.0的包我需要再Package.json里干点什么，package.json的作用是什么
4. Docker
5. 解读一下下面代码 在后面更上你对于这行的理解

FROM node:16.14.2 AS builder //在node环境中运行，node环境来自于docker镜像生成

WORKDIR /app //设置当前工作目录

COPY . . //把本地项目复制到当前工作目录下·

WORKDIR /app/App //设置工作目录

RUN yarn install && yarn testV2 //运行命令

FROM nginx:alpine //在nginx环境中运行，nginx来自于镜像生成

WORKDIR /usr/share/nginx/html //设置当前工作目录

RUN rm -rf ./\* //删除当前目录下文件

COPY --from=builder /app/App/build . //把在builder中生成的文件，复制到当前工作区，/usr/share/nginx/html

COPY nginx.conf /etc/nginx/nginx.conf //复制 nginx.conf 到 /etc/nginx/nginx.conf

ENTRYPOINT ["nginx", "-g", "daemon off;"] //设置运行nginx服务的脚本

B.这个时候我想在再运行一下buildV2脚本我应该如何书写

FROM node:16.14.2 AS builder

WORKDIR /app

COPY . .

WORKDIR /app/App

RUN yarn install && yarn buildV2

FROM nginx:alpine

WORKDIR /usr/share/nginx/html

RUN rm -rf ./\*

COPY --from=builder /app/App/build .

COPY nginx.conf /etc/nginx/nginx.conf

ENTRYPOINT ["nginx", "-g", "daemon off;"]

1. Nginx

场景 React router 为history模式多级路由, 打包后本地测试访问后子路由无法访问。为什么会出现这样的情况? nginx如何配置才能解决? 为什么它可以解决这个问题?

<https://github.com/szkj-web-he/test20220813>

main分支 是文件

nginx是nginx配置文件

(1)打包后文件经过编译，无法实现react router的功能。

(2)

Server{

1. location ^~/about/ {
2. proxy\_pass http://10.109.55.33:83/;
3. }

(3) 通过配置 location ^~/about/，可以匹配以/about开头的路径。

1. Webpack
2. 如果我想打包一个ts文件，我应该有什么配置

1.使用babel-loader，用法如下

(1)使用插件 @babel/preset-typescript

在babel.config.json中配置：

{

    "presets": [

        "@babel/preset-typescript",

    ],

}

（2）在webpack.config中配置

        extensions: ['.ts', '.js'],

    },

    module: {

        rules: [

            {

                test: /\.(t|j)s?$/,

                use: [

                    {

                        loader: 'thread-loader',

                        options: jsWorkerPool,

                    },

                    {

                        loader: 'babel-loader',

                    },

                ],

            },

        ],

},

 output: {

        path: path.resolve(\_\_dirname, '../build'),

        publicPath: '/',

        pathinfo: false,

    },

1. 如果想要配置开发模式的webpack,那就应该相应的在生产模式的配置文件上加什么配置项

mode: 'production',

1. 学习什么是tree shaking，当有个css文件在打包后丢失了如果办

（1）webpack4以后，在生产模式下，webpack内部会删除一些没有使用的代码。

（2）在package.json中的sideEffects属性中，加入该文件的路径

1. webpack一般是怎么拆分代码的

拆分可以分为三个部分

1.入口chunk

 plugins: [

        new HtmlWebpackPlugin({

            template: './public/index.html',

            favicon: './public/favicon.ico',

        }),

        new MiniCssExtractPlugin({

            filename: 'css/[name].css',

        }),

        new CompressionPlugin({ test: /\.js(\?.\*)?$/i, algorithm: 'gzip' }),

    ],

    output: {

        filename: 'js/[name].[contenthash].js',

    },

2.组件懒加载 chunk

  chunkFilename: 'js/[name].[contenthash].js',

3.runtime chunk

连接各个模块和执行的chunk。

1. source map的作用

代码映射，设置该属性后，在浏览器中可以查看源码代码。

作卷人：刘存明