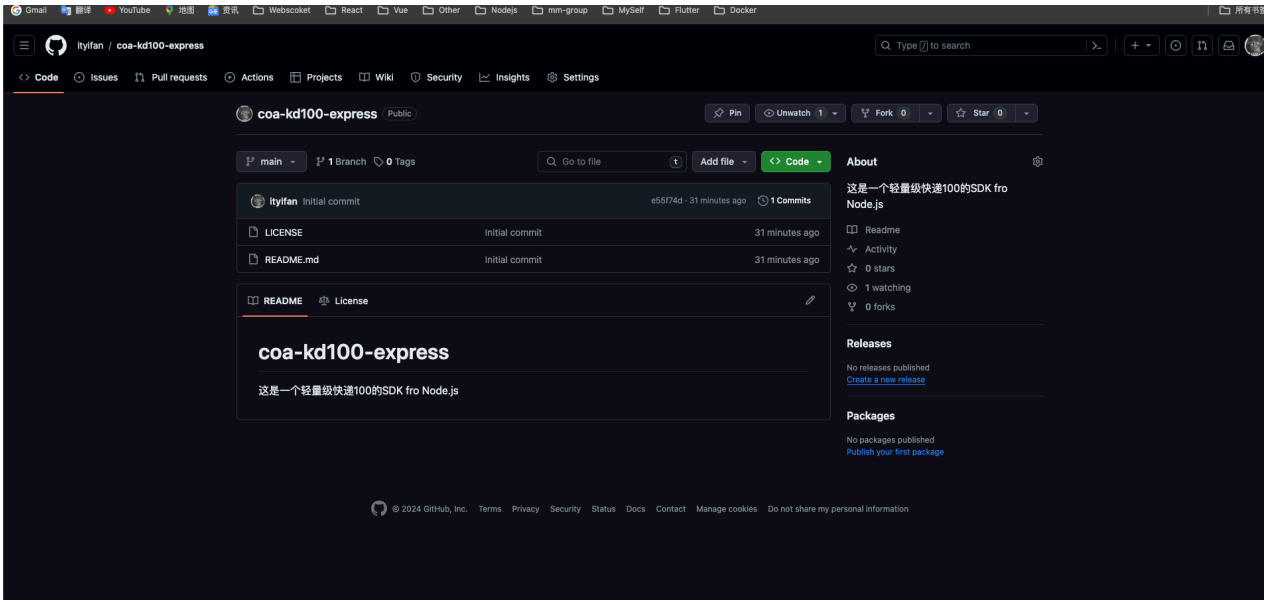
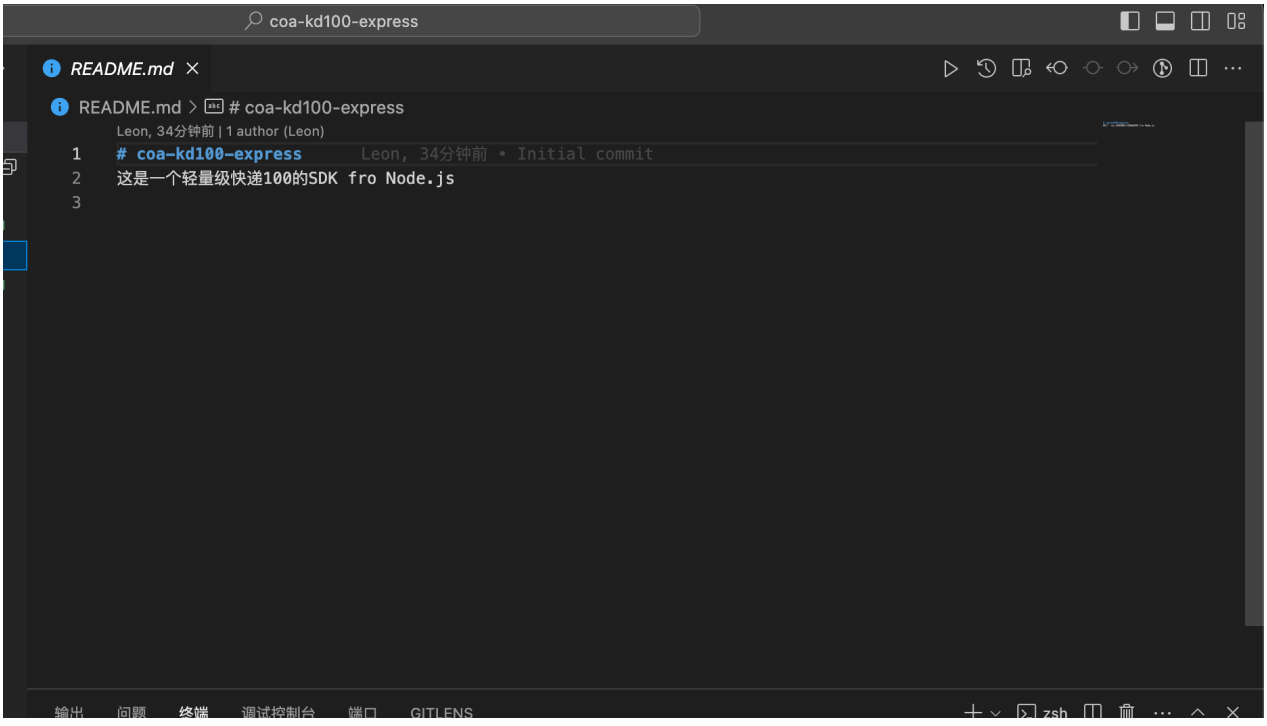
论GitHub和npm的自动发布

如果你想自己发布一个npm包，那么就根据一下的流程来走.....

1.去GitHub仓库创建一个项目



2.把项目拉下来放入VScode内



3.初始化package.json

```
npm init
```

3.1把package.josn内部部分修改为

当然 你初始化的github地址和 名称以及描述 关键词等与我的应该有所不同

```
{
  "name": "coa-kd100-express",
  "version": "1.0.0",
  "description": "这是一个轻量级快递100的SDK fro Node.js",
  "main": "index.js",
  "scripts": {
    "tsc": "tsc -w",
    "build": "rm -rf dist && tsc && cp package.json *.md dist && rm -rf
dist/test",
    "test": "NODE_PATH=dist tsc-watch --onSuccess \"node dist/test\"",
    "lint": "eslint . ",
    "prettier": "prettier -w .",
    "sync": "curl -X PUT 'https://npm.taobao.org/sync/coa-dg-pay?
sync_upstream=true'"
  },
  "repository": {
    "type": "git",
    "url": "git+https://github.com/ityifan/coa-kd100-express.git"
  },
  "keywords": [
    "coa",
    "isv",
    "kd100",
    "ts"
  ],
  "author": "Leon",
  "license": "MIT",
  "bugs": {
    "url": "https://github.com/ityifan/coa-kd100-express/issues"
  },
  "homepage": "https://github.com/ityifan/coa-kd100-express#readme",
   "devDependencies": {
    "@types/node": "^16.9.6",
    "@typescript-eslint/eslint-plugin": "^4.31.2",
    "@typescript-eslint/parser": "^4.31.2",
    "eslint": "^7.32.0",
    "eslint-config-prettier": "^8.3.0",
    "prettier": "^2.4.1",
    "typescript": "^4.4.3"
  }
}
```

4.初始化tsconfig.json

```
tsc -init / tsc --init
```

4.1将tsconfig.json内部修改为

这是我喜欢用的版本和outDir 如果你有自己喜欢的版本和outDir也可以自行修改并且调试

```
{
  "compilerOptions": {
    "strict": true,
    "module": "commonjs",
    "target": "es2019",
    "outDir": "dist",
    "declaration": true
  },
  "include": ["src"]
}
```

5.接下来我们创建文件夹和一些配置文件

src （ 文件夹 作为我们项目的根目录）

```
src
 --libs
 --services
 --test
 --index.ts
 --typings.ts
```

.eslintgnore （eslint不检查规范的文件夹）

内容如下

```
dist
```

.eslintrc.json (eslint语法检查配置 可以根据自己的喜好修改)

```
{
  "extends": [
    "eslint:recommended",
    "plugin:@typescript-eslint/recommended",
    "prettier"
  ],
  "rules": {
    "@typescript-eslint/ban-ts-comment": "off",
    "@typescript-eslint/no-explicit-any": "off",
    "@typescript-eslint/no-namespace": "off",
    "@typescript-eslint/explicit-function-return-type": "off",
    "@typescript-eslint/explicit-module-boundary-types": "off"
  }
}
```

.gitignore (一个众所周知的文件 git不提交的文件/文件夹)

```
/node_modules/
/dist/

.DS_Store
```

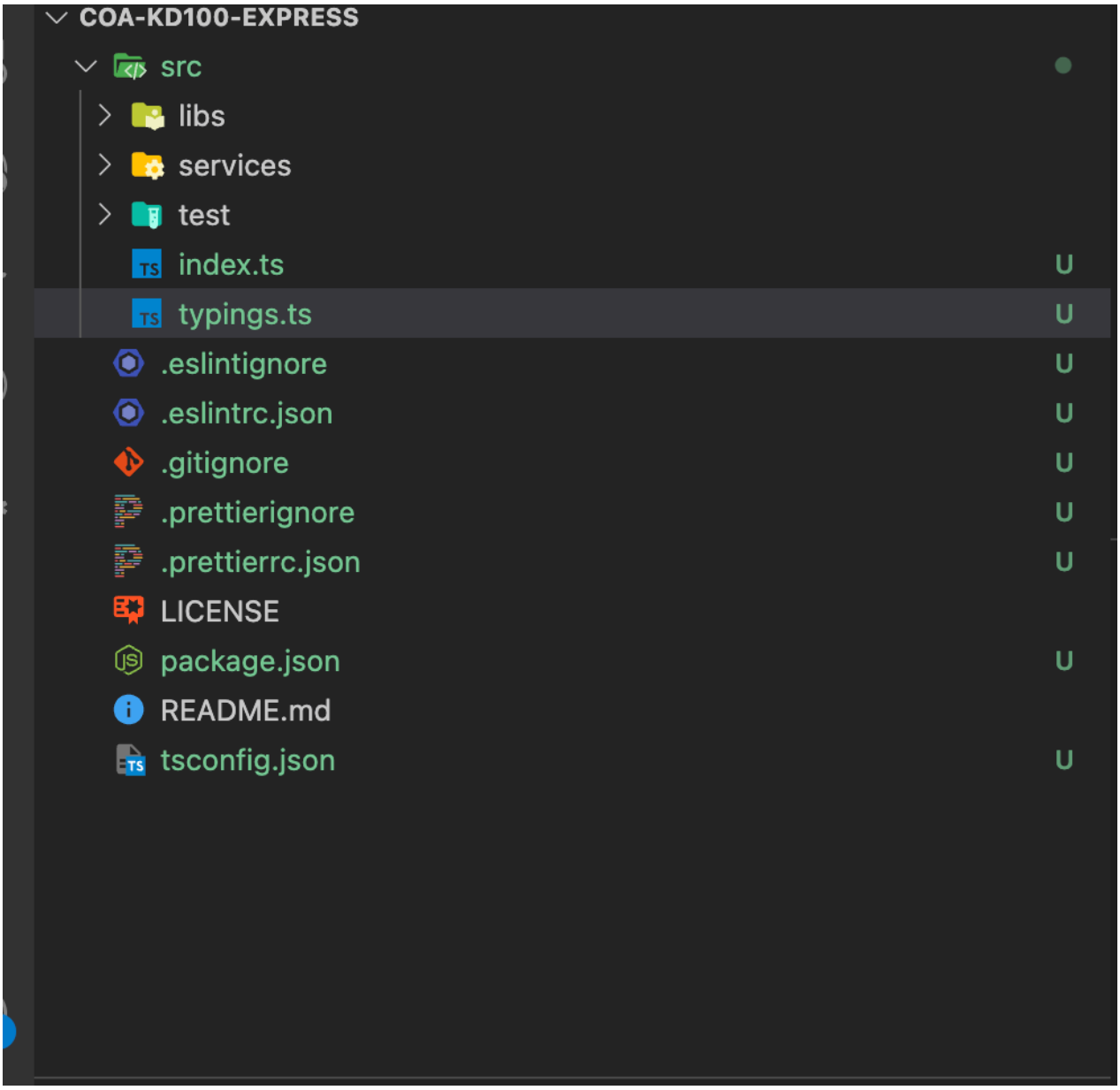
.prettierignore (prettier 不检查语法的文件夹)

```
dist
```

.prettierrc.json(prettier配置文件)

```
{
  "semi": false,
  "singleQuote": true
}
```

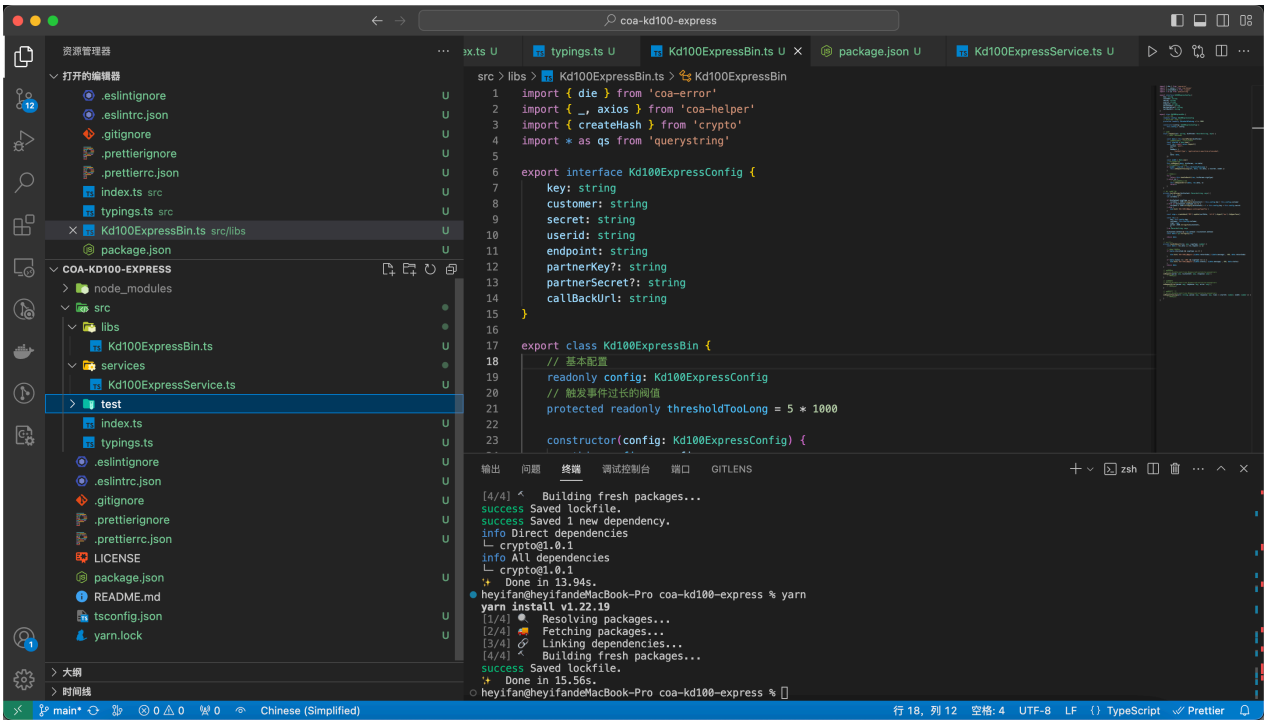
基本的就完了 我们来看一下项目现在的目录情况



6.开始写一些这个npm包内该有的代码

我这里以快递100对接为例子了 抛砖引玉一下😄

那么开始 run代码了.....



我简单的写了一个快递100的 电子面单下单、电子面单撤销、物流信息查询的api

libs\Kd100ExpressBin.ts

```ts
import { die } from 'coa-error'
import { _, axios } from 'coa-helper'
import { createHash } from 'crypto'
import * as qs from 'querystring'

export interface Kd100ExpressConfig {
    key: string
    customer: string
    secret: string
    userid: string
    endpoint: string
    partnerKey?: string
    partnerSecret?: string
    callBackUrl: string
}

export class Kd100ExpressBin {
    // 基本配置
    readonly config: Kd100ExpressConfig
    // 触发事件过长的阀值
    protected readonly thresholdTooLong = 5 * 1000

    constructor(config: Kd100ExpressConfig) {
        this.config = config
    }

    // 请求
```

```
    async request(url: string, bizParams: Record<string, any>) {
        // 组装参数并请求

        const data = this.buildParams(bizParams)
        //  请求并且记录开始、结束时间
        const startAt = Date.now()
        const res = await axios.request({
            method: 'post',
            url,
            headers: {
                'Content-Type': 'application/x-www-form-urlencoded',
            },
            data: data,
        })

        const endAt = Date.now()
        // 触发请求事件
        this.onRequest(data, bizParams, res.data)
        // 触发请求时间过长事件
        if (endAt - startAt > this.thresholdTooLong) {
            this.onRequestTooLong(url, data, res.data, { startAt, endAt })
        }

        // 处理结果
        try {
            return this.handleResult(res, bizParams.signType)
        } catch (e) {
            // 触发请求错误事件
            this.onRequestError(data, res.data, e)
            throw e
        }
    }

    // 构造请求参数
    private buildParams(bizContent: Record<string, any>) {
        const t = _.now()
        let sortData = ''

        if (bizContent.signType === 1) {
            sortData = JSON.stringify(bizContent) + this.config.key +
this.config.customer
        } else if (bizContent.signType === 2) {
            sortData = JSON.stringify(bizContent) + t + this.config.key +
this.config.secret
        } else {
            die.hint(`快递100系统提示:缺少signType字段`)
        }

        const sign = createHash('MD5').update(sortData, 'utf-
8').digest('hex').toUpperCase()

        const res = {
```

```
            key: this.config.key,
            customer: this.config.customer,
            sign: sign,
            param: JSON.stringify(bizContent),
            t,
        } as Record<string, any>

        bizContent.method && (res.method = bizContent.method)
        const data = qs.stringify(res)

        return data
    }

    // 结果验签
    private handleResult(res: any, signType: number) {
        const data = res.data || res.request || {}

        // 判断是否正确
        if (data.returnCode && signType === 2) {

            die.hint(`快递100系统提示:[${data.returnCode}] ${data.message}`, 400,
data.returnCode)
        }

        if (data.status !== '200' && signType === 1) {
            die.hint(`快递100系统提示:[${data.status}] ${data.message}`, 400,
data.status)
        }
        return data
    }

    // 请求记录
    // eslint-disable-next-line @typescript-eslint/no-unused-vars
    onRequest(param: any, bizContent: any, response: any) {
        // 重写方法
    }

    // 请求失败
    // eslint-disable-next-line @typescript-eslint/no-unused-vars
    onRequestError(param: any, response: any, error: any) {
        // 重写方法
    }

    // 请求时间过长
    // eslint-disable-next-line @typescript-eslint/no-unused-vars
    onRequestTooLong(url: string, param: any, response: any, time: { startAt:
number; endAt: number }) {
        // 重写方法
    }
}
```

services\Kd100ExpressService.ts

```
import { Kd100ExpressBin, Kd100ExpressConfig } from '../libs/Kd100ExpressBin'
import { Kd100Express } from '../typings'

export class Kd100ExpressService {
    protected bin: Kd100ExpressBin
    protected config: Kd100ExpressConfig

    constructor(bin: Kd100ExpressBin) {
        this.bin = bin
        this.config = this.bin.config
    }

    /**
     * 电子面单下单接口
     * @param params
     */
    async LabelOrder(params: any) {
        return await this.bin.request('https://api.kuaidi100.com/label/order',
params)
    }
    /**
     * 电子面单取消
     * @param params
     */

    async Eorderapi(params: any) {
        return await this.bin.request('https://poll.kuaidi100.com/eorderapi.do',
params)
    }

    /**
     * 查询接口服务
     * @param params
     */

    async PollQuery(params: Kd100Express.PollQueryReq):
Promise<Kd100Express.PollQueryRes> {
        return await
this.bin.request('https://poll.kuaidi100.com/poll/query.do', params)
    }
}
```

index.ts

```
export * from './libs/Kd100ExpressBin'
export * from './services/Kd100ExpressService'
```

typings.ts

```
export declare namespace Kd100Express {

    interface PollQueryReq {
        com: string;
        num: string;
        resultv2: string;
        phone: string;
        signType: number;
        from?: string;
        to?: string;
        show?: string;
        order: string;
    }

    interface PollQueryRes {
        message: string;
        state: string;
        status: string;
        condition: string;
        ischeck: string;
        com: string;
        nu: string;
        data: any;
    }

    interface LabelOrderReq {
        method: string
        signType: number
        payTpe: string
        expType: string
        needSubscribe: string
        pollCallBackUrl: string
        printType: string
        customParam: any
        partnerKey?: string
        partnerId: string
        kuaidicom: string
        recMan: any
        sendMan: any
        cargo: string
        count: string
        tempId: string
        partnerSecret?: string
```

```
    code?: string
    partnerName?: string
    net?: string
    checkMan?: string
    tbNet?: string
    weight?: string
    remark?: string
    siid?: string
    direction?: string
    childTempId?: string
    backTempId?: string
    valinsPay?: string
    collection?: string
    needChild?: string
    needBack?: string
    backSign?: string
    orderId?: string
    reorder?: string
    callBackUrl?: string
    salt?: string
    resultv2?: string
    needDesensitization?: string
    needLogo?: string
    thirdOrderId?: string
    oaid?: string
    caid?: string
    thirdTemplateURL?: string
    thirdCustomTemplateUrl?: string
    needOcr?: string
    ocrInclude?: string
    height?: string
    width?: string
}

interface LabelOrderRes {
    taskId: string
    kuaidinum: string
    childNum: string
    returnNum: string
    label: string
    bulkpen: string
    orgCode: string
    orgName: string
    destCode: string
    destName: string
    orgSortingCode: string
    orgSortingName: string
    destSortingCode: string
    destSortingName: string
    orgExtra: string
    destExtra: string
    pkgCode: string
```

```
        pkgName: string
        road: string
        qrCode: string
        kdComOrderNum: string
        expressCode: string
        expressName: string
    }

    interface EorderapiReq {
        partnerId: string
        kuaidicom: string
        kuaidinum: string
        orderId: string
        partnerKey?: string
        partnerName?: string
        net?: string
        code?: string
        reason?: string
    }
    interface EorderapiRes {
        returnCode: string
        result: boolean
        message: string
    }
}
```

并且简单的测试了一下三个api接口都可以成功调用 并且没有问题

7.写一下 GitHub的workflows

根目录创建文件夹
.github/worlflows

build.yml

```yaml
name: Build

on:
  push:
    branches:
      - '**'

jobs:
  build-check:
    name: Build check
```

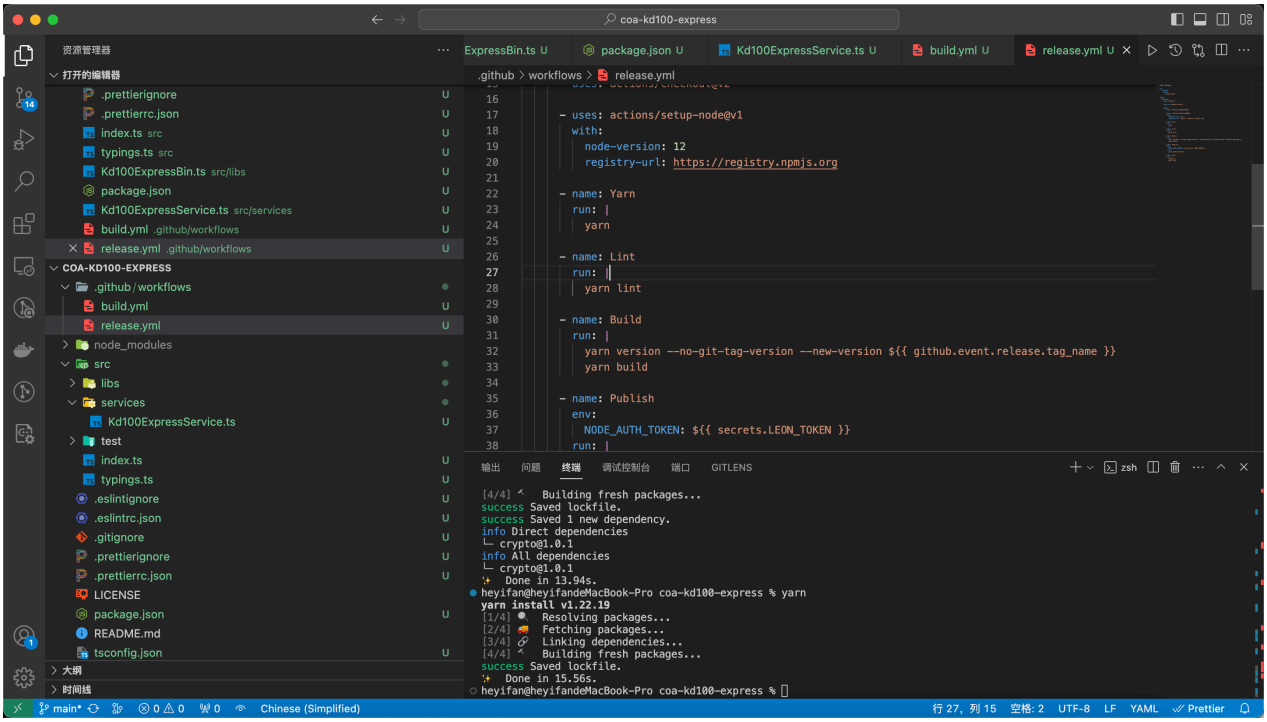
```
    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@v2

      - run: yarn
      - run: yarn lint
      - run: yarn build
```

release.yml

```
name: Release

on:
  release:
    types:
      - published

jobs:
  publish:
    name: Publish

    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@v2

      - uses: actions/setup-node@v1
        with:
          node-version: 12
          registry-url: https://registry.npmjs.org

      - name: Yarn
        run: |
          yarn

      - name: Lint
        run: |
          yarn lint

      - name: Build
        run: |
          yarn version --no-git-tag-version --new-version ${{
github.event.release.tag_name }}
          yarn build

      - name: Publish
        env:
```

```
        NODE_AUTH_TOKEN: ${{ secrets.LEON_TOKEN }}
      run: |
        yarn publish dist

    - name: Sync
      run: |
        sleep 5s
        yarn sync
```

所以目前的结构目录 是这样的

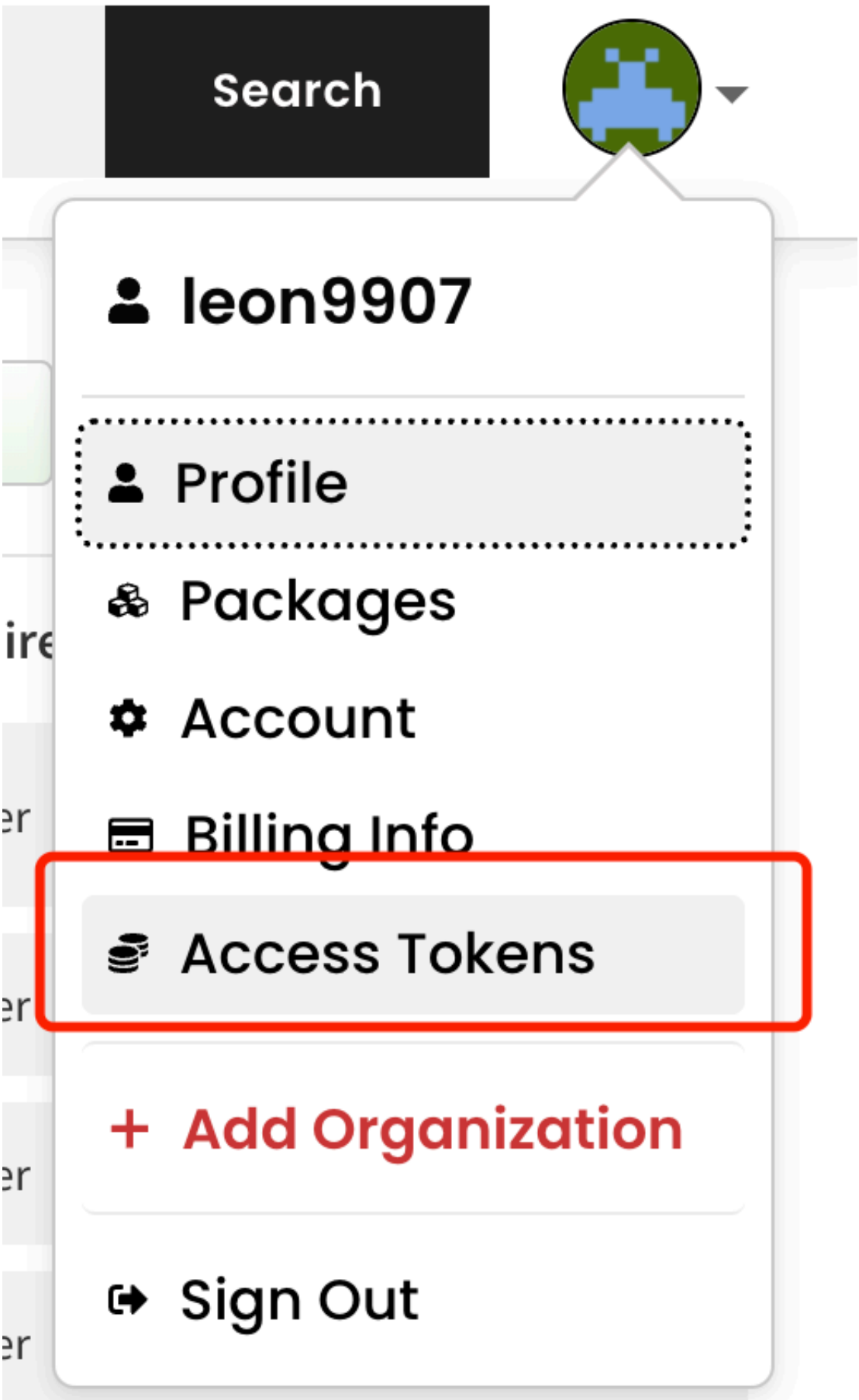


8.申请一个npm帐号

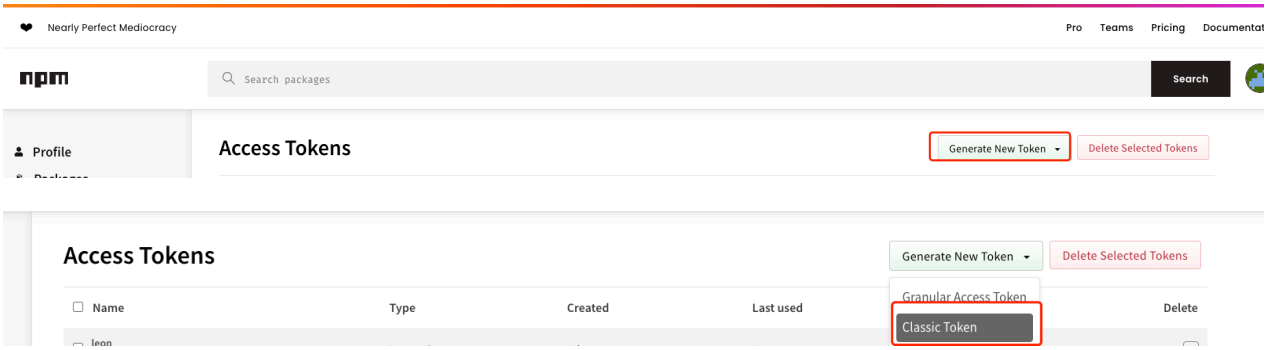
登录你的npm帐号

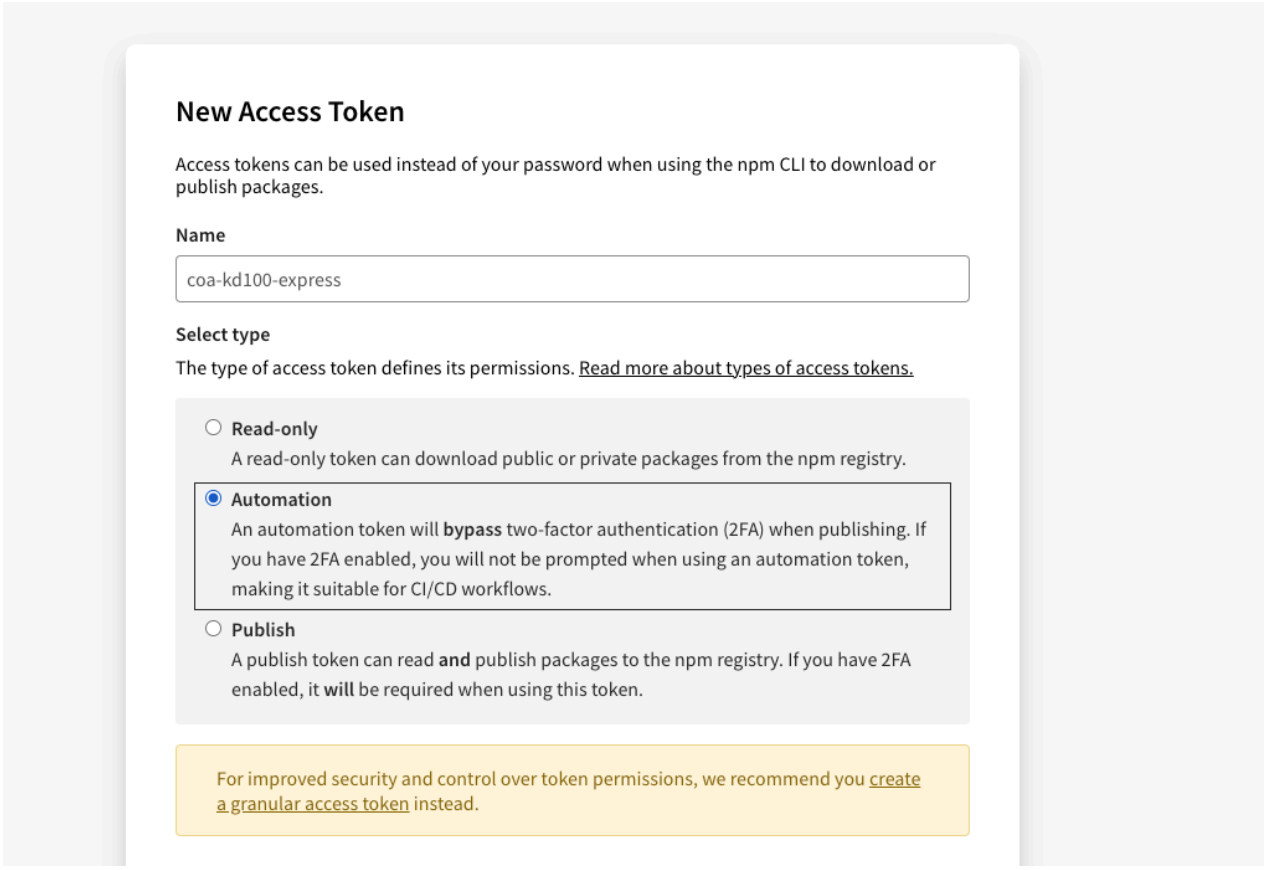
https://www.npmjs.com/

进入


Search
leon9907
Profile
Packages
Account
Billing Info
Access Tokens
+ Add Organization
Sign Out

新建一个token



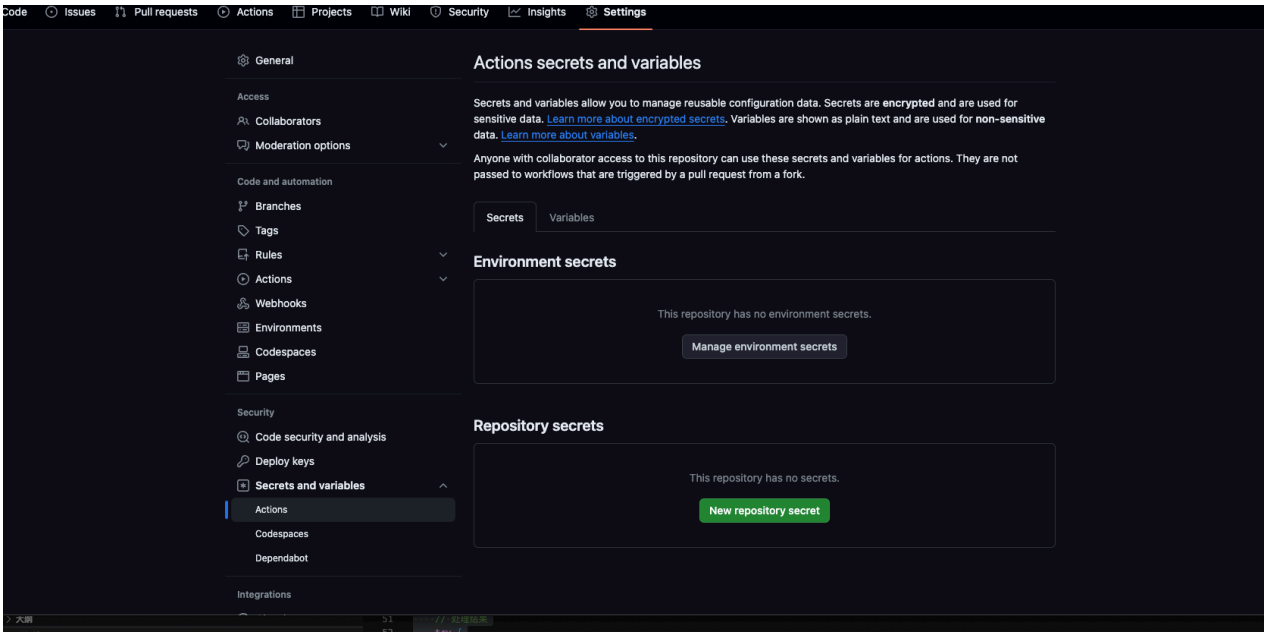


然后复制你新生成好的令牌

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

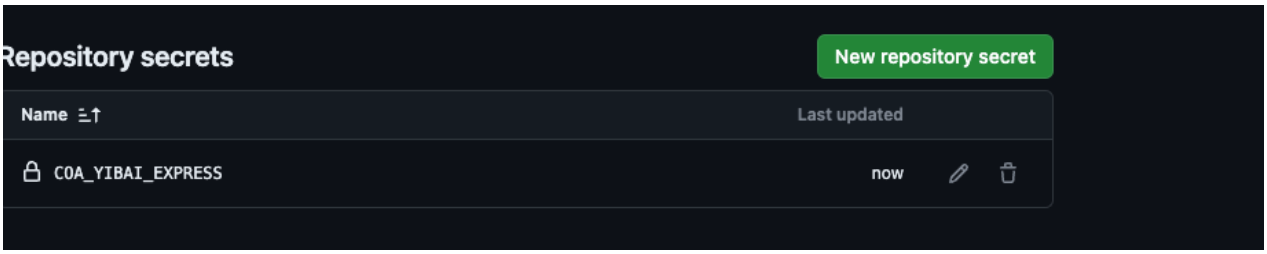
9.把令牌放到GitHub上

打开项目设置

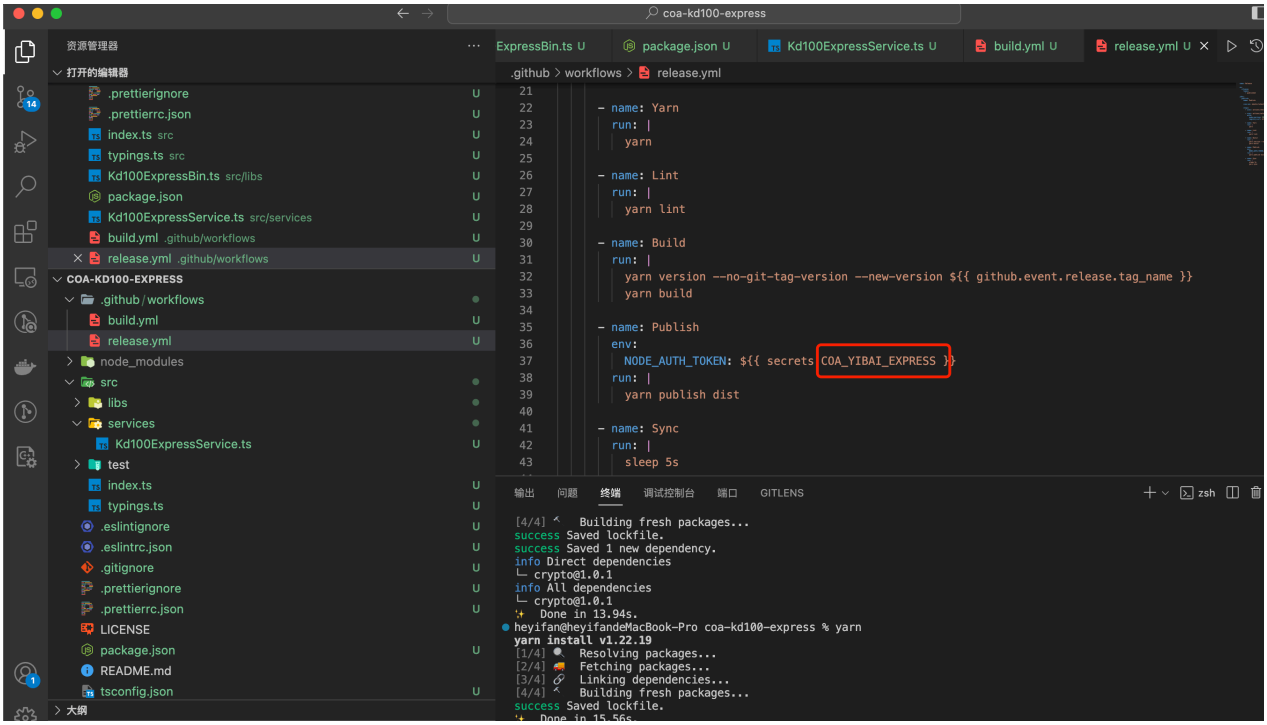


创建Repository secrets

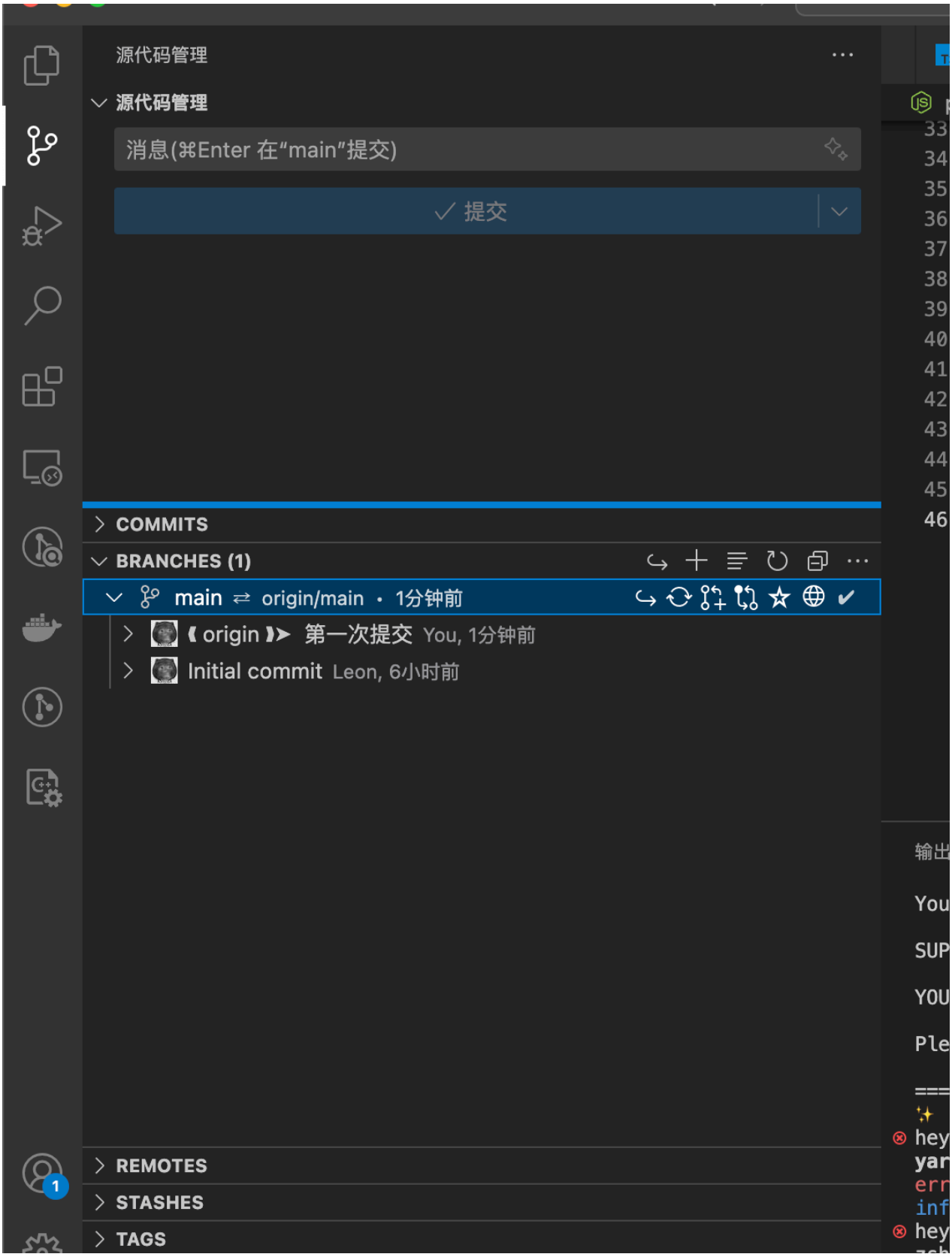
把复制好的令牌粘贴进去 写一个名字



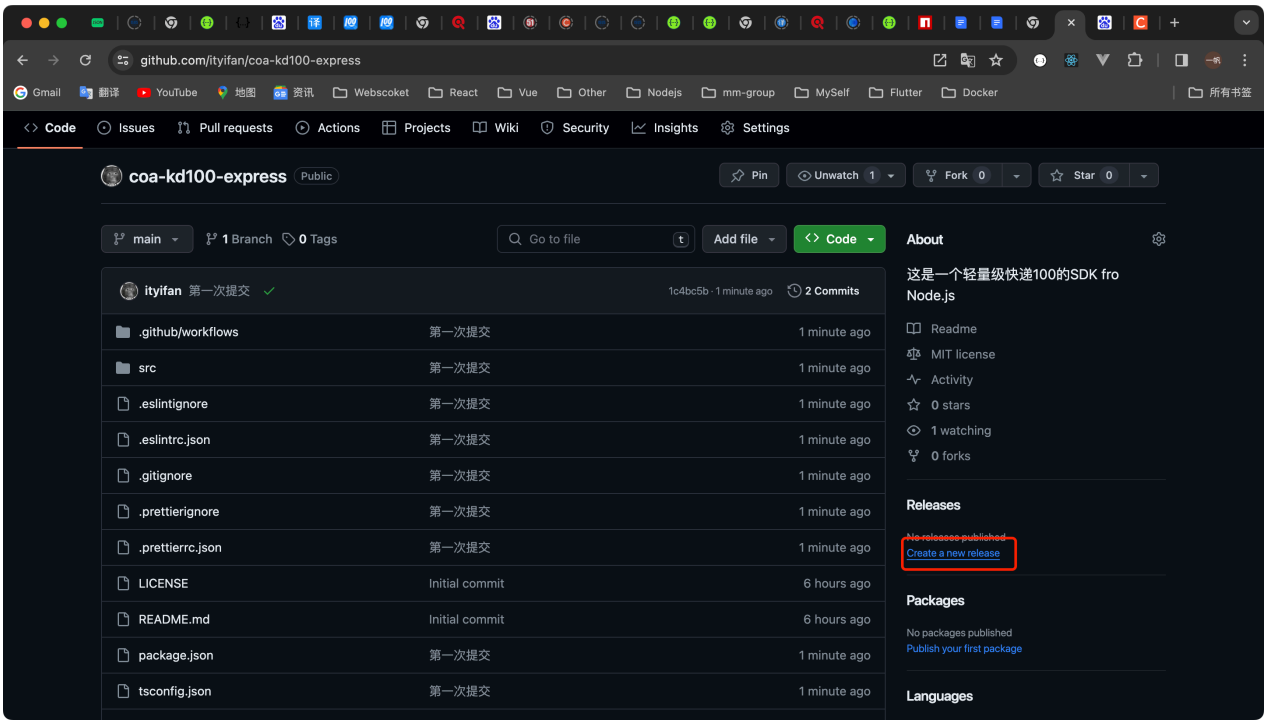
再将 密钥名称 复制下来 将release.yml的密钥名称 更改为你的密钥名称

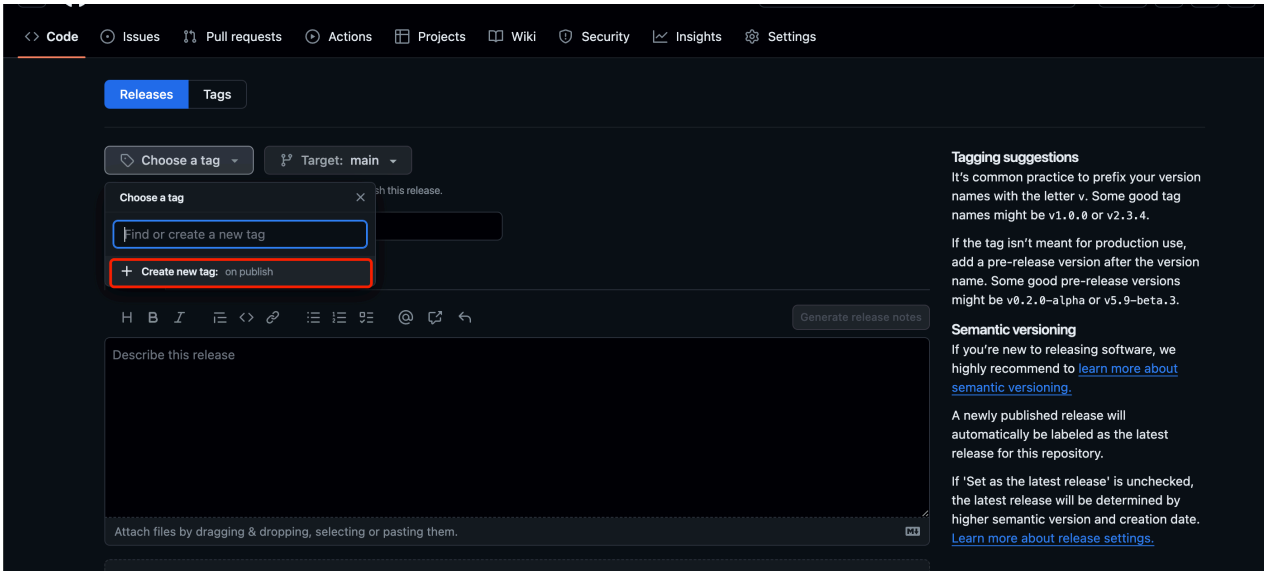


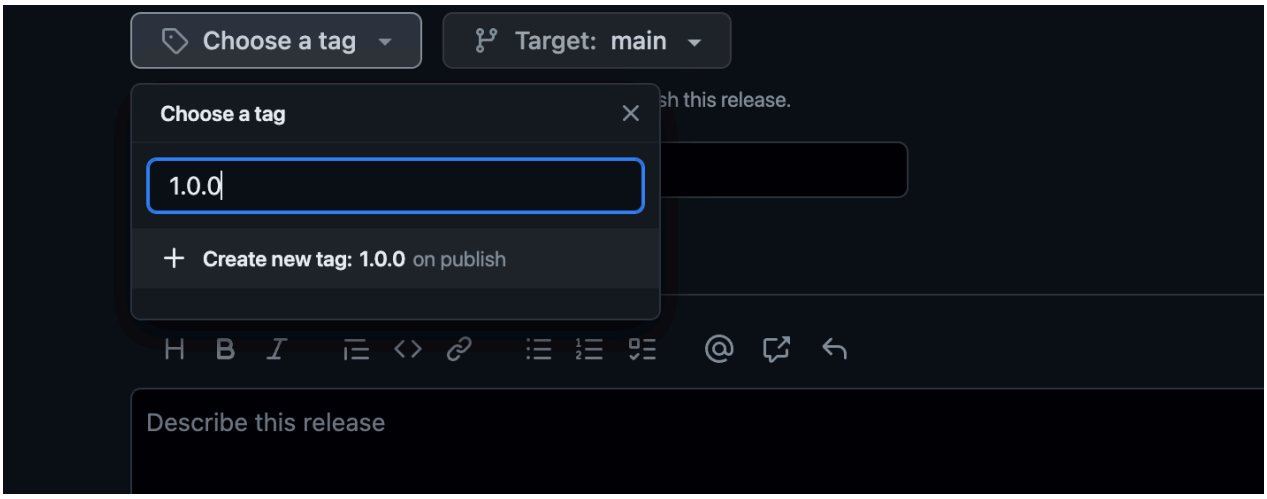
10.提交代码到github

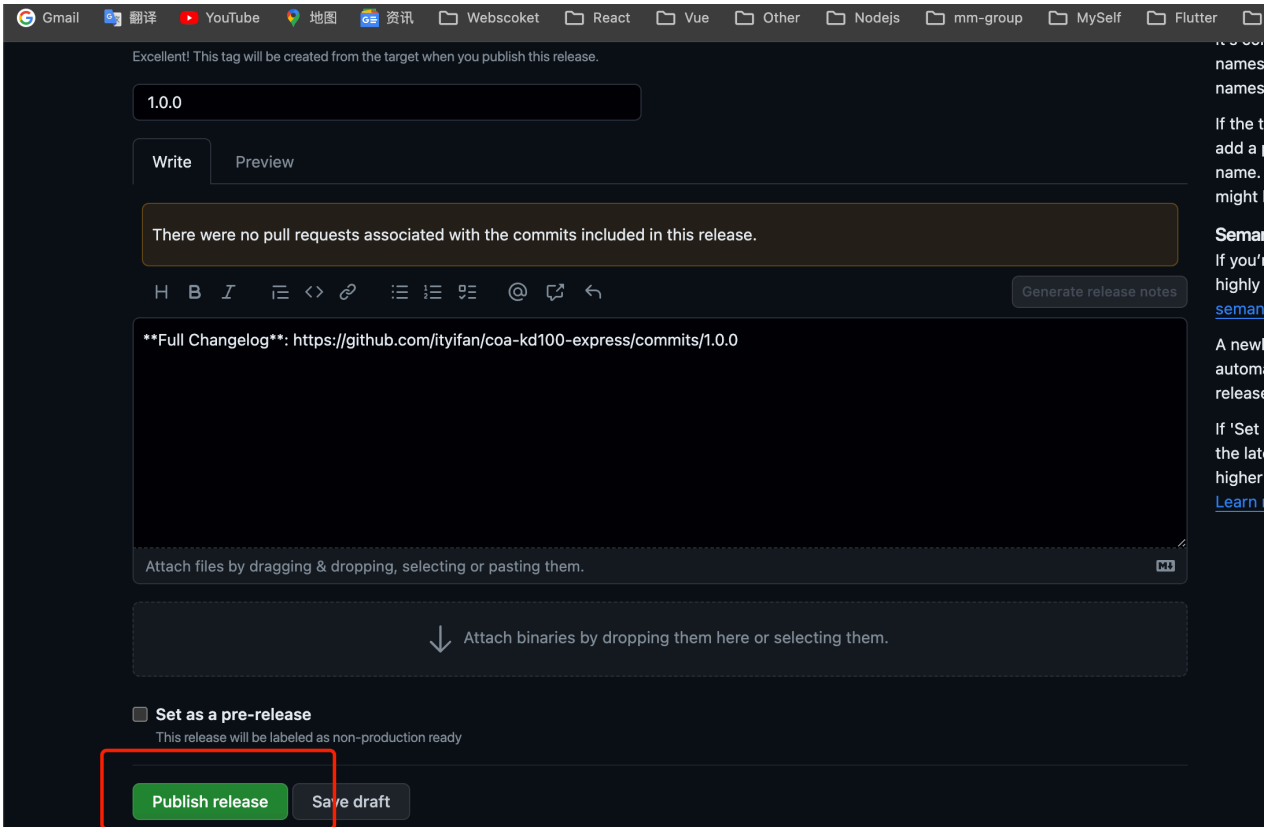


11.来到github上发一个新版本

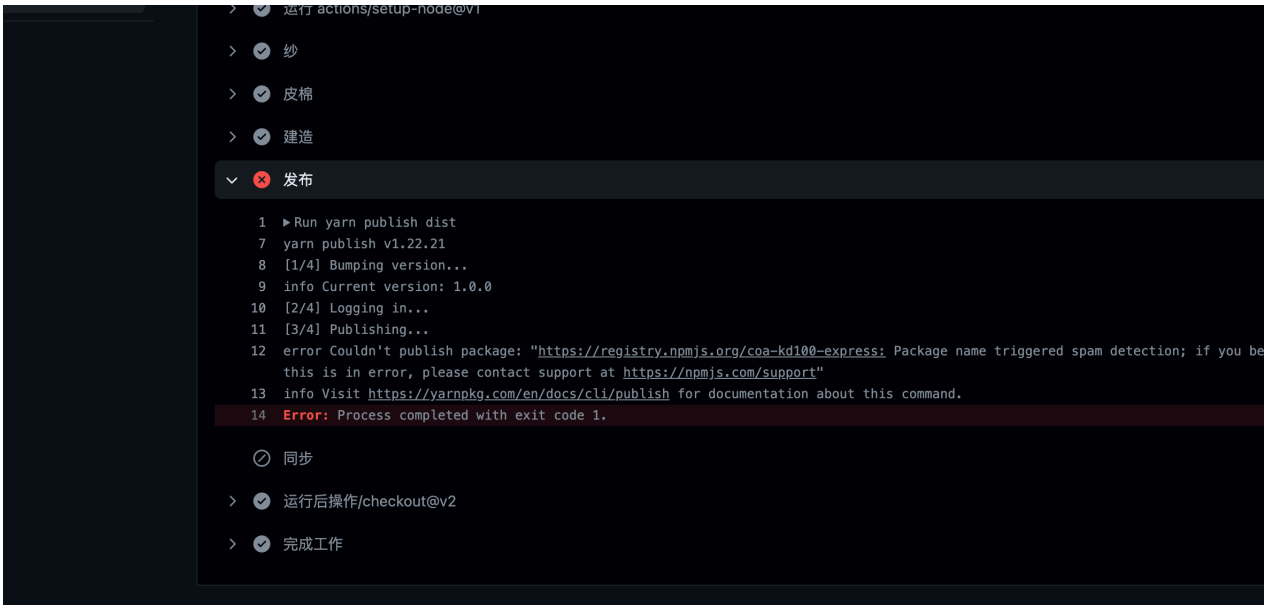

github.com/ityifan/coa-kd100-express
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
coa-kd100-express
Public
Pin
Unwatch 1
Fork 0
Star 0
main
1 Branch
0 Tags
Go to file
Add file
Code
ityifan 第一次提交
1c4bc5b · 1 minute ago
2 Commits
.github/workflows 第一次提交 1 minute ago
src 第一次提交 1 minute ago
.eslintignore 第一次提交 1 minute ago
.eslintrc.json 第一次提交 1 minute ago
.gitignore 第一次提交 1 minute ago
.prettierignore 第一次提交 1 minute ago
.prettierrc.json 第一次提交 1 minute ago
LICENSE Initial commit 6 hours ago
README.md Initial commit 6 hours ago
package.json 第一次提交 1 minute ago
tsconfig.json 第一次提交 1 minute ago
About
这是一个轻量级快递100的SDK fro
Node.js
Readme
MIT license
Activity
0 stars
1 watching
0 forks
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
Languages


Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Releases
Tags
Choose a tag
Target: main
Choose a tag
Find or create a new tag
Create new tag: on publish
Generate release notes
Describe this release
Attach files by dragging & dropping, selecting or pasting them.
Tagging suggestions
It's common practice to prefix your version names with the letter v. Some good tag names might be v1.0.0 or v2.3.4.
If the tag isn't meant for production use, add a pre-release version after the version name. Some good pre-release versions might be v0.2.0-alpha or v5.9-beta.3.
Semantic versioning
If you're new to releasing software, we highly recommend to learn more about semantic versioning.
A newly published release will automatically be labeled as the latest release for this repository.
If 'Set as the latest release' is unchecked, the latest release will be determined by higher semantic version and creation date.
Learn more about release settings.


Choose a tag
Target: main
Choose a tag
1.0.0
Create new tag: 1.0.0 on publish
Describe this release


Gmail
翻译
YouTube
地图
资讯
Webscoket
React
Vue
Other
Nodejs
mm-group
MySelf
Flutter
Excellent! This tag will be created from the target when you publish this release.
1.0.0
Write
Preview
There were no pull requests associated with the commits included in this release.
Generate release notes
Full Changelog: https://github.com/ityifan/coa-kd100-express/commits/1.0.0
Attach files by dragging & dropping, selecting or pasting them.
Attach binaries by dropping them here or selecting them.
Set as a pre-release
This release will be labeled as non-production ready
Publish release
Save draft

发现一个错误



我们更改一下release.yml的nodejs版本号 改为17 然后重新提交代码 重新发布一个版本号为1.0.1

12.首次发布npm包

本地没有登录npm 帐号的要先登录一下哦

npm publish

```
heyifan@heyifandeMacBook-Pro coa-kd100-express % npm publish
npm notice
npm notice 📦  coa-kd100-express@1.0.0
npm notice === Tarball Contents ===
npm notice 4B    .eslintignore
npm notice 396B  .eslintrc.json
npm notice 236B  .github/workflows/build.yml
npm notice 768B  .github/workflows/release.yml
npm notice 4B    .prettierignore
npm notice 43B   .prettierrc.json
npm notice 1.1kB LICENSE
npm notice 69B   README.md
npm notice 1.3kB package.json
npm notice 87B   src/index.ts
npm notice 3.7kB src/libs/Kd100ExpressBin.ts
npm notice 989B  src/services/Kd100ExpressService.ts
npm notice 2.8kB src/typings.ts
npm notice 169B  tsconfig.json
npm notice === Tarball Details ===
npm notice name:          coa-kd100-express
npm notice version:       1.0.0
npm notice filename:      coa-kd100-express-1.0.0.tgz
npm notice package size:  4.4 kB
npm notice unpacked size: 11.6 kB
npm notice shasum:        7c82aae6fa9b3064cee401bc72141aae50263444
npm notice integrity:     sha512-suNqWkHKXdZQY[...]FDfeSttZnoWGA==
npm notice total files:   14
npm notice
npm notice Publishing to https://registry.npmjs.org/
npm notice Open https://www.npmjs.com/login/b33a443e-aa98-4cb4-9960-b31d5ad6e281 to use your security key for authentication
or enter OTP from your authenticator app
```

遇到一个错误 说我们项目名称奇怪 那么我们把coa-kd100-express改为coa-kd-express后再次提交

```
npm ERR! code E403
npm ERR! 403 403 Forbidden - PUT https://registry.npmjs.org/coa-kd100-express - Package name triggered spam detection; if yo
u believe this is in error, please contact support at https://npmjs.com/support
npm ERR! 403 In most cases, you or one of your dependencies are requesting
npm ERR! 403 a package version that is forbidden by your security policy, or
npm ERR! 403 on a server you do not have access to.
```

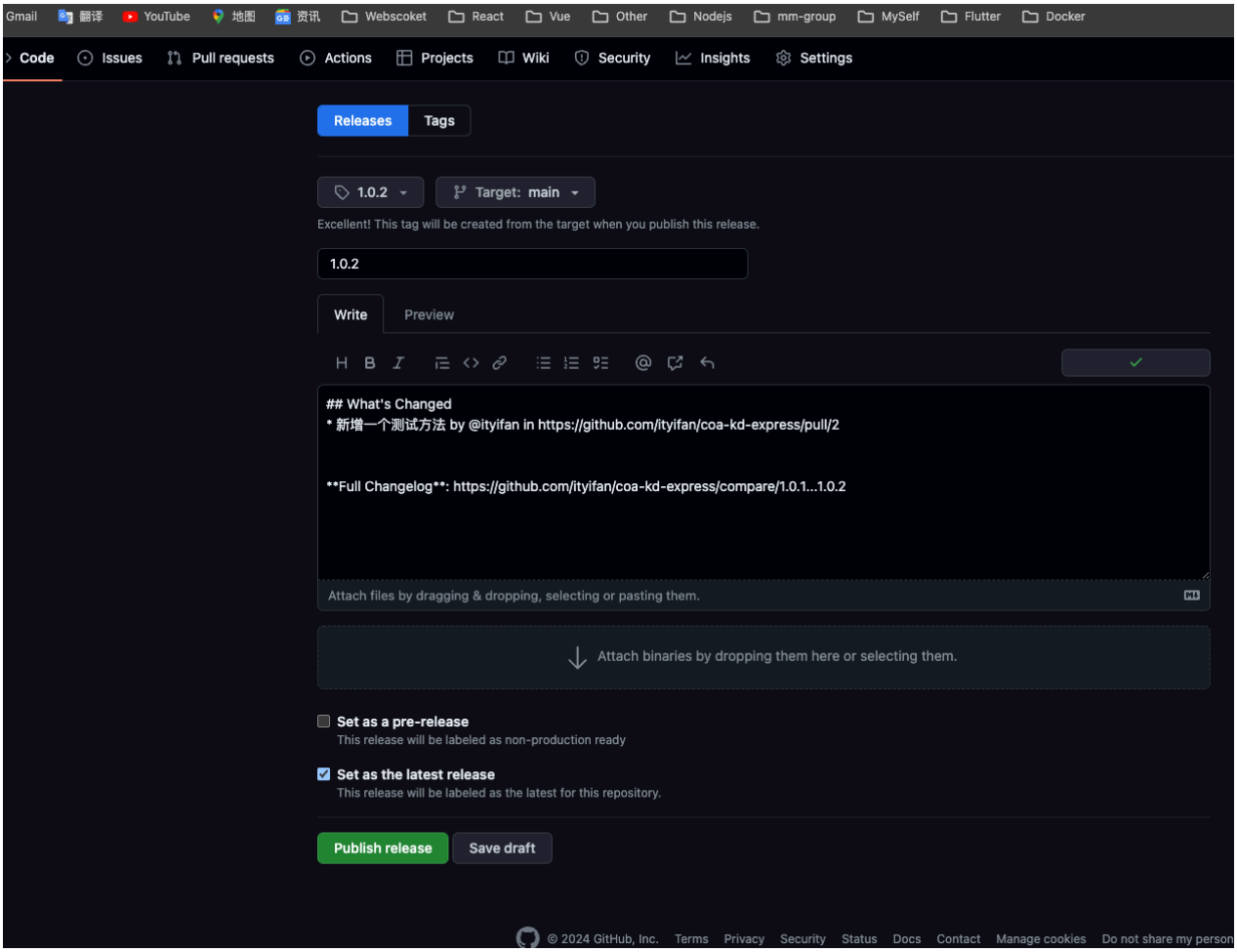
输入我们的令牌

```
Enter OTP: 558925414707 7344
+ coa-kd-express@1.0.0
heyifan@heyifandeMacBook-Pro coa-kd100-express %
```

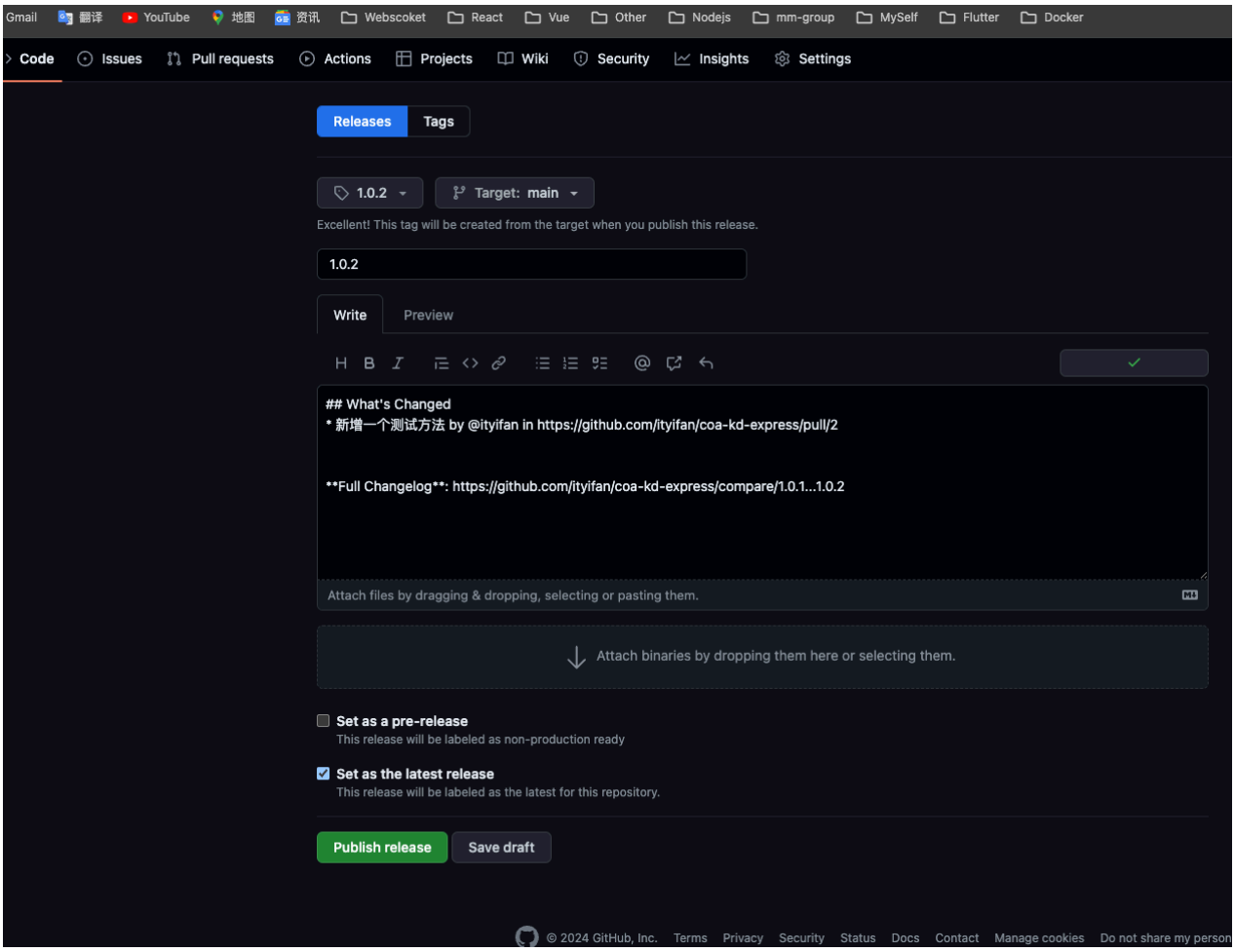
发布成功

13.测试使用发布版本的方式 自动同步npm

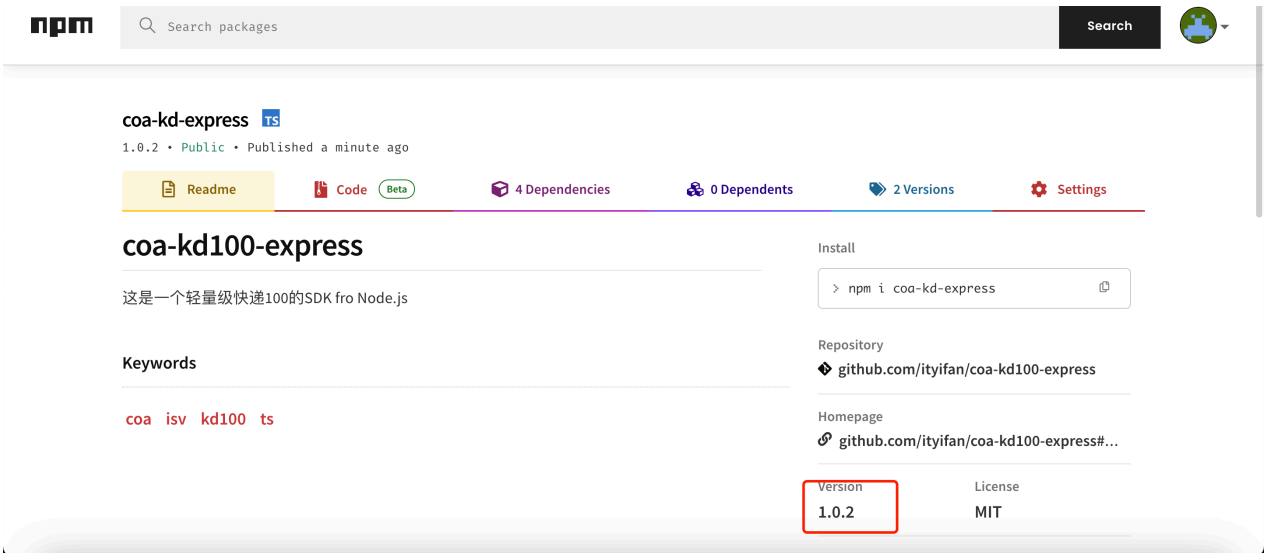
我们随意改动一些代码 并且发布一个新版本



发布



回头查看npm的版本

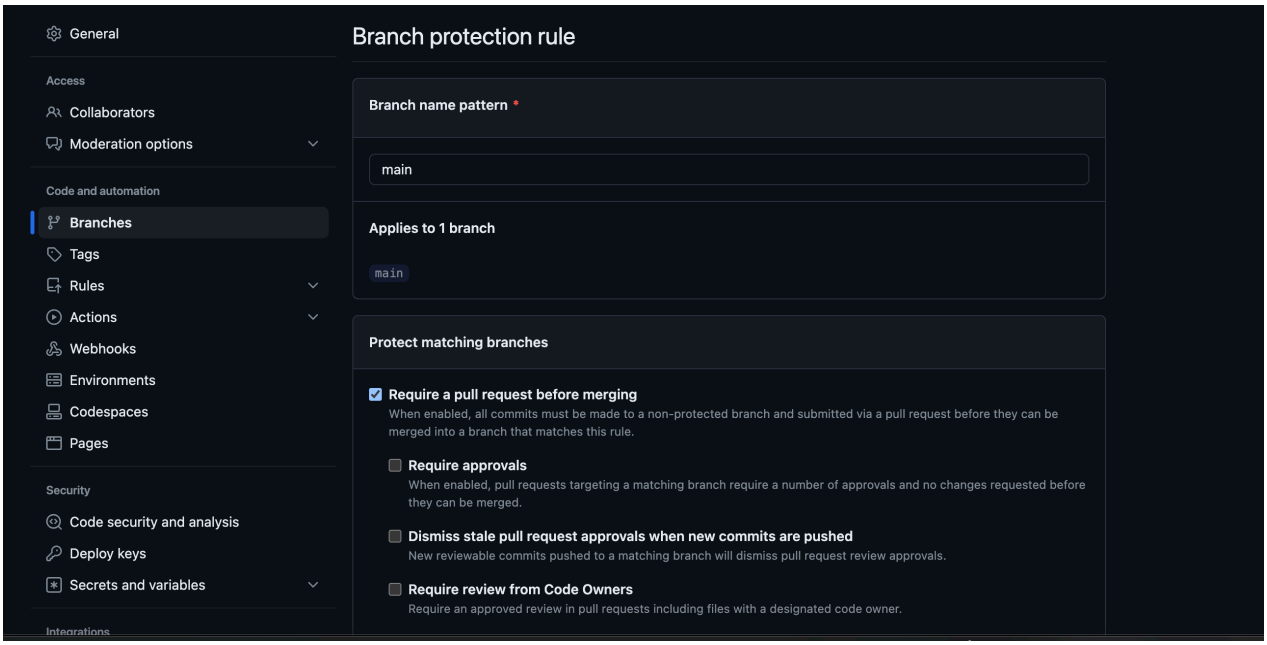


此致 艺术已成

总结一下

这次 这个自动发布的流程还是有许多可避免的地方
比如 项目名称不要命名太奇怪了 不要包含数字吧
不然package.json的一些配置文件 更改起来也非常的麻烦

最后不要忘了在github上把main分支设置为保护分之哦 要不然别人也可以修改你代码 不需要审批



如果你想写一些 sdk使用方法 不妨在项目中的readme.md文件中 尝试写一些md格式的使用说明 并且发布一个版本