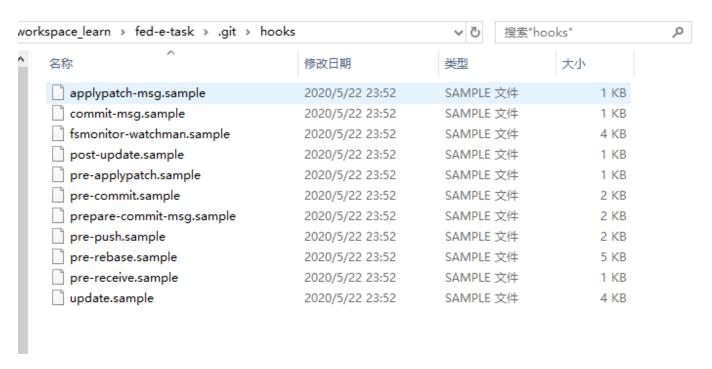
# 第十六节 GitHooks

#### 1.定义

#### 2.配合eslint操作

# 1.定义

- 通过Git hooks在代码提交前强制lint
- git hook 也称之为git 钩子,每个钩子都对应一个任务
- 通过shell脚本可以编写钩子任务触发时要具体执行的操作
- 在一个git项目中有hooks那么多的钩子任务



可以修改pre-commit文件,并添加shell脚本,在commit的时候触发该任务

# 2.配合eslint操作

在代码提交之前对代码进行lint操作 但是前端开发者不擅长使用shell Husky可以实现git hooks的使用需求

1.安装husyk

1 yarn add husky -D

### 2.修改package.json

```
1 {
"name": "04-06-githook",
3
    "version": "1.0.0",
4 "main": "index.js",
5 "license": "MIT",
 6 "devDependencies": {
      "eslint": "^7.2.0",
 7
      "eslint-config-standard": "^14.1.1",
8
      "eslint-plugin-import": "^2.21.2",
9
      "eslint-plugin-node": "^11.1.0",
10
      "eslint-plugin-promise": "^4.2.1",
11
12
      "eslint-plugin-standard": "^4.0.1"
13
    },
14 "scripts": {
      "link": "yarn eslint ."
15
16
    },
17 "husky": {
     "hooks": {
18
        "pre-commit": "yarn run link"
19
20
     }
21 }
22 }
```

当代码进行commit的时候就会检查lint

但是我们需要在检查之后做一些后续的操作,例如格式化操作,然后在添加到暂存区等操作这时候需要lint-staged,配合husky进行其他的功能

3.安装lint-staged

```
1 yarn add lint-staged -D
```

### 4.修改package.json

```
1 {
2  "name": "04-06-githook",
3  "version": "1.0.0",
```

```
4 "main": "index.js",
5 "license": "MIT",
   "devDependencies": {
6
7
      "eslint": "^7.2.0",
      "eslint-config-standard": "^14.1.1",
8
      "eslint-plugin-import": "^2.21.2",
9
      "eslint-plugin-node": "^11.1.0",
10
11
      "eslint-plugin-promise": "^4.2.1",
      "eslint-plugin-standard": "^4.0.1",
12
      "lint-staged": "^10.2.11"
13
14
    },
15
    "scripts": {
      "link": "yarn eslint .",
16
      "precommit": "lint-staged"
17
18
    },
   "husky": {
19
20
      "hooks": {
        "pre-commit": "yarn run precommit"
21
22
     }
23
    },
24
   "lint-staged": {
25
      "*.js": [ // 对某类的文件进行操作,而且是一个数组
26
       "eslint",
27
      "git add"
28
    ]
29 }
30 }
```