

main

...

gin-vue-admin / server / utils / breakpoint_continue.go / <> Jump to

piexlmax 增加自动化代码匹配进入业务库功能 (#1249) ... ✓

History

6 contributors



112 lines (99 sloc) | 3.03 KB

...

```
1 package utils
2
3 import (
4     "errors"
5     "os"
6     "strconv"
7     "strings"
8 )
9
10 // 前端传来文件片与当前片为什么文件的第几片
11 // 后端拿到以后比较次分片是否上传 或者是否为不完全片
12 // 前端发送每片多大
13 // 前端告知是否为最后一片且是否完成
14
15 const (
16     breakpointDir = "./breakpointDir/"
17     finishDir     = "./fileDir/"
18 )
19
20 //@author: [piexlmax](https://github.com/piexlmax)
21 //@function: BreakPointContinue
22 //@description: 断点续传
23 //@param: content []byte, fileName string, contentNumber int, contentTotal int, fileMd5 string
24 //@return: error, string
25
26 func BreakPointContinue(content []byte, fileName string, contentNumber int, contentTotal int, fileMd5 string) error {
27     path := breakpointDir + fileMd5 + "/"
28     err := os.MkdirAll(path, os.ModePerm)
```

```
29     if err != nil {
30         return path, err
31     }
32     pathC, err := makeFileContent(content, fileName, path, contentNumber)
33     return pathC, err
34 }
35
36 //@author: [piexlmax](https://github.com/piexlmax)
37 //@function: CheckMd5
38 //@description: 检查Md5
39 //@param: content []byte, chunkMd5 string
40 //@return: CanUpload bool
41
42 func CheckMd5(content []byte, chunkMd5 string) (CanUpload bool) {
43     fileMd5 := MD5V(content)
44     if fileMd5 == chunkMd5 {
45         return true // 可以继续上传
46     } else {
47         return false // 切片不完整, 废弃
48     }
49 }
50
51 //@author: [piexlmax](https://github.com/piexlmax)
52 //@function: makeFileContent
53 //@description: 创建切片内容
54 //@param: content []byte, fileName string, FileDir string, contentNumber int
55 //@return: string, error
56
57 func makeFileContent(content []byte, fileName string, FileDir string, contentNumber int) (string,
58     if strings.Index(fileName, ".") > -1 || strings.Index(FileDir, ".") > -1 {
59         return "", errors.New("文件名或路径不合法")
60     }
61     path := FileDir + fileName + "_" + strconv.Itoa(contentNumber)
62     f, err := os.Create(path)
63     if err != nil {
64         return path, err
65     } else {
66         _, err = f.Write(content)
67         if err != nil {
68             return path, err
69         }
70     }
71     defer f.Close()
72     return path, nil
73 }
74
75 //@author: [piexlmax](https://github.com/piexlmax)
76 //@function: makeFileContent
77 //@description: 创建切片文件
```

```
78 // @param: fileName string, FileMd5 string
79 // @return: error, string
80
81 func MakeFile(fileName string, FileMd5 string) (string, error) {
82     rd, err := os.ReadDir(breakpointDir + FileMd5)
83     if err != nil {
84         return finishDir + fileName, err
85     }
86     _ = os.MkdirAll(finishDir, os.ModePerm)
87     fd, err := os.OpenFile(finishDir+fileName, os.O_RDWR|os.O_CREATE|os.O_APPEND, 0o644)
88     if err != nil {
89         return finishDir + fileName, err
90     }
91     defer fd.Close()
92     for k := range rd {
93         content, _ := os.ReadFile(breakpointDir + FileMd5 + "/" + fileName + "_" + strconv.Itoa(k))
94         _, err = fd.Write(content)
95         if err != nil {
96             _ = os.Remove(finishDir + fileName)
97             return finishDir + fileName, err
98         }
99     }
100     return finishDir + fileName, nil
101 }
102
103 // @author: [piexlmax](https://github.com/piexlmax)
104 // @function: RemoveChunk
105 // @description: 移除切片
106 // @param: FileMd5 string
107 // @return: error
108
109 func RemoveChunk(FileMd5 string) error {
110     err := os.RemoveAll(breakpointDir + FileMd5)
111     return err
112 }
```