Package blit_cli

import "github.com/ruymanbr/blit/pkg/blit_cli"

Overview
Index

Overview ▾

Index ▾

Package files

blit_cli.go

func **ByteToReadableSize**

```
func ByteToReadableSize(bigNum int64) string
```

ByteToReadableSize transform a byte size into human readable form sizes (kb, Mb, Gb, Tb, Pb). Takes 1 argument and returns a HR string for size

1: bigNum int64 (size in bytes)

Returns:

```
1: string                              (size in human readable form: Pb, Tb, Gb, etc)
```

func **CleanData**

```
func CleanData(rawData [][]string) ([][]string, []string)
```

CleanData removes first column for [][]string matrix. Ideally the format returned by EncapData() function in second position

1: rawData [][]string (Raw data from Encap(), including dirs conditional y/n in first colum

Returns:

```
1: [][]string                    (Same matrix without first colum)
2: []string                      (Folder y/n confirmation string obtained from argument to this
function)
```

func **DirSize**

```
func DirSize(path string) (int64, error)
```

DirSize obtains Dir size recursively

1: path string (Path where files are located)

Returns:

```
1: int64                              (Sum of total file sizes in given path)
```

func **EncapData**

```
func EncapData(fileInfo []fs.FileInfo, path string) ([][]string, error, int64)
```

EncapData extracts data from a []fs.FileInfo dataset in a given path

1: fileInfo []fs.FileInfo (obtained from os.Open File -> Readdir()) 2: path string (Path where files are located)

Returns:

```
2: [][]string                    (File info -as in [n_files]{IsDir, LastM, FName, FSize_HR_Format}  )
3: error                              (Returns this error when trying to obtain
os.Stat(/path/to/file/name/) for each file
3: int64                              (Sum of total file sizes in given path)
```

func **EncapSizes**

```
func EncapSizes(fileInfo []fs.FileInfo) [][]int
```

EncapSizes returns a [][]int slice with data from a []fs.FileInfo dataset in a given path

1: fileInfo []fs.FileInfo (obtained from os.Open File -> Readdir())

Returns:

```
1: [][]int                              (File sizes matrix)
```

func FastSwitchSli

```
func FastSwitchSli(strUnordered [][]string, orderedSli [][]int, origPos int) [][]string
```

FastSwitchSli sorts a [n_files][5]string dataset obtained from <- GetPathInfo() <- EncapData().

Takes 3 arguments: 1: [][]string (Unordered string matrix with folder files data) 2: [][]int (Sorted slice with file size and original position in primitive raw data slice) 3: int (original position of files, in ordered fileSize slice's rows. Basically its col_index)

Returns:

```
1: [][]string                      (Fully formatted array with file data. Ordered by size. later derived
to RenderData() function for CLI display purpose)
```

func FileSizeSort

```
func FileSizeSort(sli [][]int, sizePos int)
```

FileSizeSort sorts a [][]int slice matrix of file data, by size.

Takes 2 arguments:

1: sli [][]int (size matrix with size and original position as column values in every row) 2: sizePort int (as first argument (Bigger first, smaller last) by calling Swap() function

```
<No return>
```

func GetPath

```
func GetPath(args []string) (string, bool)
```

GetPath extracts path from CLI argument, if not given it returns current directory path

Takes 1 argument: 1: args []string (os.Args)

Returns:

```
1: string                               (argument path or current working directory)
2: bool                                 (Yes for argument with path from CLI call to blit program)
```

func GetPathInfo

```
func GetPathInfo(root string) ([]fs.FileInfo, error)
```

GetPathInfo extracts info from a given path.

Takes 1 argument: 1: root string (Path to extract info from)

Returns (same as EncapData() :

```
1: []fs.FileInfo                    (slice with info from files and folders)
2: error                                (not nilfor failing to open or failing reading it)
```

func HandlePath

```
func HandlePath(path string) ([]fs.FileInfo, string, error)
```

HandlePath handles a given path calling functions in package blit_cli

Takes 1 argument: 1: path string (what system path to be listed)

Returns:

```
1: []fs.FileInfo                    (Data from files listed)
2: string                               (Sanitized path. Returned from  SanitizeLastSlash() with proper
slashing format)
3: error                                (Returns this error when trying to obtain
os.Stat(/path/to/file/name/) for each file
```

func Openbrowser

```
func Openbrowser(url string)
```

Openbrowser opens default browser in system at a given URL

Takes 1 argument: 1: url string (what URI to open in brwoser)

Returns:

```
<No Return>
```

func RenderData

```
func RenderData(dirs []string, data [][]string, totSize int64, totFiles int)
```

RenderData renders a table in CLI. Takes 4 arguments with information from Files in path given as first argument to the program

1: []string (Slice with y/n values for Directory) 2: [][]string (Sorted Slice from biggest file to lowest size) 3: int64 (Total scanned file size combined) 4: int (Total files in given path)//

```
<No return>
```

func **SanitizeLastSlash**

```
func SanitizeLastSlash(path string) string
```

SanitizeLastSlash verifies that last slash is added to given path or returns it with it

Takes 1 argument:

```
1: path string                    (what system path to be listed)
```

Returns:

```
1: string                               (Sanitized path with slash at the end)
```

func **Swap**

```
func Swap(sli [][]int, i int)
```

Swap switches positions of 2 rows from [][]int slice. Rows swapped are i and i+1 index (Takes i int as second argument)

```
Takes 2 arguments:
1: sli[][]int                    (Slice containing file size information in 2 columns)
2: i int                              (i and i+1 positions where rows are going to be swapped)

<No return>
```

type **File**

```
type File struct {
	IsDir string `json:"IsDir"`
	LastM string `json:"LastM"`
	FName string `json:"FName"`
	FSize string `json:"FSize"`
}
```

func StructurizeFiles

```
func StructurizeFiles(filesStr [][]string) []File
```

StructurizeFiles converts [][]string data from files into []File struct type so it can be converted into Json

Takes 1 argument: 1: files [][]string (files in 2D array format)

Returns:

```
1: []File                         (Preformated to be json capable)
```

type PathError

```
type PathError struct {
	// contains filtered or unexported fields
}
```

func (*PathError) Error

```
func (p *PathError) Error() string
```