Go Project Documentation

Kalash Abdulaziz, Yahya Chikar, Antony Huynh, Maelys Sable, Yam Pakzad

March 21, 2025

Contents

1 Disk Utilities (disk.go)

Function: `GetDisksInfoJSON`

Description: Returns a JSON string describing each detected physical disk and its capacity in gigabytes.

- **Parameters:** None
- **Returns:**
  - `string` — JSON string of disk information
  - `error` — if disk data collection fails

Function: `getPhysicalDisks`

Description: Identifies physical disk devices by filtering partitions and removing duplicates or virtual devices.

- **Parameters:** None
- **Returns:**
  - `[]string` — list of disk device names
  - `error` — if partition scanning fails

Function: `getDiskCapacity`

Description: Detects the operating system and delegates disk capacity retrieval to an appropriate function.

- **Parameters:**
  - `diskName (string)` — device name
- **Returns:**
  - `uint64` — disk capacity in bytes
  - `error` — if detection fails

Function: `getLinuxDiskCapacity`

Description: Gets disk capacity on Linux using the `lsblk` command.

- **Parameters:** `diskName (string)`
- **Returns:** `(uint64, error)`

Function: `getWindowsDiskCapacity`

Description: Uses WMIC to get disk size on Windows.

- **Parameters:** `diskName (string)`
- **Returns:** `(uint64, error)`

Function: `getMacOSDiskCapacity`

Description: Uses macOS `diskutil` to retrieve total disk size.

- **Parameters:** `diskName (string)`
- **Returns:** `(uint64, error)`

Function: `trimLastDigit`

Description: Removes trailing digits from a disk device name.

- **Parameters:** `diskName (string)`
- **Returns:** `string`

Function: `isWSL`

Description: Checks whether the program is running inside Windows Subsystem for Linux (WSL).

- **Parameters:** None
- **Returns:** `bool`

---

2 Partition Utilities (partition.go)

Function: `mountPartition`

Description: Mounts a partition to a specified mount point using a specific filesystem.

- **Parameters:**
  - `device (string)` — device path
  - `mountPoint (string)` — mount directory
  - `fstype (string)` — file system type
- **Returns:** `error`

Function: `unmountPartition`

Description: Unmounts the given partition from the filesystem.

- **Parameters:** `device (string)`
- **Returns:** `error`

Function: `GetMountPoint`

Description: Determines a valid mount base directory from predefined paths.

- **Returns:** `string`

Function: `removeDevPrefix`

Description: Strips the /dev prefix from a device path.

- **Parameters:** `name (string)`
- **Returns:** `string`

Function: `ensureLeadingSlash`

Description: Ensures a string begins with a leading slash.

- **Parameters:** `s (string)`
- **Returns:** `string`

Function: `MountUnmountedPartitions`

Description: Automatically mounts any unmounted partitions of a given disk.

- **Parameters:** `diskName (string)`
- **Returns:** `error`

Function: `GetPartitionsInfo`

Description: Uses lsblk to retrieve partition info for a given disk.

- **Parameters:** `diskName (string)`
- **Returns:** `([]PartitionInfo, error)`

Function: `formatPartitionInfo`

Description: Converts lsblk JSON output into the internal PartitionInfo struct.

- **Parameters:** `dev (LsblkDevice)`
- **Returns:** `PartitionInfo`

Function: `GetPartitionsForDiskJSON`

Description: Returns all partition info for a disk in JSON format, mounting any that are unmounted.

- **Parameters:** `diskName (string)`
- **Returns:** `(string, error)`

3 File Utilities (file.go)

Function: `getMountPoint`

Description: Finds the mount point associated with a specific partition device.

- **Parameters:** `partition (string)`
- **Returns:**
  - `string` — mount point path
  - `error` — if not found or access fails

Function: `getFileInfo`

Description: Gathers detailed metadata for a given file or directory entry.

- **Parameters:**
  - `path (string)` — full path to the file
  - `entry (os.DirEntry)` — directory entry object
- **Returns:**
  - `FileInfo` — struct with metadata
  - `error` — if info lookup fails

Function: `AddTrailingSlash`

Description: Appends a trailing slash to a string if not already present.

- **Parameters:** `path (string)`
- **Returns:** `string`

Function: `AddSlash`

Description: Ensures that a string begins with a slash.

- **Parameters:** `s (string)`
- **Returns:** `string`

Function: `RemoveDoubleSlashes`

Description: Replaces repeated double slashes with a single slash in a path.

- **Parameters:** `s (string)`
- **Returns:** `string`

Function: `listRootFiles`

Description: Lists and returns JSON metadata of all files in a given directory path relative to a mount point.

- **Parameters:**
  - `mountPath (string)` — base mount path
  - `path (string)` — relative path inside mount
- **Returns:**
  - `string` — JSON-encoded list of file info
  - `error` — if reading or formatting fails

---

4 Block Utilities (block.go)

Function: `ReadBlock`

Description: Reads a specific 512-byte block from a file and returns its contents as a hexadecimal string.

- **Parameters:**
  - `file (*os.File)` — open file pointer
  - `blockNumber (int)` — index of the block to read
- **Returns:**
  - `string` — hex representation of block contents
  - `error` — if file seeking or reading fails

Function: `GenerateJSON`

Description: Reads an entire file in 512-byte blocks and returns a JSON object mapping block indices to hex data.

- **Parameters:** `filePath (string)` — path to the input file
- **Returns:**
  - `string` — JSON string of all file blocks in hex
  - `error` — if reading or marshaling fails

---

5 Filter Utilities (filter.go)

Function: `FilterFilesByName`

Description: Filters a list of files from a JSON input based on whether the filename contains a specified substring (case-insensitive).

- **Parameters:**
  - `jsonData ([]byte)` — JSON-encoded data representing a list of files
  - `filter (string)` — the substring to filter filenames by
- **Returns:**
  - `[]byte` — JSON-encoded data of filtered file list
  - `error` — if decoding or encoding fails

---

6 Utility Functions (utils.go)

Function: `roundFloat`

Description: Rounds a floating-point number to a given number of decimal places.

- **Parameters:**
  - `value (float64)` — the number to round
  - `precision (int)` — number of decimal places
- **Returns:** `float64` — rounded float value

---

7 API Endpoints (api.go)

Endpoint: `GET /disks`

Description: Returns a list of all physical disks with their capacity in JSON format.

- **Query Parameters:** None
- **Returns:** JSON object with disk information or an error message

---

Endpoint: `GET /partitions`

Description: Returns partition information for a specified disk in JSON format.

- **Query Parameters:**
  - `disk (string)` — required disk identifier
- **Returns:** JSON object of partitions or an error message

Endpoint: `GET /files`

Description: Lists all files in the given path of a mounted partition, with optional filtering.

- **Query Parameters:**
  - `partition (string)` — required partition identifier
  - `path (string)` — optional relative path inside mount
  - `filter (string)` — optional name filter for files
- **Returns:** JSON array of file metadata, optionally filtered by name

---

Endpoint: `GET /blocks`

Description: Reads the file content in blocks from a mounted partition and returns it as a JSON map of block indexes to hex strings.

- **Query Parameters:**
  - `partition (string)` — required partition identifier
  - `path (string)` — required path to the file
- **Returns:** JSON map of block number to hex string content

---

8 API Response Examples

/blocks — Example Response


```
{
  "0": "2320456e61626c6520506f7765726c6576656c31306b20696e7374616e742070726f6d70742e.
  "1": "207468656d652065616368207469d65206f682d6d792d7a7368206973206c6f616465642c...
  "2": "20556e636f6d6d656e742074686520666f6c6c6f77696e67206c696e6520746f2075736520...
}
```


/disks — Example Response


```
[
  {
    "name": "/dev/nvme0n1",
    "capacity_gb": 476.94
  },
  {
    "name": "/dev/sda",
    "capacity_gb": 7.68
  }
]
```


/partitions — Example Response


```
[
  {
    "name": "/dev/nvme0n1p1",
    "fs_type": "vfat",
    "mount_point": "/mnt/nvme0n1p1",
    "size": "260M",
    "fs_size": "256M",
    "fs_used": "90,5M",
    "uuid": "761A-740F",
    "part_type_name": "EFI System",
    "physical_sector_size": 512,
    "logical_sector_size": 512
  },
  ...
]
```

/files — Example Response


```
{
  "mount_path": "/",
  "files": [
    {
      "name": "bin",
      "type": "symlink",
      "is_symlink": true,
      "size_bytes": 7,
      "permissions": "Lrwxrwxrwx",
      "hard_links": 1,
      "inode": 13,
      "owner_uid": 0,
      "owner_gid": 0,
      "block_size": 4096,
      "blocks_allocated": 0,
      "last_modified": "2024-11-21 09:56:21",
      "last_access": "2025-03-12 00:05:56"
    },
    {
      "name": "boot",
      "type": "directory",
      "is_symlink": false,
      "size_bytes": 4096,
      "permissions": "drwxr-xr-x",
      "hard_links": 5,
      "inode": 1,
      "owner_uid": 0,
      "owner_gid": 0,
      "block_size": 4096,
      "blocks_allocated": 8,
      "last_modified": "Unknown",
      "last_access": "Unknown"
    },
    ...
  ]
}
```