| Node.js File System | Description |
| --- | --- |
| [Node.js fs.readFile() Method](https://www.geeksforgeeks.org/node-js-fs-readfile-method/) | This method read the entire file into buffer. To load the fs module we use require() method. |
| [Node.js fs.exists() Method](https://www.geeksforgeeks.org/node-js-fs-exists-method/) | The fs.exists() method is used to test whether the given path exists or not in the file system. |
| [Node.js fs.existsSync() Method](https://www.geeksforgeeks.org/node-js-fs-existssync-method/) | The fs.existsSync() method is used to synchronously check if a file already exists in the given path or not. |
| [Node.js fs.mkdir() Method](https://www.geeksforgeeks.org/node-js-fs-mkdir-method/) | The fs.mkdir() method i Node.js is used to create a directory asynchronously. |
| [Node.js fs.truncate() Method](https://www.geeksforgeeks.org/node-js-fs-truncate-method/) | The fs.truncate() method in node.js is used to change the size of the file i.e either increase or decrease the file size. |
| [Node.js fs.renameSync() Method](https://www.geeksforgeeks.org/node-js-fs-renamesync-method/) | The fs.renameSync() method is used to synchronously rename a file at the given old path to the given new path. |
| [Node.js fs.rmdir() Method](https://www.geeksforgeeks.org/node-js-fs-rmdir-method/) | The fs.rmdir() method is used to delete a directory at the given path. |
| [Node.js fs.stat() Method](https://www.geeksforgeeks.org/node-js-fs-stat-method/) | The fs.stat() method is used to return information about the given file or directory. |
| [Node.js fs.mkdtempSync() Method](https://www.geeksforgeeks.org/node-js-fs-mkdtempsync-method/) | The fs.mkdtempSync() method is an inbuilt application programming interface of fs module |
| [Node.js fs.realpath() Method](https://www.geeksforgeeks.org/node-js-fs-realpath-method/) | The fs.realPath() method is used to compute the canonical pathname of the given path. |
| [Node.js fs.open() Method](https://www.geeksforgeeks.org/node-js-fs-open-method/) | This is used. fs.readFile() is only for reading the file and similarly fs.writeFile() |
| [Node.js fs.mkdirSync() Method](https://www.geeksforgeeks.org/node-js-fs-mkdirsync-method/) | This is an inbuilt application programming interface of fs module which provides an API for interacting with the file system in a manner closely modeled around standard POSIX functions. |
| [Node.js fs.statSync() Method](https://www.geeksforgeeks.org/node-js-fs-statsync-method/) | The fs.statSync() method is used to synchronously return information about the given file path. |
| [Node.js fs.write() Method](https://www.geeksforgeeks.org/node-js-fs-write-method/) | The fs.write() method is an inbuilt application programming interface of fs module which is used to specify the position |
| [Node.js fs.createReadStream() Method](https://www.geeksforgeeks.org/node-js-fs-createreadstream-method/) | This is an inbuilt application programming interface of fs module which allow you to open up a file/stream and read the data present in it. |
| [Node.js fs.openSync() Method](https://www.geeksforgeeks.org/node-js-fs-opensync-method/) | This is an inbuilt application programming interface of fs module which is used to return an integer value that represents the file descriptor. |
| [Node.js fs.rmdirSync() Method](https://www.geeksforgeeks.org/node-js-fs-rmdirsync-method/) | The fs.rmdirSync() method is used to synchronously delete a directory at the given path. |
| [Node.js fs.ftruncate() Method](https://www.geeksforgeeks.org/node-js-fs-ftruncate-method/) | The fs.ftruncate() method is used to change the size of the file i.e. either increase or decrease the file size. |
| [Node.js fs.truncateSync() Method](https://www.geeksforgeeks.org/node-js-fs-truncatesync-method/) | The fs.truncateSync() method is used to synchronously change the size of the file |
| [Node.js fs.readdir() Method](https://www.geeksforgeeks.org/node-js-fs-readdir-method/) | The fs.readdir() method is used to asynchronously read the contents of a given directory. |
| [Node.js fs.writeFileSync() Method](https://www.geeksforgeeks.org/node-js-fs-writefilesync-method/) | The ‘fs’ module of Node.js implements the File I/O operation. Methods in the fs module can be synchronous as well as asynchronous. |
| [Node.js fs.readFileSync() Method](https://www.geeksforgeeks.org/node-js-fs-readfilesync-method/) | The fs.readFileSync() method is an inbuilt application programming interface of fs module which is used to read the file and return its content. |
| [Node.js fs.lstat() Method](https://www.geeksforgeeks.org/node-js-fs-lstat-method/) | The fs.lstat() method is similar to the fs.stat() method except that it is used to return |
| [Node.js fs.symlinkSync() Method](https://www.geeksforgeeks.org/node-js-fs-symlinksync-method/) | The fs.symlinkSync() method is used to synchronously create a symlink to the specified path. |
| [Node.js fs.realpathSync() Method](https://www.geeksforgeeks.org/node-js-fs-realpathsync-method/) | The fs.realpathSync() method is used to synchronously compute the canonical pathname of a given path. |
| [Node.js fs.fstat() Method](https://www.geeksforgeeks.org/node-js-fs-fstat-method/) | The fs.fstat() method is used to return information about the given file descriptor. |
| [Node.js fs.mkdtemp() Method](https://www.geeksforgeeks.org/node-js-fs-mkdtemp-method/) | The fs.mkdtemp() method is used to create a unique temporary directory. |
| [Node.js fs.lstatSync() Method](https://www.geeksforgeeks.org/node-js-fs-lstatsync-method/) | This  is used to synchronously return information about the symbolic link that is being used to refer to a file or directory. |
| [Node.js fs.readdirSync() Me thod](https://www.geeksforgeeks.org/node-js-fs-readdirsync-method/) | This is used to synchronously read the contents of a given directory. |
| [Node.js fs.writeFile() Method](https://www.geeksforgeeks.org/node-js-fs-writefile-method/) | The fs.writeFile() method is used to asynchronously write the specified data to a file. |
| [Node.js fs.fstatSync() Method](https://www.geeksforgeeks.org/node-js-fs-fstatsync-method/) | The fs.fstatSync() method is used to synchronously return information about the given file descriptor. |
| [Node.js fs.copyFileSync() Function](https://www.geeksforgeeks.org/node-js-fs-copyfilesync-function/) | This  is used to synchronously copy a file from the source path to destination path. |
| [Node.js fs. lchown() Method](https://www.geeksforgeeks.org/node-js-fs-lchown-method/) | fs.chownSync() method which does dereference the links to their path. |
| [Node.js fs. lchownSync() Method](https://www.geeksforgeeks.org/node-js-fs-lchownsync-method/) | fs.chownSync() method which does dereference the links to their path. |
| [Node.js fs.closeSync() Method](https://www.geeksforgeeks.org/node-js-fs-closesync-method/) | fs.closeSync() on a file descriptor while some other operation is being performed on it may lead to undefined behavior. |
| [Node.js fs.close() Method](https://www.geeksforgeeks.org/node-js-fs-close-method/) | This is used to asynchronously close the given file descriptor thereby clearing the file that is associated with it. |
| [Node.js fs.fchmod() Method](https://www.geeksforgeeks.org/node-js-fs-fchmod-method/) | The fs.fchmod() method is used to change the permissions of a given file descriptor. |
| [Node.js fs.link() Method](https://www.geeksforgeeks.org/node-js-fs-link-method/) | The fs.link() method is used to create a hard link to the given path. |
| [Node.js fs.symlink() Function](https://www.geeksforgeeks.org/node-js-fs-symlink-function/) | The fs.symlink() method is used to create a symlink to the specified path. |
| [Node.js fs.unlinkSync() Method](https://www.geeksforgeeks.org/node-js-fs-unlinksync-method/) | The fs.unlinkSync() method is used to synchronously remove a file or symbolic link from the filesystem. |
| [Node.js fs.chmodSync() Method](https://www.geeksforgeeks.org/node-js-fs-chmodsync-method/) | The fs.chmodSync() method is used to synchronously change the permissions of a given path. |
| [Node.js fs.chmod() Method](https://www.geeksforgeeks.org/node-js-fs-chmod-method/) | These permissions can be specified using string constants or octal numbers that correspond to their respective file modes. |
| [Node.js fs.unlink() Method](https://www.geeksforgeeks.org/node-js-fs-unlink-method/) | The fs.unlink() method is used to remove a file or symbolic link from the filesystem. |
| [Node.js fs.fchmodSync() Method](https://www.geeksforgeeks.org/node-js-fs-fchmodsync-method/) | The fs.fchmodSync() method is an inbuilt application programming of fs module which is used to synchronously change the permissions of a given file descriptor. |
| [Node.js fs.readlinkSync() Method](https://www.geeksforgeeks.org/node-js-fs-readlinksync-method/) | This  is used to synchronously return a symbolic link’s value, i.e. the path it is linked to. |
| [Node.js fs. linksync() Method](https://www.geeksforgeeks.org/node-js-fs-linksync-method/) | The hard link created would still point to the same file even if the file is renamed. |
| [Node.js fs.futimesSync() Method](https://www.geeksforgeeks.org/node-js-fs-futimessync-method/) | This is used to synchronously change the modification and access timestamps of given file descriptor. T |
| [Node.js fs.utimesSync() Method](https://www.geeksforgeeks.org/node-js-fs-utimessync-method/) | The fs.utimesSync() method is used to synchronously change the modification and access timestamps of a file. |
| [Node.js fs.utimes() Method](https://www.geeksforgeeks.org/node-js-fs-utimes-method/) | The fs.utimes() method is used to asynchronously change the modification and access timestamps of a file. |
| [Node.js fs.opendirSync() Method](https://www.geeksforgeeks.org/node-js-fs-opendirsync-method/) | The fs.opendirSync() method is used to synchronously open a directory in the file system. |
| [Node.js fs.opendir() Method](https://www.geeksforgeeks.org/node-js-fs-opendir-method/) | It creates an fs.Dir object that is used to represent the directory. |
| [Node.js fs.appendFile() Function](https://www.geeksforgeeks.org/node-js-fs-appendfile-function/) | The fs.appendFile() method is used to asynchronously append the given data to a file. A new file is created if it does not exist. |
| [Node.js fs.chown() Method](https://www.geeksforgeeks.org/node-js-fs-chown-method/) | The fs.chown() method is used to asynchronously change the owner and group of the given path. |
| [Node.js fs.fchown() Function](https://www.geeksforgeeks.org/node-js-fs-fchown-function/) | The fs.fchown() method is used to change the owner and group of the given file descriptor. |
| [Node.js fs.copyFile() Function](https://www.geeksforgeeks.org/node-js-fs-copyfile-function/) | The fs.copyFile() method is used to asynchronously copy a file from the source path to destination path. |
| [Node.js fs.appendFileSync() Function](https://www.geeksforgeeks.org/node-js-fs-appendfilesync-function/) | A new file is created if it does not exist. The optional options parameter can be used to modify the behavior of the operation. |
| [Node.js fs. fchownSync() Method](https://www.geeksforgeeks.org/node-js-fs-fchownsync-method/) | The function accepts a user id and group id that can be used to set the respective owner and group. It does not return anything. |
| [Node.js fs.chownSync() Method](https://www.geeksforgeeks.org/node-js-fs-chownsync-method/) | The fs.chownSync() method is used to synchronously change the owner and group of the given path. |
| [Node.js fs.accessSync() Method](https://www.geeksforgeeks.org/node-js-fs-accesssync-method/) | The fs.accessSync() method is used to synchronously test the permissions of a given file or directory. |
| [Node.js fs.access() Method](https://www.geeksforgeeks.org/node-js-fs-access-method/) | The permissions to be checked can be specified as a parameter using file access constants. |
| [Node.js fs.watchFile() Method](https://www.geeksforgeeks.org/node-js-fs-watchfile-method/) | The fs.watchFile() method is used to continuously watch for changes on the given file. |
| [Node.js fsPromises.mkdtemp() Method](https://www.geeksforgeeks.org/node-js-fspromises-mkdtemp-method/) | The fsPromises.mkdtemp() method is an inbuilt method which creates a unique temporary directory and resolves the Promise with the created directory path. |
| [Node.js fsPromises.chmod() Method](https://www.geeksforgeeks.org/node-js-fspromises-chmod-method/) | The fsPromises.chmod() method is used to change the permissions of a given path. |
| [Node.js fs.unwatchFile() Method](https://www.geeksforgeeks.org/node-js-fs-unwatchfile-method/) | The fs.unwatchFile() method is used to stop watching for changes on the given file. |
| [Node.js fsPromises.open() Method](https://www.geeksforgeeks.org/node-js-fspromises-open-method/) | The fsPromises.open() method is used to asynchronously open a file that returns a Promise that, when resolved, yields a FileHandle object. |
| [Node.js fsPromises.mkdir() Method](https://www.geeksforgeeks.org/node-js-fspromises-mkdir-method/) | This is used to asynchronously create a directory then resolves the Promise with either no arguments, or the first directory path created if recursive is true. |
| [Node.js fs.promise.readdir() Method](https://www.geeksforgeeks.org/node-js-fs-promise-readdir-method/) | The fs.promise.readdir() method defined in the File System module of Node.js. |
| [Node.js fsPromises.appendFile() Function](https://www.geeksforgeeks.org/node-js-fspromises-appendfile-function/) | The fsPromises.appendFile() method is used to asynchronously append the given data to a file. |
| [Node.js fsPromise.lstat() Method](https://www.geeksforgeeks.org/node-js-fspromise-lstat-method/) | The fs.promise.lstat() method is defined in the File System module of Node.js. |
| [Node.js fsPromises.stat() Method](https://www.geeksforgeeks.org/node-js-fspromises-stat-method/) | The fsPromises.stat() method is used to return information about the given file or directory. |
| [Node.js fsPromises.realpath() Method](https://www.geeksforgeeks.org/node-js-fspromises-realpath-method/) | The fsPromises.realPath() method determines the actual location of path using the same semantics as the fs.realpath.native() function |
| [Node.js fs.read() Method](https://www.geeksforgeeks.org/node-js-fs-read-method/) | Node.js is used for server-side scripting. Reading and writing files are the two most important operations that are performed in any application. |
| [Node.js fsPromises.chown() Method](https://www.geeksforgeeks.org/node-js-fspromises-chown-method/) | The fsPromises.chown() method is used to change the ownership of a file then resolves the Promise with no arguments upon success. |
| [Node.js fs.createWriteStream() Method](https://www.geeksforgeeks.org/node-js-fs-createwritestream-method/) | This is an inbuilt application programming interface of fs module which allows to quickly make a writable stream for the purpose of writing data to a file. |
| [Node.js fs.futimes() Method](https://www.geeksforgeeks.org/node-js-fs-futimes-method/) | The fs.futimes() method is used to asynchronously change the modification and access timestamps of given file descriptor. |
| [Node.js fsPromises.truncate() Method](https://www.geeksforgeeks.org/node-js-fspromises-truncate-method-2/) | The fsPromises.truncate() method is defined in the File System module of Node.js. |
| [Node.js fsPromises.symlink() Method](https://www.geeksforgeeks.org/node-js-fspromises-symlink-method/) | The fsPromises.symlink() method is used to create a symlink to the specified path then resolves the Promise with no arguments upon success. |
| [Node.js fsPromises.lchmod() Method](https://www.geeksforgeeks.org/node-js-fspromises-lchmod-method/) | The fsPromises.lchmod() method is used to change the permissions of a given path. |
| [Node.js fsPromises.lchown() Method](https://www.geeksforgeeks.org/node-js-fspromises-lchown-method/) | The fsPromises.lchown() method is used to change the ownership of a file then resolves the Promise with no arguments upon success |
| [Node.js fsPromises.truncate() Method](https://www.geeksforgeeks.org/node-js-fspromises-truncate-method/) | The fsPromises.truncate() method in node.js is used to change the size of the file i.e. either increase or decrease the file size. |
| [Node.js fsPromises.opendir() Method](https://www.geeksforgeeks.org/node-js-fspromises-opendir-method/) | The fsPromises.opendir() method is used to asynchronously open a directory in the file system. |
| [Node.js fsPromises.utimes() Method](https://www.geeksforgeeks.org/node-js-fspromises-utimes-method/) | The fsPromises.utimes() method is used to asynchronously change the modification and access timestamps of a file. |
| [Node.js fsPromises.copyFile() Method](https://www.geeksforgeeks.org/node-js-fspromises-copyfile-method/) | The fsPromises.copyFile() method is used to asynchronously copy a file from the source path to destination path. |
| [Node.js fsPromises.rename() Method](https://www.geeksforgeeks.org/node-js-fspromises-rename-method/) | The fsPromises.rename() method is used to asynchronously rename a file at the given old path to a given new path. |
| [Node.js fs.promises.appendFile() Method](https://www.geeksforgeeks.org/node-js-fs-promises-appendfile-method/) | The fs.promises.appendFile() method of File System module in Node.js is used to interact with the hard disk of the user’s computer. |
| [Node.js fsPromises.rmdir() Method](https://www.geeksforgeeks.org/node-js-fspromises-rmdir-method/) | The fsPromises.rmdir() method is used to delete a directory at the given path. It can also be used recursively to remove nested directories. |
| [Node.js fsPromises.writeFile() Method](https://www.geeksforgeeks.org/node-js-fspromises-writefile-method/) | The fsPromises.writeFile() method is used to asynchronously write the specified data to a file. By default, the file would be replaced if it exists. |
| [Node.js fsPromises.readFile() Method](https://www.geeksforgeeks.org/node-js-fspromises-readfile-method/) | The fsPromises.readFile() method is used to read the file. This method read the entire file into buffer. |
| [Node.js fsPromises.access() Method](https://www.geeksforgeeks.org/node-js-fspromises-access-method/) | The fsPromises.access() method is used to test the permissions of a given file or directory specified by path. |
| [Node.js stats.ino Property](https://www.geeksforgeeks.org/node-js-stats-ino-property/) | The stats.ino property is an inbuilt application programming interface of the fs.Stats class is used to get the “Inode” number of the file specified by the file system. |
| [Node.js stats.birthtimeNs Property](https://www.geeksforgeeks.org/node-js-stats-birthtimens-property/) | This is an inbuilt application programming interface of the fs.Stats class is used to get the timestamp when the file is created since the POSIX epoch expressed in milliseconds. |
| [Node.js stats.atimeNs Property](https://www.geeksforgeeks.org/node-js-stats-atimens-property/) | This is  an inbuilt application programming interface of the fs.Stats class is used to get the timestamp when the file is accessed last time since the POSIX epoch expressed in nanoseconds. |
| [Node.js fsPromises.opendir() Method](https://www.geeksforgeeks.org/node-js-fspromises-opendir-method-2/) | The File System module is basically to interact with the hard disk of the user’s computer. The method is used to asynchronously open a directory. |
| [Node.js stats.mtimeNs Property](https://www.geeksforgeeks.org/node-js-stats-mtimens-property/) | The stats.mtimeNs property is an inbuilt application programming interface of the fs. |
| [Node.js stats.isFIFO() Method](https://www.geeksforgeeks.org/node-js-stats-isfifo-method/) | This  is an inbuilt application programming interface of the fs.Stats class which is used to check whether fs.Stats object describes a first-in-first-out (FIFO) pipe or not. |
| [Node.js stats.isSymbolicLink() Method](https://www.geeksforgeeks.org/node-js-stats-issymboliclink-method/) | This is an inbuilt application programming interface of the fs.Stats class which is used to check whether fs.Stats object describes a symbolic link or not. |
| [Node.js stat.isSocket() Method](https://www.geeksforgeeks.org/node-js-stat-issocket-method/) | The stats.isSocket() method is an inbuilt application programming interface of the fs. |
| [Node.js stats.dev Property](https://www.geeksforgeeks.org/node-js-stats-dev-property/) | The stats.dev property is an inbuilt application programming interface of the fs. |
| [Node.js stats.mtime Property](https://www.geeksforgeeks.org/node-js-stats-mtime-property/) | The stats.mtime property is an inbuilt application programming interface of the fs.Stats class is used to get the timestamp when the file is modified last time. |
| [Node.js filehandle.readFile() Method](https://www.geeksforgeeks.org/node-js-filehandle-readfile-method/) | The filehandle.readFile() method is used to asynchronously read the file contents. This method reads the entire file into the buffer. It Asynchronously reads the entire contents of a file. |
| [Node.js stats.ctimeNs Property](https://www.geeksforgeeks.org/node-js-stats-ctimens-property/) | The stats.ctimeNs property is an inbuilt application programming interface of the fs. |
| [Node.js stats.atime Property](https://www.geeksforgeeks.org/node-js-stats-atime-property/) | This used to get the time and date when the file is accessed last time. |
| [Node.js stats.birthtime Property](https://www.geeksforgeeks.org/node-js-stats-birthtime-property/) | This is used to get the timestamp when the file is created. |
| [Node.js stats.ctime Property](https://www.geeksforgeeks.org/node-js-stats-ctime-property/) | The stats.ctime property is an inbuilt application programming interface of the fs. |
| [Node.js stats.ctimeMs Property from fs.Stats Class](https://www.geeksforgeeks.org/node-js-stats-ctimems-property-from-fs-stats-class/) | The stats.ctimeMs property is an inbuilt application programming interface of the fs. |
| [Node.js stats.mtimeMs Property from fs.Stats Class](https://www.geeksforgeeks.org/node-js-stats-mtimems-property-from-fs-stats-class/) | The stats.mtimeMs property is an inbuilt application programming interface of the fs.Stats class is used to get the timestamp |
| [Node.js stats.blocks Property from fs.Stats Class](https://www.geeksforgeeks.org/node-js-stats-blocks-property-from-fs-stats-class/) | The stats.blocks property is an inbuilt application programming interface of the fs.Stats class is used to get the number of blocks allocated for the file. |
| [Node.js stats.isCharacterDevice() Method from fs.Stats Class](https://www.geeksforgeeks.org/node-js-stats-ischaracterdevice-method-from-fs-stats-class/) | The stats.isCharacterDevice() method is an inbuilt application programming interface of the fs.Stats class which is used to check whether fs.Stats object is of a character device or not. |
| [Node.js stats.atimeMs Property from fs.Stats Class](https://www.geeksforgeeks.org/node-js-stats-atimems-property-from-fs-stats-class/) | The stats.atimeMs property is an inbuilt application programming interface of the fs.Stats class is used to get the timestamp |
| [Node.js stats.birthtimeMs Property from fs.Stats Class](https://www.geeksforgeeks.org/node-js-stats-birthtimems-property-from-fs-stats-class/) | This is used to get the timestamp when the file is created since the POSIX epoch expressed in milliseconds. |
| [Node.js filehandle.stat() Method from class: FileHandle](https://www.geeksforgeeks.org/node-js-filehandle-stat-method-from-class-filehandle/) | The File System module is basically to interact with the hard disk of the user’s computers. |
| [Node.js stats.blksize Property from fs.Stats Class](https://www.geeksforgeeks.org/node-js-stats-blksize-property-from-fs-stats-class/) | The stats.blksize property is an inbuilt application programming interface of the fs.Stats class is used to get the block size for I/O operations in the file system in bytes. |
| [Node.js stats.size Property from fs.Stats Class](https://www.geeksforgeeks.org/node-js-stats-size-property-from-fs-stats-class/) | The stats.size property is an inbuilt application programming interface of the fs.Stats class is used to get the size of the file in bytes. |
| [Node.js stats.nlink Property from fs.Stats Class](https://www.geeksforgeeks.org/node-js-stats-nlink-property-from-fs-stats-class/) | The stats.nlink property is an inbuilt application programming interface of the fs.Stats class which is used to get the number of hard-links for the file. |
| [Node.js stats.mode Property from fs.Stats Class](https://www.geeksforgeeks.org/node-js-stats-mode-property-from-fs-stats-class/) | The stats.mode property is an inbuilt application programming interface of the fs.Stats class which is used to get the file type and mode as bit-field. |
| [Node.js stats.isFile() Method from fs.Stats Class](https://www.geeksforgeeks.org/node-js-stats-isfile-method-from-fs-stats-class/) | The stats.isFile() method is an inbuilt application programming interface of the fs.Stats class which is used to check whether fs.Stats object describes a file or not. |
| [Node.js stats.gid Property from fs.Stats Class](https://www.geeksforgeeks.org/node-js-stats-gid-property-from-fs-stats-class/) | This is used to get the numeric (number / bigint) identity of the group to which the file belongs to. |
| [Node.js stats.uid Property from fs.Stats Class](https://www.geeksforgeeks.org/node-js-stats-uid-property-from-fs-stats-class/) | The stats.uid property is an inbuilt application programming interface of the fs.Stats class is used to get the numeric (number / bigint) identity of the user to which the file belongs to. |
| [Node.js stats.rdev Property from fs.Stats Class](https://www.geeksforgeeks.org/node-js-stats-rdev-property-from-fs-stats-class/) | This  is used to get the numeric (number / bigint) identity of the device in which the file is stored in the file is considered to be “special”. |
| [Node.js fs.filehandle.utimes() Method](https://www.geeksforgeeks.org/node-js-fs-filehandle-utimes-method/) | The fs.filehandle.utimes() method is an inbuilt application programming interface of class fs.filehandle within File System |
| [Node.js fs.filehandle.truncate() Method](https://www.geeksforgeeks.org/node-js-fs-filehandle-truncate-method/) | This is used to truncate the particular file object and only that much byte will be retained which passed as an integer in truncate() method. |
| [Node.js fs.Dirent.isFIFO() Method](https://www.geeksforgeeks.org/node-js-fs-dirent-isfifo-method/) | This is used to check if the particular dirent describes a first in first out pipe or not. |
| [Node.js fs.filehandle.fd() Method](https://www.geeksforgeeks.org/node-js-fs-filehandle-fd-method/) | This is used to provide the numeric file descriptor of this file handle object. |
| [Node.js fs.Dirent.isSocket() Method](https://www.geeksforgeeks.org/node-js-fs-dirent-issocket-method/) | This is used to check if the particular dirent describes a Socket or not. |
| [Node.js fs.Dirent.isFile() Method](https://www.geeksforgeeks.org/node-js-fs-dirent-isfile-method/) | This is used to check if the particular dirent describes a File or not. |
| [Node.js fs.filehandle.write() Method](https://www.geeksforgeeks.org/node-js-fs-filehandle-write-method/) | This is an inbuilt application programming interface of class fs.filehandle within File System module which is used to write the data form the buffer to that particular file. |
| [Node.js fs.Dir.close() Method](https://www.geeksforgeeks.org/node-js-fs-dir-close-method/) | This  is used to close the directory’s underlying resource handle asynchronously. |
| [Node.js fs.Dirent.isCharacterDevice() Method](https://www.geeksforgeeks.org/node-js-fs-dirent-ischaracterdevice-method/) | This is used to check if the particular dirent describes a character device or not. |
| [Node.js fs.Dir.read() Method](https://www.geeksforgeeks.org/node-js-fs-dir-read-method/) | This is used to read the each next directory (dirent) one by one asynchronously. |
| [Node.js fs.Dirent.isBlockDevice() Method](https://www.geeksforgeeks.org/node-js-fs-dirent-isblockdevice-method/) | This is used to check if the particular dirent describes a block device or not. |
| [Node.js filehandle.close() Method](https://www.geeksforgeeks.org/node-js-filehandle-close-method/) | This is used to asynchronously close the given file descriptor thereby clearing the file that is associated with it. |
| [Node.js filehandle.read() Method](https://www.geeksforgeeks.org/node-js-filehandle-read-method/) | In order to read files without file descriptor the readFile() method of the filehandle package can be used. |
| [Node.js stats.isDirectory() Method from fs.Stats Class](https://www.geeksforgeeks.org/node-js-stats-isdirectory-method-from-fs-stats-class/) | This is used to check whether fs.Stats object describes a file system directory or not. |
| [Node.js filehandle.writeFile() Method from class: FileHandle](https://www.geeksforgeeks.org/node-js-filehandle-writefile-method-from-class-filehandle/) | This is used to defined in the File System module of Node.js. The File System module is basically to interact with the hard disk of the user’s computer. |
| [Node.js fs.Dir.path() Method](https://www.geeksforgeeks.org/node-js-fs-dir-path-method/) | This is used to defined in the File System module of Node.js. The File System module is basically to interact with the hard disk of the user’s computer. |
| [Node.js fs.filehandle.datasync() Method](https://www.geeksforgeeks.org/node-js-fs-filehandle-datasync-method/) | fs.filehandle within File System module which is used to sync the data of the file. |
| [Node.js fs.filehandle.chmod() Method](https://www.geeksforgeeks.org/node-js-fs-filehandle-chmod-method/) | fs.filehandle within File System module which is used to change the permission of the particular file. |
| [Node.js stats.isBlockDevice() Method from fs.Stats Class](https://www.geeksforgeeks.org/node-js-stats-isblockdevice-method-from-fs-stats-class/) | This  is used to check whether fs.Stats object is of a block device or not. |
| [Node.js fs.filehandle.chown() Method](https://www.geeksforgeeks.org/node-js-fs-filehandle-chown-method/) | This is used to change the ownership of the particular file. |
| [Node.js fs.filehandle.sync() Method](https://www.geeksforgeeks.org/node-js-fs-filehandle-sync-method/) | fs.filehandle within File System module which is used to synchronize this file’s in-core state with the storage device. |
| [Node.js filehandle.truncate() Method](https://www.geeksforgeeks.org/node-js-filehandle-truncate-method/) | The truncate() method used to modify the inner contents of the file by ‘len’ bytes. |
| [Node.js fs.dirent.name Property](https://www.geeksforgeeks.org/node-js-fs-dirent-name-property/) | The fs.Dirent.name property is an inbuilt application programming interface of class fs.Dirent |
| [Node.js filehandle.appendFile() Method](https://www.geeksforgeeks.org/node-js-filehandle-appendfile-method/) | The File System module is basically to interact with the hard disk of the user’s computer. |
| [Node.js fs.Dirent.isDirectory() Method](https://www.geeksforgeeks.org/node-js-fs-dirent-isdirectory-method/) | This is used to check if the particular dirent describes a Directory or not. |
| [Node.js fs.Dirent.isSymbolicLink() Method](https://www.geeksforgeeks.org/node-js-fs-dirent-issymboliclink-method/) | This is used to check if the particular dirent describes a SymbolicLink or not. |
| [Node.js fs.Dir.readSync() Method](https://www.geeksforgeeks.org/node-js-fs-dir-readsync-method/) | fs.Dir within File System module which is used to read the each next dirent of directory. |
| [Node.js fs.Dir.closeSync() Method](https://www.geeksforgeeks.org/node-js-fs-dir-closesync-method/) | This is used to close the directory’s underlying resource handle synchronously. |
| [Node.js fs.promises.link() Method](https://www.geeksforgeeks.org/node-js-fs-promises-link-method/) | The fs.promises.link() method is an inbuilt application programming interface of the fs.promises class which is used to make a new name for a file. |
| [Node.js fs.watch() Method](https://www.geeksforgeeks.org/node-js-fs-watch-method/) | fs module which is used to continuously watch for changes in the given file or directory. |
| [Node.js fs.readlink() Method](https://www.geeksforgeeks.org/node-js-fs-readlink-method/) | The fs.readlink() method is an inbuilt application programming interface of fs module which is used to asynchronously return a symbolic link’s value. |
| [Node.js fs.fdatasync() Method](https://www.geeksforgeeks.org/node-js-fs-fdatasync-method/) | The fs.fdatasync() (Added in v0.1.96) method is an inbuilt application programming interface of the fs module which is similar to fs.fsync() |
| [Node.js fs.fsync() Method](https://www.geeksforgeeks.org/node-js-fs-fsync-method/) | It has both synchronous and asynchronous forms. The asynchronous form always takes a completion callback as its last argument. |
| [Node.js fs.rmSync() Method](https://www.geeksforgeeks.org/node-js-fs-rmsync-method/) | The fs.rmSync() method is used to synchronously delete a file at the given path. |
| [Node.js fs.rm() Method](https://www.geeksforgeeks.org/node-js-fs-rm-method/) | The fs.rm() method is used to delete a file at the given path. It can also be used recursively to remove directories. |
| [Node.js fs.writeSync() Method](https://www.geeksforgeeks.org/node-js-fs-writesync-method/) | This is used to work with files on the computer. The functions of the module can be used by importing the fs module. |