tree object internally.

In this stage you'll implement the `git ls-tree` command with the `--name-only` flag. Here's how the output looks with the `--name-only` flag:

```
$ git ls-tree --name-only <tree_sha>
dir1
dir2
file1
```

The tester uses `--name-only` since this output format is easier to test against.

We recommend implementing the full `ls-tree` output too since that'll require that you parse all data in the tree object, not just filenames.

Tree Object Storage

▼ Click to expand/collapse

Just like blobs, tree objects are stored in the `.git/objects` directory. If the hash of a tree object is `e88f7a929cd70b0274c4ea33b209c97fa845fdbc` , the path to the object would be `./git/objects/e8/8f7a929cd70b0274c4ea33b209c97fa845fdbc` .

The format of a tree object file looks like this (after Zlib decompression):

```
tree <size>\0
<mode> <name>\0<20_byte_sha>
<mode> <name>\0<20_byte_sha>
```

(The above code block is formatted with newlines for readability, but the actual file doesn't contain newlines)

- The file starts with `tree <size>\0` . This is the "object header", similar to v

Tests failed.   [Show logs]