Ganpat University

॥ विद्यया समाजोत्कर्षः ॥

#StudentsFirst
#CharacterMust

(Software Packages)

By

Prof. Hiteshri Modi

Node.js as a File Server

- The Node.js file system module allows you to work with the file system on your computer.
- To include the File System module, use the require() method:

```
var fs = require('fs');
```

- Common use for the File System module:
  - Read files
  - Create files
  - Update files
  - Delete files
  - Rename files

Read Files

- fs.readFile() method is used to read files on your computer.

Syntax:

fs.readFile(filename, encoding, callback_function)

Parameters: The method accept three parameters as mentioned above and described below:

- **filename:** It holds the name of the file to read or the entire path if stored at other location.
- **encoding:** It holds the encoding of file. Its default value is **'utf8'**.
- **Callback function:** It is a callback function that is called after reading of file. It takes two parameters:
    - **err:** If any error occurred.
    - **data:** Contents of the file.

Return Value: It returns the contents/data stored in file or error if any.

fs.readFile()

```
var http = require('http');
var fs = require('fs');
http.createServer(function (req, res) {
  fs.readFile('demofile1.html', function(err, data) {
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.write(data);
    return res.end();
  });
}).listen(8080);
```

Create Files

- fs.appendFile()
- fs.open()
- fs.writeFile()

fs.appendFile()

- The fs.appendFile() method appends specified content to a file. If the file does not exist, the file will be created:

```
var fs = require('fs');

fs.appendFile('mynewfile1.txt', 'Hello content!', function (err) {
  if (err) throw err;
  console.log('Saved!');
});
```

fs.open()

- The fs.open() method takes a "flag" as the second argument, if the flag is "w" for "writing", the specified file is opened for writing.

- If the file does not exist, an empty file is created:

```
var fs = require('fs');

fs.open('mynewfile2.txt', 'w', function (err, file) {
  if (err) throw err;
  console.log('Saved!');
});
```

fs.writeFile()

- fs.writeFile() method replaces the specified file and content if it exists. If the file does not exist, a new file, containing the specified content, will be created:

```
var fs = require('fs');

fs.writeFile('mynewfile3.txt', 'Hello content!', function (err) {
  if (err) throw err;
  console.log('Saved!');
});
```

Update Files

- The File System module has methods for updating files:
  - fs.appendFile()
  - fs.writeFile()

fs.appendFile()

- The fs.appendFile() method appends the specified content at the end of the specified file:

```
var fs = require('fs');

fs.appendFile('mynewfile1.txt', ' This is my text.', function (err) {
  if (err) throw err;
  console.log('Updated!');
});
```

fs.writeFile()

- The fs.writeFile() method replaces the specified file and content:

```
var fs = require('fs');

fs.writeFile('mynewfile3.txt', 'This is my text', function (err) {
  if (err) throw err;
  console.log('Replaced!');
});
```

Delete Files

- To delete a file with the File System module, use fs.unlink() method.
- The fs.unlink() method deletes the specified file:

```
var fs = require('fs');

fs.unlink('mynewfile2.txt', function (err) {
  if (err) throw err;
  console.log('File deleted!');
});
```

Rename Files

- To rename a file with the File System module, use the fs.rename() method.
- The fs.rename() method renames the specified file:

```
var fs = require('fs');

fs.rename('mynewfile1.txt', 'myrenamedfile.txt', function (err) {
  if (err) throw err;
  console.log('File Renamed!');
});
```