Web-www.webisoftech.com
E-mail- enquiry@webisoftech.com
Phone- +91 9766962674

Chapter 10
Node JS File System

Node.js File System (FS)

In Node.js, file I/O is provided by simple wrappers around standard POSIX functions. Node File System (fs) module can be imported using following syntax:

Syntax:

```
var fs = require("fs")
```

Node.js FS Reading File

Every method in fs module has synchronous and asynchronous forms.

Asynchronous methods take a last parameter as completion function callback. Asynchronous method is preferred over synchronous method because it never blocks the program execution where as the synchronous method blocks.

Let's take an example:

Create a text file named "input.txt" having the following content.

File: input.txt

File: main.js

```
var fs = require("fs");
// Asynchronous read
fs.readFile('input.txt', function (err, data) {
   if (err) {
      return console.error(err);
   }
   console.log("Asynchronous read: " + data.toString());
});
// Synchronous read
var data = fs.readFileSync('input.txt');
console.log("Synchronous read: " + data.toString());
```

console.log("Program Ended");

Open Node.js command prompt and run the main.js:

//.....

Node.js Open a file

Syntax:

Following is the syntax of the method to open a file in asynchronous mode:

fs.open(path, flags[, mode], callback)

Parameter explanation:

Following is the description of parameters used in the above syntax:

path: This is a string having file name including path.

flags: Flag specifies the behavior of the file to be opened. All possible values have been mentioned below.

mode: This sets the file mode (permission and sticky bits), but only if the file was created. It defaults to 0666, readable and writeable.

callback: This is the callback function which gets two arguments (err, fd).

Node.js Flags for Read/Write

Following is a list of flags for read/write operation:

r :- open file for reading. an exception occurs if the file does not exist.

r+ :- open file for reading and writing. an exception occurs if the file does not exist.

rs:- open file for reading in synchronous mode.

Web-www.webisoftech.com
E-mail- enquiry@webisoftech.com
Phone- +91 9766962674

rs+ :- open file for reading and writing, telling the os to open it synchronously. see notes for 'rs' about using this with caution.

w:- open file for writing. the file is created (if it does not exist) or truncated (if it exists).

wx:- like 'w' but fails if path exists.

w+ :- open file for reading and writing. the file is created (if it does not exist) or truncated (if it exists).

wx+ :- like 'w+' but fails if path exists.

a:- open file for appending. the file is created if it does not exist.

ax:- like 'a' but fails if path exists.

a+ :- open file for reading and appending. the file is created if it does not exist.

ax+ :- open file for reading and appending. the file is created if it does not exist.

Create a JavaScript file named "main.js" having the following code to open a file input.txt for reading and writing.

File: main.js

```
var fs = require("fs");
// Asynchronous - Opening File
console.log("Going to open file!");
fs.open('input.txt', 'r+', function(err, fd) {
   if (err) {
       return console.error(err);
   }
  console.log("File opened successfully!");
});
```

Open Node.js command prompt and run the main.js:

node main.js

//.....

Web-www.webisoftech.com
E-mail- enquiry@webisoftech.com
Phone- +91 9766962674

Node.js File Information Method

Syntax:

Following is syntax of the method to get file information.

fs.stat(path, callback)

Parameter explanation:

Path: This is string having file name including path.

Callback: This is the callback function which gets two arguments (err, stats) where stats is an object of fs.Stats type.

Node.js fs.Stats class Methods

stats.isfile()
returns true if file type of a simple file.

stats.isdirectory()
returns true if file type of a directory.

stats.isblockdevice()
returns true if file type of a block device.

stats.ischaracterdevice()
returns true if file type of a character device.

stats.issymboliclink()
returns true if file type of a symbolic link.

stats.isfifo()
returns true if file type of a fifo.

stats.issocket()
returns true if file type of asocket.

Web-www.webisoftech.com
E-mail- enquiry@webisoftech.com
Phone- +91 9766962674

Let's take an example to create a JavaScript file named main.js having the following code:

File: main.js

```
var fs = require("fs");
console.log("Going to get file info!");
fs.stat('input.txt', function (err, stats) {
   if (err) {
      return console.error(err);
   }
   console.log(stats);
   console.log("Got file info successfully!");
   // Check file type
   console.log("isFile ? " + stats.isFile());
   console.log("isDirectory ? " + stats.isDirectory());
});
```