

Node.js: working with files: creating a basic helper class

Node.js provides us with a complete API to work with files.

We can create the following helper class:

```
'use strict';

const path = require('path');
const fs = require('fs');
const ABSPATH =
path.dirname(process.mainModule.filename);

class Files {
  static read(path, encoding = 'utf8') {
    return new Promise((resolve, reject) => {
      let readStream = fs.createReadStream(ABSPATH
+ path, encoding);
      let data = '';

      readStream.on('data', chunk => {
        data += chunk;
      }).on('end', () => {
        resolve(data);
      }).on('error', err => {
        reject(err);
      });
    });
  }
}
```

```
static create(path, contents) {
    return new Promise((resolve, reject) => {
        fs.writeFile(ABSPATH + path, contents,
(err, data) => {
            if(!err) {
                resolve(data);
            } else {
                reject(err);
            }
        });
    });
}
```

```
static remove(path) {
    return new Promise((resolve, reject) => {
        fs.unlink(ABSPATH + path, err => {
            if(!err) {
                resolve(path);
            } else {
                reject(err);
            }
        });
    });
}
```

```
static exists(path) {
    return new Promise((resolve, reject) => {
        fs.access(ABSPATH + path,
fs.constants.F_OK, err => {
            if(!err) {
                resolve(true);
            } else {
                reject(err);
            }
        });
    });
}
```

```
        });  
    });  
}  
  
module.exports = Files;
```

Example:

```
'use strict';  
  
const Files = require('./lib/Files');  
  
Files.create('/files/test.txt', 'Test').then(data =>  
{  
    console.log(data);  
}).catch(err => {  
    console.log(err);  
});
```