

FileManager

FileManager

```
new FileManager()
```

Source: [util/fileManager.js, line 22](#)

Creates a file manager. This class can read and write files.

Methods

```
convertFileToFolderObject(subdir, fileName) → {Promise.<any>}
```

Source: [util/fileManager.js, line 260](#)

Helper function for the 'getProjectFiles' function, This function converts directories to directory endpoints, then recursively calls the getProjectFiles function to add children endpoints to itself.

Parameters:

Name	Type	Description
subdir		Directory name to convert to an object.
fileName		File name to convert to an object.

Returns:

Returns an object representing this directory and all children of this directory.

Type `Promise.<any>`

```
convertFileToObject(fileName) → {Object}
```

Source: [util/fileManager.js, line 247](#)

Helper function for the 'getProjectFiles' function, simply converts a string name, into an object indicating that this file is an endpoint, not a directory.

Parameters:

Name	Type	Description

Name	Type	Description
<code>fileName</code>		File name to convert to an object.

Returns:

Type `Object`

```
getProjectFiles(subdir, ignore) → {Promise.<any>}
```

Source: [util/fileManager.js, line 178](#)

This function is a recursive call, it will propagate through all subdirectories of 'subdir' until all child directories have been traversed.

Parameters:

Name	Type	Description
<code>subdir</code>	<code>String</code>	The directory to open and convert to a JSON object.
<code>ignore</code>	<code>Ignore</code>	This is the blacklist information to reference when generating this object.

Returns:

This promise will resolve once all subdirectories have been traversed and a valid save object is generated.

Type `Promise.<any>`

```
initialize() → {Promise.<any>}
```

Source: [util/fileManager.js, line 36](#)

This function is used to initialize this file manager. When this function is called, a promise is returned. The promise will resolve once the file defined by 'SAVE_PATH/CONFIG_FILE' has been read and parsed into a JSON object.

Returns:

Type `Promise.<any>`

```
loadFile(fileName) → {Promise.<any>}
```

Source: [util/fileManager.js, line 61](#)

This function allows a user to load a file relative to the 'SAVE_PATH' for example, if the user were to pass 'sampleLanguage.json' to this file, the framework would try to load the file 'root/sampleLanguage.json' Once the file has been found, the contents will be read as utf8 text and returned as a promise. This promise resolves once all lines of the file have been read and are contained within the 'data' object.

Parameters:

Name	Type	Description
fileName		

Returns:

Type `Promise.<any>`

```
(async) readFromProperties(fieldName) → {Promise.<any>}
```

Source: [util/fileManager.js, line 106](#)

This function is the inverse of 'writeToProperties' It allows a user to read the field 'fieldName' off of the properties object.

Parameters:

Name	Type	Description
fieldName		The field to read.

Returns:

This function returns a promise that resolves when the data is read off of the field, and rejects when their is an error reading that specific field.

Type `Promise.<any>`

```
writeToFile(fileName, data) → {Promise.<any>}
```

Source: [util/fileManager.js, line 155](#)

This function allows a user to write data to an arbitrary file. The user can specify the file in 'fileName' and the contents of that file in the 'data' object. The file referenced by file name will be in the path defined by 'SAVE_PATH/fileName'.

Parameters:

Name	Type	Description
fileName		The name of the file to write to.
data		The data to write to the file.

Returns:

Type `Promise.<any>`

```
(async) writeToProperties(field, data) → {Promise.<any>}
```

Source: [util/fileManager.js, line 82](#)

This function allows a developer to easily write to the config json object. This object is persisted between instances of the application running.

Parameters:

Name	Type	Description
field	String	This is the field on the config object that you want to set.
data	Object	This is the value that 'field' should be set to.

Returns:

This function returns a promise which resolves if the write was successful, or rejects if there was an error.

Type `Promise.<any>`

Documentation generated by [JSDoc 3.5.5](#) on Mon Apr 22 2019 11:05:28 GMT-0400 (Eastern Daylight Time) using the [docdash](#) theme.