Creating hashes in Node

There are times when we want to create a hash from either a string or an existing file. This has multiple uses. Some that come to mind are:

- Validate the the file has not changed. As long as the file remains the same the hash will remain the same too
- Use it as a unique identifier for a file by combining the hash with the file name

This is different than creating a UUID with Node. The UUID is random and is not tied to the file. The hash is tied to the file and will produce the same hash if the file hasn't changed.

The hash methods are part of the crypto module so they don't require third party installation.

Using module syntax (which will require you to save the file with the `.mjs` extension), it looks like this:

```
import crypto from 'crypto';
```

The createHash will create a hash.

The udpate method will update the hash with the new data, this data will only change if the data (the file) has changed.

Finally the digest will calculate the hash using the specified encoding.

```
const hashedValue = crypto.createHash('md5')
    .update(fileName)
    .digest('hex');
```

You can then use the hash to create a unique identifier for the file. In the following example we use the hash and the file name to create a unique file name. Changing the content of the file will change the hash, change the file name and bust any caches the file might be stored in

```
const fullName =
    `${fileName}-${hashedValue}.txt`;
```