Validator

# Is-empty.js

const isEmpty = value =>

value === undefined ||

value === null ||

(typeof value === "object" && Object.keys(value).length === 0) ||

(typeof value === "string" && value.trim().length === 0);

module.exports = isEmpty;

We can use this function to first check or sanitise the input we are validating. If the value supplied to this function is:

* Undefined
* Null
* An object with no keys
* A String with the length of 0

Then this function will return *True.*

# Login.js

The example below shows a function that we could use to validate the information a user submits when they are trying to login.

**Let errors = {} –** we first create an errors object that we will populate and return if there are any issues.

**isEmpty** – We are then using the isEmpty function we defined earlier to ensure that the input we are trying to validate is in a format that is acceptable for the **Validator** module.

**//Login validation rules –** Here we have 3 ‘rules’ that we are applying to the user input we are validating, if any of these ‘rules’ fail then the *errors* object we created earlier is populated with a correspondent reason as to why the rule failed.

1. Is the *email* value of the user input present?
2. Is the *email* value of the user input an email?
3. Is the *password* value of the user input present?

const Validator = require("validator");

const isEmpty = require("./is-empty");

module.exports = function validateLoginInput(data) {

let errors = {};

// if these values are not present in the data object we are validating then they will be set to empty strings for the Validator.isEmpty

data.email = !isEmpty(data.email) ? data.email : "";

data.password = !isEmpty(data.password) ? data.password : "";

//Login validation rules  
if (Validator.isEmpty(data.email)) {

errors.email = "Email field is required";

}

if (!Validator.isEmail(data.email)) {

errors.email = "Email is invalid";

}

if (Validator.isEmpty(data.password)) {

errors.password = "Password field is required";

}

return {

errors,

isValid: isEmpty(errors)

};

};