MYOB Code Challenge

First of all, I want to thank you for the opportunity of develop this challenge. In this document I present useful information about the workflow of the system, the technologies and the testing as well as all the assumptions for the system.

Understanding the system

Functional details

The system is presented in web. It is not necessary to install anything, by clicking over '*Index.html'* file a website is open and the system is ready to upload a file.

In order to launch the tests, just click in '*TestRunner.html'*

Technical details:

For developing this system, the technologies have been HTML, CSS, Javascript as well as Jasmine for the tests. My objective has been 'Keep It Short and Simple' (KISS), attached is the library which allows the Jasmine framework to execute the tests.

This system starts its workflow in Index.html. This webpage, when the user loads a file, invokes a function placed in the file '*InputFileController'*. After that the system process the uploaded file ('*EmployeeDataController'*) and when all the data are parsed, the system does all the operations needed to get the output data ('*PaySlipsOperationsController'* and '*ATOTableTaxes'*). Once all the data is stored, the system uses '*TableBuilder'* to show the results and to create a link ('*OutputFileController'*) just in case the user wants to download the data into a csv file.

In the folder call 'spec' there are 3 files which contain the unit tests code.

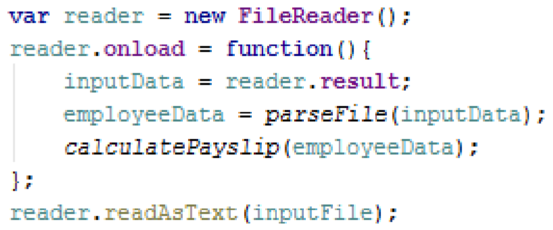
Testing

In the folder Test Cases Files you can find seven csv files used to test the system. I tried to cover all the possibilities with it.

Assumptions:

Technical

It is considered that some functions of Javascript are not necessary tested, since are native code, so they must have been tested by the system, specifically, I refer to the functions related with the load and read of the file, in *InputFileController*:



```
var reader = new FileReader();
reader.onload = function(){
    inputData = reader.result;
    employeeData = parseFile(inputData);
    calculatePayslip(employeeData);
};
reader.readAsText(inputFile);
```

Is consider that, reader.onload is going to execute the code inside the function.

Since the HTML elements cannot be tested by coding, there are no tests about this. This have been visually tested.

Functional

The upload file is restricted to have csv extension. If the file has other extension the system detected that and launch an alert warning the user.

It is considered that *annualSalary* and *super* input data are **mandatory.** Neither first name, last name, or pay period are consider as mandatory data. If the user wants to not fill this data, the system can calculate the payslip.

It is considered that if the data are not filled, it must exist a comma to separate the data, i.e. if the user does not want to fill first name and last name, the record will be:

“,,60050,9%,01 March – 31 March”

The same for the mandatory data:

“David,Rudd,60050, ,01 March - 31 March”

Any other case, for example, if the document has not five fields for every record, the system will not process that record and shows an error in the place of this record. So, if the input data is:

“60050,9%,01 March – 31 March”

The system will show this error:

“Not the correct input data filled. The record is not processed”

If the data are filled, but one (or more) of the mandatory data is lost, the system will show:

“Mandatory data not filled. The operations cannot be done”

It is considered that the user might need to download a csv file with the data shown in the webpage, so a link is enabled to download a csv file with the output data. The input data will be always with a csv file.

The rates are the correspondant to *“2015-2016 Apply from 1 of July 2015”* (current ones)