**U Bank - Code Challenge Solution**

Read Me

Author: Kamal Saluja

Date: 11-Feb-2019

Table of Contents

[Document Objective 3](#_Toc851568)

[Problem statement: 3](#_Toc851569)

[Assumptions 3](#_Toc851570)

[Solution Structure 3](#_Toc851571)

[Key Features 5](#_Toc851572)

[References 5](#_Toc851573)

[Sample Data used 6](#_Toc851574)

[Improvement Areas 6](#_Toc851575)

## Document Objective

Objective of this document is to provide information about the solution for coding challenge. It contains details about problem, proposed solution structure, technical stack & key features (ES 6), sample data and some nice to have features i.e. can be further included with the solution.

## Problem statement:

Requirement is to deliver a java script function that can receive path of input file in specific structure, and returns the collection of employees where each employee should contain name and total number of leaves.

## Assumptions

Few assumptions have been made during development to ensure desired function to run efficiently.

1. Function to return friendly message in case ‘File not found’ at provided location in input parameters.
2. Function to return friendly message ‘No Records Found’ in case file doesn’t contain any records.
3. In case input file contains duplicate dates for one employee, function will remove it and process only unique dates.

e.g. if file has records for employee ‘Jon 20180202 20180202 20180405 20180202’ it will output as ‘{ name: ‘Jon’, total: 2 }’

## Solution Structure

Node JS is used to implement the solution for provided problem. Below are some key java script files are created.

|  |  |
| --- | --- |
| **File / Folder** | **Description** |
| Components / Controller / LeaveController.js | This file is used to export the required function (getEmployeeLeaves) i.e. takes the input from end user as file name provide result in desired structure.  Idea behind creating such folder structure is to place all endpoints under Controller folder and then create index.js under Components folder which can be helpful at time of bootstrapping. It will allow single import statement to load all modules from controllers.  e.g. ***const controllers = require(‘../Controller’)*** will point to index.js that will load all modules at single line.  Screenshots for results are provided in references section in this document. |
| Components / test / LeaveController.spec.js | This file contains 6 unit test cases for below behaviours   1. Process records when no comments or duplicates 2. Process records & ignore comments 3. Process records after removing duplicate dates for one employee 4. When empty file or file with spaces found.   Jasmine-node is used as dependency for test cases framework.  Screenshots for results are provided in references section in this document. |
| Services / file-api / FileService.js | This file contains class i.e. responsible for reading the file. |
| Services / file-api / Index.js | This file provides single instance for File Service. |
| Main.js | This file is used to demonstrate how function can be used to integrate with any project. |

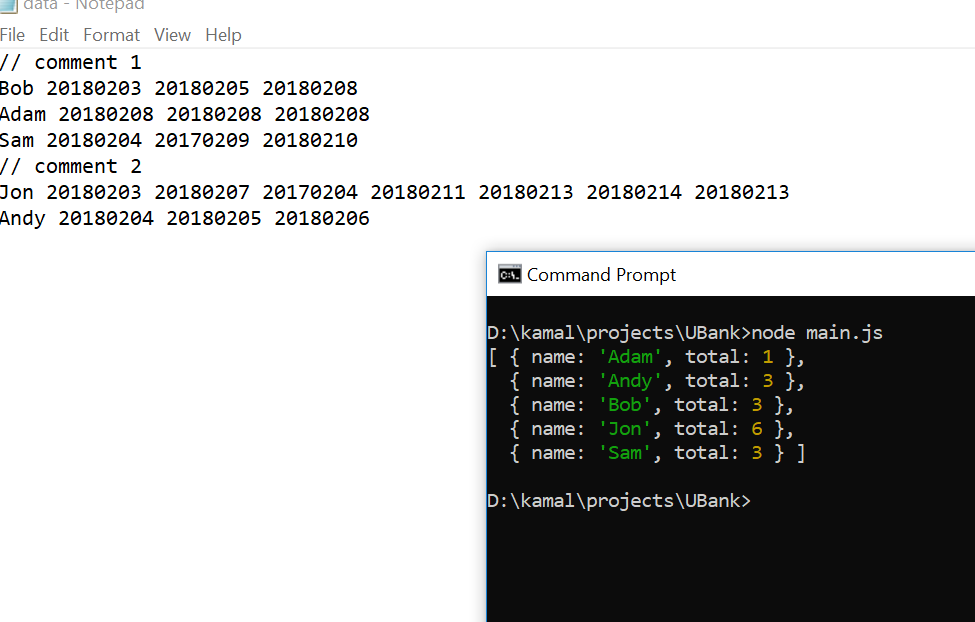
## Key Features

Node JS is used to implement the solution (getEmployeeLeaves). There are some key features of ES 6 / Node 8 are considered into it.

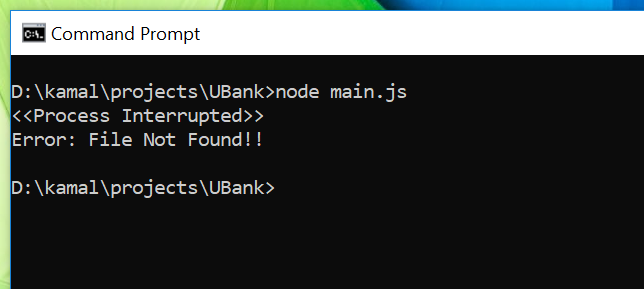
1. Asynchronous code using async / await pattern
2. Set(s) are used to remove duplicates
3. Classes / constructors are used for File Service instead of Constructor Functions.

## References

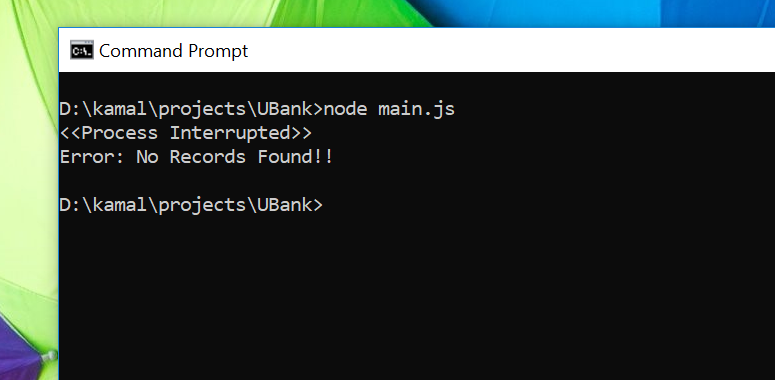
Scenario 1: Produce results when records are available. Comments are duplicates (dates) are handled in preparing result.



Scenario 2: Friendly error will be returned in case, file not found on provided location



Scenario 3: Friendly error will be returned in case file is empty or contains only spaces.



## 

## Sample Data used

// comment 1

Bob 20180203 20180205 20180208

Adam 20180208 20180208 20180208

Sam 20180204 20170209 20180210

// comment 2

Jon 20180203 20180207 20170204 20180211 20180213 20180214 20180213

Andy 20180204 20180205 20180206

## Improvement Areas

As per requirement, for assessment purpose, dependencies are not supposed to use hence there are some further enhancements that can be considered in future using some dependencies.

1. Errors can be logged to some source
2. Express JS can be integrated to expose function as API endpoint.
3. JOI can be use to set some schema validations for input records.