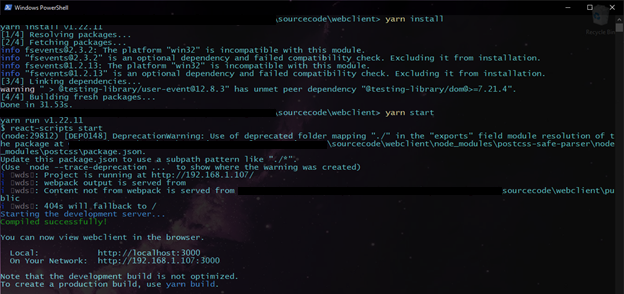
Required software

1. Nodejs @ 16.5.0
2. Yarn @ 1.22.11

Frontend – Installation

* Go to sourcecode/webclient
* Yarn install
* Yarn start



Backend – Installation

* Go to sourcecode/webserver
* Yarn install
* Yarn dev



API endpoints

* api/v1/diff/left
  + A post endpoint to create left data
* api/v1/diff/right
  + A post endpoint to create right data
* api/v1/diff
  + A get endpoint to do comparison between the left and right data
  + The data comparison code can be found in the sourcecode/webserver/src/helper/dataHelper.ts file

Possible Result 01: Same base64 binary data length but different content

The left and right json base64 binary data has the same length but difference content. Checks for the first and last index of difference and the length of difference. (Will ignore the differences after the last index of difference)

{

  “isSame”: false,

  “isSameLength”: true,

  “leftDataLength”: 8,

  “rightDataLength”: 8,

  “initialOffset”: 4,

  “finalOffset”: 7,

  “length”: 3,

}

Possible Result 02: Different base64 binary data

The left and right json base64 binary data has different json base64 binary data. Returns the length of the data respectively.

{

  “isSame”: false,

  “isSameLength”: false,

  “leftDataLength”: 3,

  “rightDataLength”: 8,

  “initialOffset”: -1,

  “finalOffset”: -1,

  “length”: -1,

}

Possible Result 03: Same base64 binary data

The left and right has the same json base64 binary data and length

  “isSame”: true,

  “isSameLength”: true,

  “leftDataLength”: 8,

  “rightDataLength”: 8,

  “initialOffset”: -1,

  “finalOffset”: -1,

  “length”: -1,

}

Frontend - Testing

* Ensure the yarn install command has been executed at the sourcecode/webclient folder
* yarn run test

Frontend – Testing Coverage

* Application should have
  + Input form – text field
  + Input form – left submit button
  + Input form – right submit button
  + Left object selector
  + Right object selector
* Input form
  + Left submit button should call onSubmit method with valid input
  + Right submit button should call onSubmit method with valid input
  + Should not call onSubmit method and prompt for value with invalid input (empty)

Backend – Testing

* Ensure the yarn install command has been executed at the sourcecode/webserver folder
* yarn run test
* Expected error can be ignore as it is due to the timeout for calling the service endpoint, execute the **yarn run test** command a few times and the error will disappear

Backend – Testing Coverage

* Application should have
  1. Left service – the left service end point for api/v1/diff/left
  2. Right service – the right service end point for api/v1/diff/right
  3. Diff service – the diff service end point for api/v1/diff
* Application should
  1. Left service – able to call the endpoint api/v1/diff/left via the left service
  2. Right service – able to call the endpoint api/v1/diff/right via the right service
  3. Diff service – able to call the endpoint api/v1/diff via the diffservice
* Service function
  1. The diff service should only accept the “leftData” and “rightData” as query parameters, reject otherwise
  2. The left and right service should only accept json base64 binary data, reject otherwise
  3. The left and right service should process the json base64 binary data and returns a json result mentioned in the API endpoint section

Improvements – overall

1. Use of docker to host both the frontend and backend system
2. Create test report for both the frontend and backend system
3. Make use of Eslint to fix and enforce coding style for both frontend and backend system

Improvements – Frontend

1. Refactor frontend code to identify common components
2. Add file upload into the input form history (Planned to do but not enough time)
3. ~~Fix the bug on the comparison history as it may not update correctly occasionally~~ (done)

Improvements – Backend

1. Disable unused REST endpoints in the backend server
2. Persist the compare history (Planned to do but not enough time)
3. Data comparison can be done in the frontend to unload server overhead in big scale data processing

Things to note:

* The text will only be highlighted if the two base64 binary data are of the same length
* If the frontend is not able to create data
  + stop the backend using ctrl + c
  + go to “sourcecode/webserver”
  + remove the data folder
  + restart the webserver using yarn dev