Emily Buckley

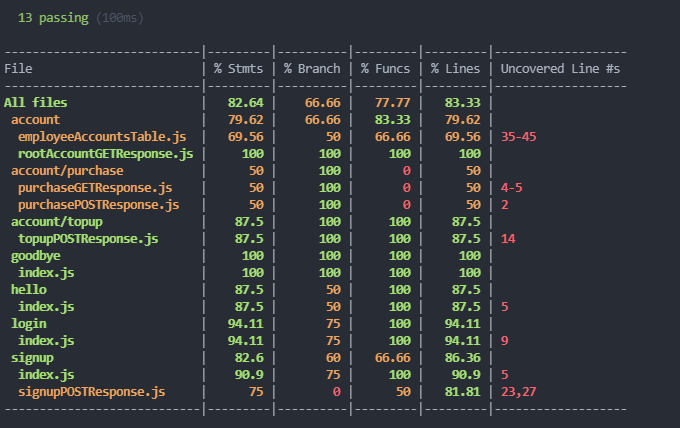
First catering API tESTING DOCUMENT

# Manual Tests

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number | Test Description | Test Tools | Expected Outcome | Results of Test | Pass/ Fail |
| 1 | A GET request to the default endpoint replies with a response saying good morning or good afternoon depending on the time of day.  Endpoint:  https://dol4161gx7.execute-api.eu-west-1.amazonaws.com/ | Postman | Expecting a 200 response  With this response as the message: "Good afternoon, please tap your card to begin" as it is currently 1pm |  | Pass |
| 2 | A GET request to the goodbye endpoint replies with goodbye if there are no params  Endpoint: https://dol4161gx7.execute-api.eu-west-1.amazonaws.com/goodbye | Postman | Expecting a 200 response with this in the body: “Goodbye!” | The API responded with 500 and “Internal server error”, I need to look back on this lambda to fix this issue, I will also be retesting this later | Fail |
| 3 | A GET request to the goodbye endpoint replies with goodbye and the users name if a name query string parameter has been passed in  Endpoint: https://dol4161gx7.execute-api.eu-west-1.amazonaws.com/goodbye?name=Pickle Rick | Postman | Expecting a 200 with this in the body: “Goodbye Pickle Rick” |  | Pass |
| 4 | A GET request to the login endpoint asks the user for their pin  Endpoint: https://dol4161gx7.execute-api.eu-west-1.amazonaws.com/login?cardId=r2KlqdaGLk0dFJo9 | Postman | Expecting a 200 response with the message "Good afternoon Emily Buckley, please enter in your pin" |  | Pass |
| 5 | A GET request to the login endpoint with a cardId that hasn’t been registered in the database  Endpoint:  https://dol4161gx7.execute-api.eu-west1.amazonaws.com/login?cardId=1 | Postman | Expecting a 404 response with "You are being redirected to signup" |  | Pass |
| 6 | A GET request to the account endpoint with the cardId and pin that the user has requested, the card is found and the pin is correct  Endpoint:  https://dol4161gx7.execute-api.eu-west-1.amazonaws.com/account?cardId=r2KlqdaGLk0dFJo9&pin=9023 | Postman | Expecting a 200 response with "Emily Buckley, you currently have £0 in your account. What would you like to do? please select 1. top-up or 2. purchase food" |  | Pass |
| 7 | A GET request to the account endpoint with the cardId and pin that the user has requested, the card is found and the pin is incorrect  Endpoint:  https://dol4161gx7.execute-api.eu-west-1.amazonaws.com/account?cardId=r2KlqdaGLk0dFJo9&pin=0000 | Postman | Expecting a 401 response with "Your pin is incorrect, you are being redirected back to login" in the body |  | Pass |
| 8 | A GET request to the account endpoint with a malformed request, it doesn’t include the pin  Endpoint:  https://dol4161gx7.execute-api.eu-west-1.amazonaws.com/account?cardId=r2KlqdaGLk0dFJo9 | Postman | Expecting a 400 response with "bad request, you are being redirected back to default endpoint" |  | Pass |
| 9 | A PUT request (or any unallowed method) to the account endpoint  Endpoint: https://dol4161gx7.execute-api.eu-west-1.amazonaws.com/account?cardId=r2KlqdaGLk0dFJo9&pin=9023 | Postman | Expecting a 405 response with “PUT, is not allowed” in the body | A 404 is returned, this is because API gateway does not allow through requests that have not been set up so they will not be allowed through to my lambda. The piece of code that was to handle this I have now taken out as it is not necessary. Even though my test failed the API still passes as this is the correct response. | Pass |
| 10 | A GET request for a cardId that is not assigned yet.  Endpoint: https://dol4161gx7.execute-api.eu-west-1.amazonaws.com/account?pin=9023&cardId=not found | Postman | Expecting a 404 response with “Your card is not assigned, you are being redirected to signup” |  | Pass |
| 11 | A POST request to the account/topup endpoint that includes the amount the user wants to update  Endpoint: https://dol4161gx7.execute-api.eu-west-1.amazonaws.com/account/topup?cardId=r7jTG7dqBy5wGO4L&pin=0000  Body: {  “updateBalance”: 10.00  } | Postman | Expecting a 201 response with the database being updated. |  | Pass |
| 12 | A GET request to the account/purchase endpoint that includes their cardId and pin as query string parameters.  Endpoint: https://dol4161gx7.execute-api.eu-west-1.amazonaws.com/account/purchase?cardId=r7jTG7dqBy5wGO4L&pin=0000 | Postman | Expecting a 200 Response with a list of items that are currently in stock in the shop |  | Pass |
| 13 | A GET request to the signup endpoint.  Endpoint: https://dol4161gx7.execute-api.eu-west-1.amazonaws.com/signup?cardId=not registered | Postman | Expecting a 200 response with "Your card isn’t registered, please enter in your EmployeeID, Name, Email, mobile number and secret pin. Please remember your pin as you will need it for logging in" in the body |  | Pass |
| 14 | A POST request to the signup endpoint.  Endpoint: https://dol4161gx7.execute-api.eu-west-1.amazonaws.com/signup?cardId=u3llqdhGLj9dFKo1  Body: {  "employeeId":"dawk23c",  "mobileNo":"07238198321",  "name": "Greg Carter",  "pin":"3333",  "email":"greg.carter@f1.com"  } | Postman  DynamoDB console | Expecting a 201 response with the users name inputted into the message responded "Greg Carter your signed up! You will now be redirected to the login page where you will be prompted with your pin"  You should also see the database being updated with these details |  | Pass |
| 15 | A POST request to the purchase endpoint  Endpoint: https://dol4161gx7.execute-api.eu-west-1.amazonaws.com/account/purchase?cardId=r7jTG7dqBy5wGO4L&pin=0000 | Postman | Expecting a 200 response with “this endpoint has not been set up” |  | Pass |
| 15 | Fixed the goodbye endpoint that had a bug in test 2. I had to update the function to not require queryStringParameters by using optional chaining. | Postman | Expecting a 200 response with “Goodbye!” in the body. |  | Pass |

# Unit tests

I created unit tests for each file, I could not do this inside the lambda console, so I cloned all of my code and created tests using ‘mocha’ and ‘sinon’.



There were a total of 13 tests, I tried to cover all error scenarios however some edge cases were not covered because of the time restriction, this allowed me to iron out some issues I was initially having with the top-up endpoint. Going through this process helped me greatly.

I managed to cover the simple functions easily however some of the more complicated files had paths that I did not have time to create the tests for them, this included the purchaseGETResponse.js which needed two databases to be mocked as part of the test.