Unit Tests Report

SlotMeIn

1 June 2021

Backend:

The backend was tested with automated unit tests using the Jest framework. Each API route has a unit testing file that covers all of the API calls defined. These tests include the files where the functionality is implemented and any files implementing database queries for the routes. The following list details what was tested for each route, the percentage of statement coverage for each backend file, and which team member wrote and performed the tests.

1. **attendeesRoute.test.js - Tester: Justyn**
    a. getTotalAttendees tests
        i. get the attendees for a valid event (200)
        ii. get the attendees from a non existent event (200)
    b. attendees.js Statement Coverage - **100%**
2. **businessRoute.test.js - Tester: Justyn**
    a. signup tests
        i. successful new business account signup (201)
        ii. unsuccessful account signup, name already taken (409)
    b. login tests
        i. successful login (200)
        ii. email not found (404)
        iii. incorrect password (401)
    c. getBusiness tests
        i. gets a valid businesses information (200)
        ii. invalid token type (403)
        iii. bad token (401)
    d. getBusinessEvents tests
        i. gets a businesses events (200)
        ii. invalid token type (403)
        iii. bad token (401)
    e. {businessid}/events tests
        i. Gets a businesses events (200)
        ii. Business id is not found (404)
    f. checkBusinessID tests
        i. Given token is a valid ID (200)
        ii. Invalid token given (403)
        iii. Bad token (401)
    g. Get Business By ID tests
        i. Given token is a valid ID (200)
        ii. Invalid token given (404)

h. Businesses get tests
    i. successful retrieval of all businesses (200)
i. uploadProfileImage tests
    i. successful image upload (200) (Through Swagger UI, Don't want to dump 10MB strings in testing)
    ii. Invalid File type (400) (Through Swagger UI, Don't want to dump 10MB strings in testing)
j. getProfileImage tests
    i. successfully get image (201)
    ii. User token (403)
    iii. Bad token (401)
k. businesses.js Statement Coverage - **67.96%** (because upload was tested manually through Swagger UI)
l. businessDb.js Statement Coverage - **88.89%**

3. eventsRoute.test.js - Tester: Justyn

a. getEvents tests
    i. business token getting events (200)
    ii. user token getting events (200)
    iii. bad token (401)
b. getEventByID tests
    i. successfully get an event by its ID (200)
    ii. invalid ID (404)
c. signup tests
    i. successfully signed up for event (200)
    ii. user does not meet restrictions for event (403)
    iii. bad event id (404)
    iv. user already signed up (409)
    v. user is not a member of business (403)
    vi. event is at capacity (403)
d. publicEvents tests
    i. get all public events (200)
e. publicAndMemberEvents tests
    i. get all public and member events events (200)
f. get Categories tests
    i. get the categories from the database (200)
g. events post tests
    i. successful creation of new event (201)
    ii. successful creation of repeating event (201)
    iii. bad business token (401)
    iv. user token (403)
h. events delete tests
    i. Successfully deleted event (200)
    ii. Successfully deleted repeating event (200)

iii. Event not found (404)
iv. Delete event created by another business (403)

i. getSearchEvents tests
i. successfully retrieved business events (200)
ii. successfully retrieved user events (200)

j. events.js Statement Coverage - **96.09%**
k. eventsDb.js Statement Coverage - **77.36%**
l. attendeesDb.js Statement Coverage - **61.90%**

4. membersRoute.test.js - Tester: Justyn

a. getMember tests
i. normal get members (200)
ii. Bad token (401)

b. getMemberBusinesses tests
i. user will businesses (200)
ii. user with no businesses (200)
iii. bad email (400)

c. getRestrictedEvents tests
i. user with a membership with restricted events (200)
ii. user with a membership and no restricted events (200)
iii. bad email (400)

d. addMembers tests
i. adds 3 members (2 non users, 1 already a user) (200)
ii. adds existing member (200)
iii. bad business token (401)

e. removeMembers tests
i. removes a member (200)
ii. bad business token (401)

f. members.js Statement Coverage - **92.73%**
g. memberDb.js Statement Coverage - **86.27%**

5. usersRoute.test.js - Tester: Ethan

a. signup tests
i. Signup with valid account (200)
ii. Signup with existing email (409)

b. login tests
i. Successful login (200)
ii. Login with incorrect password (401)
iii. Login with non-existing email (404)

c. getUser tests
i. getUser With Valid User Token (200)
ii. getUser With Invalid User Token (401)

d. getUserEvents tests
i. getUserEvents With Valid User Token (200)
ii. getUserEvents With Invalid User Token (401)

e. removeUserAttending tests
    i. removeUserAttending With Valid Event ID (200)
    ii. removeUserAttending With Invalid Event ID (403)
f. users.js Statement Coverage - **93.18%**
g. userDb.js Statement Coverage - **84.62%**

Frontend:

_____The frontend was tested manually with dummy data that was setup in the database's Docker image.

1. **Home.js**
    a. When not logged in, shows grid of public events
    b. When logged in, shows events from businesses user is a member of + public events
    c. Clicking on an event takes you to proper event page
2. **Login.js**
    a. Displays error message on unsuccessful login
    b. Business account checkbox properly switches between logging into user or business account
    c. Can submit form by pressing button or pressing Enter
    d. Redirects to Home on successful login
3. **Register.js**
    a. Business account checkbox changes form appropriately
        i. Business account: Name, email, password, description, phone number
        ii. User account: Name, email, password, birthdate
    b. Shows error message if email is already in use
    c. Can submit form by pressing button or pressing Enter
    d. Redirects to Home on successful login
4. **ViewEvents.js**
    a. Shows events from businesses user is a member of
    b. Shows public events
    c. Search fields for start time, end time, or text search
    d. Event filters for businesses, event categories, event restrictions
5. **AllEvents.js**
    a. Shows all upcoming events, both public & from businesses user is a member of
6. **IndividualEvent.js**
    a. Shows all event information properly
    b. Sign up button switches to withdraw if user signs up for event
    c. Sign up button is disabled if event has no spots left
    d. Delete event button shows if logged into business account

    e. Delete event button is disabled if business did not create the event
    f. Business info is shown on the left side
    g. Other events by the same business shown on the right side
    h. Clicking on related events takes you to the proper event page
    i. Invalid event ID in URL shows 404 Not Found page
7. **PublicBusinessProfile.js**
    a. Business info is shown on the left side
    b. Events by the business shown on the right side
    c. Invalid business ID in URL shows 404 Not Found page
8. **UserProfile.js**
    a. Shows user's info (name, email)
    b. Displays calendar view with the events user has signed up for
9. **BusinessProfile.js**
    a. Shows overview of business's upcoming events
    b. Shows business info
    c. Business can upload a profile image
    d. Business can view and add members by email
    e. Business can share links to their profile through Facebook or Twitter
10. **CreateEvent.js**
    a. Fields for event name, start time, end time, capacity, description
    b. Can create a repeating event
    c. Can set event categories and restrictions
    d. Start time cannot come after end time
    e. Can submit form by pressing button or pressing Enter
11. **NavBar.js**
    a. Buttons link to appropriate pages
    b. Logout button unauthenticated user and redirects to Home
    c. Create Event button only shows when logged into BusinessProfile
    d. NavBar does not show on Home page when not logged in
12. **Footer.js**
    a. About/Contact links go to appropriate pages
    b. Displays at the bottom of each page
13. **NotFound.js**
    a. Displays "404 Not Found" text