Bettina Kron 2023-04-26

Testrapport för Bookfinder-API

Denna rapport beskriver testningen av API:et Bookfinder och innehåller manuell testning och automatisk testning med vardera 13 testfall. För varje testfall beskrivs kortfattat uppgiften, motivering till varför testet görs, det förväntade resultatet, det faktiska resultatet samt eventuella problem/fel eller kommentarer. I vissa av be

En del test ligger tillsammans i en undermapp till BOOKFINDER-collection. Detta därför att jag läste att det är ”best practice” att lägga test som hör ihop i separata mappar.

**Manuell testning**

**Uppgift 1**

*Verify that the API returns the correct HTTP status code (e.g., 200 OK) for a successful GET request.*

**Motivering**

Kontrollera om API:t returnerar en bok med tillhörande information.

**Request**

[{{baseURL}}/api/books](http://localhost:3480/api/books)

[{{baseURL}}/api/books/6445109b0d36e24c9a94c0e0](http://localhost:3480/api/books/6445109b0d36e24c9a94c0e0)

**Förväntat resultat**

Statuskod 200 samt titel på boken, författarens namn samt annan detaljerad information om boken.

**Faktiska resultat**

Se förväntat resultat

**Problem/Fel**

Inga

**Uppgift 2**

*Check if the API returns the expected data format (e.g., JSON, XML) in the response.*

**Motivering**

Kontrollera att API:t returnerar data i JSON-format.

**Förväntat resultat**

Att API:t returnerar information i dataformatet JSON samt meddelande om att formatet är JSON.

**Faktiska resultat**

I Body visas datan i JSON-format. I Headers/Content-Type visas value: /json

**Problem/Fel**

Inga

**Uppgift 3**

*Ensure that the API returns the correct HTTP status code (e.g., 400 Bad Request) for an invalid request.*

**Motivering**

Kontrollera att API:t returnerar status kod 400 när jag gör en felaktig request.

**Request**

{{baseURL}}/api/boo

**Förväntat resultat**

Ett statusmeddelande ”404 Not Found”

**Faktiska resultat**

Se förväntat resultat

**Problem/Fel**

Inga

**Uppgift 4**

*Test if the API returns the correct data when querying with specific filters or search criteria.*

**Motivering**

Testa att filtrera sökningar för att få specifik information i retur, t ex på författare eller genre.

**Request**

{{baseURL}}/api/books?genre=Drama

{{baseURL}}/api/books?author=Irvin Langosh

**Förväntat resultat**

Att API:t returnerar ”status 200 OK” samt alla objekt där ”genre” = ”Drama” respektive det objekt vars ”author” är Irvin Langosh.

**Faktiska resultat**

Se förväntat resultat.

**Problem/Fel**

Inga

**Uppgift 5**

*Verify that the API returns paginated results when a large number of records are requested.*

**Motivering**

Testa för att få ut specifikt antal objekt, i detta fall tre böcker med tillhörande information.

**Request**

{{baseURL}}/api/books?limit=3

**Förväntat resultat**

Att API:t returnerar ”status 200 OK” samt de tre första objekten.

**Faktiska resultat**

Se förväntat resultat.

**Problem/Fel**

Inga

**Uppgift 6**

*Test if the API handles special characters and non-English text correctly in input data and returned responses using an automated testing tool.*

**Motivering**

Testa om API:t hanterar specialtecken som t ex åäöë, i detta fall ö.

**Request**

{{baseURL}}/api/books?title=ö

**Förväntat resultat**

Error 404 (Not found)

**Faktiska resultat**

Status code 200 OK samt en tom array.

**Problem/Fel**

Egentligen inget fel eftersom det inte finns någon titel i API:t som innehåller ”ö”

**Uppgift 7**

*Test the API’s response when sending concurrent requests to ensure that it can handle multiple users and maintain data consistency.*

**Motivering**

Kolla om API:t klarar konkurrerande/simultana requests.

**Request**

{{baseURL}}/api/books

**Förväntat resultat**

Någon form av krasch, error code 503 eller liknande

**Faktiska resultat**

Status code 200 OK.

**Problem/Fel**

Svårt att testa detta manuellt. Jag klickade flera gånger på musen. Jag testade också med Run Collection/run manually/iteration:10. Testade kanske för få gånger.

**Uppgift 8**

*Test if the API correctly handles different HTTP methods (GET, POST, PUT, DELETE) for each endpoint and returns appropriate status codes and responses for each method.*

**Motivering**

Testa att API:t klarar CRUD, i detta fall: skapa (create) en ny bok, hämta (read) den nya boken genom författarnamnet, uppdatera (put) någon genre och antal sidor samt ta bort (delete) boken.

**Request**

POST {{baseURL}}/api/books

GET {{baseURL}}/api/books?author=Kazuo Ishiguro

PUT {{baseURL}}/api/books/64464c7bfb76d9c72260835c

DELETE {{baseURL}}/api/books/64464c7bfb76d9c72260835c

**Förväntat resultat**

Status code 200 OK samt (uppdaterad) information om nya boken. Någon sorts information om att boken är borttagen.

**Faktiska resultat**Se förväntat resultat. Delete gav informationen ”null”.  
**Problem/Fel**

Inga

**Uppgift 9**

*Check if the API correctly handles updates to existing records, ensuring that changes are saved and reflected in subsequent requests.*

**Motivering**

Kolla om API:t klarar att hantera uppdateringar av en bok.

**Request**

{{baseURL}}/api/books/6445109b0d36e24c9a94c0e4

**Förväntat resultat**

Status code 200 OK samt (uppdaterad) information om nya boken.

**Faktiska resultat**

Se förväntat resultat.

**Problem/Fel**

Inget

**Uppgift 10**

*Test the API’s performance under heavy load, simulating a large number of users making requests simultaneously.*

**Motivering**

Kolla om/att API:ts prestanda under hög belastning. Detta görs med 100 anrop med hjälp av ”Run collection (Runner)”.

**Request**

{{baseURL}}/api/books

**Förväntat resultat**

Statuskod 200 OK men med fördröjning av något slag.

**Faktiska resultat**

Statuskod 200 OK   
Duration: 2 sek 319 msek

Avrage response time: 12 msek

**Problem/Fel/Kommentar**

Inget fel. Viss fördröjning pga att 100 anrop gjordes. *Ett* anrop med samma get-request tog 22 msek.

**Uppgift 11**

*Verify that the API can recover gracefully from failures, such as database connection issues without compromising data integrity.*

**Motivering**

Kolla om API:t klarar ett avbrott från servern och databasen. Detta gjorde jag genom att:

1. Skicka en vanlig get-request, 2. Stoppa servern och anslutningen till databasen MongoDB 3. Skicka samma get-request igen 4. Starta servern och återansluta till databasen igen 5. Skicka samma get-request igen.

**Request**

{{baseURL}}/api/books

**Förväntat resultat**

Status code 200 OK respektive status code 500 (Server error) eller kanske code 404 (Not found)

**Faktiska resultat**

Status code 200 OK respektive ”Error: connect ECONNREFUSED” Could not send request

**Problem/Fel/Kommentar**

Inga

**Uppgift 12**

*Test the API’s ability to handle edge cases, such as requests with missing or invalid parameters, and ensure that appropriate error messages are returned.*

**Motivering**

Kolla om API:t klarar att hantera när användaren skickar in fel parametrar, i detta fall ”publisher” (som inte finns).

**Request**

{{baseURL}}/api/books?publisher=Bloomsbury

**Förväntat resultat**

Status code 400 Bad request.

**Faktiska resultat**

Se förväntat resultat.

**Problem/Fel**

Inget

**Uppgift 13**

*Verify that the API correctly implements rate limiting or throttling mechanisms to prevent abuse or excessive use of resources.*

**Motivering**

Kontrollera om/att API:et implementerar hastighetsbegränsande mekanismer korrekt om man skickar ett stort antal förfrågningar inom en kort tidsram. Kontrolleras med ”Runner” och 500 itereringar.

**Request**

{{baseURL}}/api/books/6445109b0d36e24c9a94c0e0

**Förväntat resultat**

Lämpligt felmeddelande att hastighetsgränsen har överskridits

**Faktiska resultat**

Code 200 OK. Gjorde 500 itereringar och under den tiden överskreds ingen gräns.



**Problem/Fel**

Vet ej. Kanske behövde testa många fler requests än 500.

**Automatiserad testning**

**Uppgift 1**

*Validate that the API returns the correct HTTP status code (e.g., 200 OK) for a successful GET request.*

**Motivering**

Testa att responsen är korrekt, dvs ger statuskod 200 OK

**Request**

{{baseURL}}/api/books?rating=5

**Förväntat resultat**

statuskod 200 OK

**Faktiska resultat**

Se förväntat resultat

**Problem/Fel**

Inga

**Uppgift 2**

*Verify that the API returns the expected data format (e.g., JSON, XML) in the response.*

**Motivering**

Testa att responsen är JSON-format/innehåller JSON-kod

**Request**

Samma som i uppgift 1

**Förväntat resultat**

PASS Status 200 OK

**Faktiska resultat**

Se förväntat resultat

**Problem/Fel**

Inga

**Uppgift 3**

*Ensure that the API returns the correct HTTP status code (e.g., 400 Bad Request) for an invalid requests.*

**Motivering**

Testa att responsen visar statuskod 400 Bad Request om felaktig parameter skickas med requesten

**Request**

{{baseURL}}/api/books?field=authors

(field finns inte)

{{baseURL}}/api/books?hello

**Förväntat resultat**

Statuskod 400 Bad Requst

**Faktiska resultat**

FAIL testet ger tillbaka status 200 OK

**Problem/Fel**

Jag har försökt lägga in try/catch med 400 i api.js men jag har inte fått det att fungera. Ändrar jag koden får jag det att funka men då kraschar limit och sort.

**Uppgift 4**

*Create an automated test that sends a request with specific filters or search criteria and checks if the API returns the correct data.*

**Motivering**

Testa att ett specifikt filter returnerar korrekt information

**Request**

{{baseURL}}/api/books?genre=Drama

**Förväntat resultat**

PASS Status 200 OK

**Faktiska resultat**

Se förväntat resultat

**Problem/Fel**

Inga

**Uppgift 5**

*Write an automated test to verify that the API returns paginated results when a large number of records are requested.*

**Motivering**

Testa att API:t returnerar korrekt antal poster per sida

**Request**

{{baseURL}}/api/books?limit=3

**Förväntat resultat**

PASS Status 200 OK

**Faktiska resultat**

Se förväntat resultat

**Problem/Fel**

Inga

**Uppgift 6**

*Test if the API handles special characters and non-English text correctly in input data and returned responses using an automated testing tool.*

**Motivering**

Testa om API:t returnerar specialtecken och icke-engelska tecken korrekt

**Request**

{{baseURL}}/api/books?title=Kärlekens ÅÄÖ

**Förväntat resultat**

PASS Status 200 OK

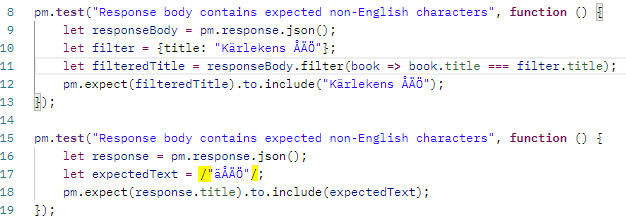
**Faktiska resultat**

Se förväntat resultat

**Problem/Fel/Kommentarer**

Jag lade först upp en ny bok innehållande icke-engelska bokstäver.

Jag gjorde sedan två olika test men fick FAIL på båda, tills jag angav ”/” (se gulmarkerat nedan). Det testet fick status PASS, så API:et tycks kunna hantera specialtecken.



**Uppgift 7**

*Develop an automated test that sends concurrent requests to the API to ensure that it can handle multiple users and maintain data consistency.*

**Motivering**

Testa att API:t kan hantera samtidiga förfrågningar och upprätthålla ”datakonsistens” (vilket jag tolkar som att säkerställa att datat i databasen förblir korrekt).

**Request**

{{baseURL}}/api/books?genre=Drama

**Förväntat resultat**

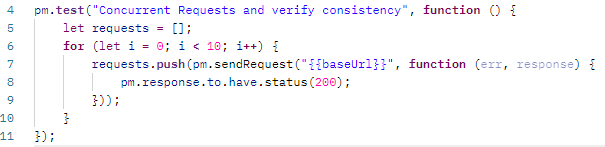
PASS Status 200 OK

**Faktiska resultat**

Se förväntat resultat

**Problem/Fel/Kommentarer**

Jag loopade ”requesten” 10 gånger och villkoret var att svaret/responsen skulle ha status 200, vilket den blev.



**Uppgift 8**

*Create an automated test and test if the API correctly handles different HTTP methods (GET, POST,*

*PUT, DELETE) for each endpoint and returns appropriate status codes and responses for each method.*

**Motivering**

För att testa om API:t hanterar de olika HTTP-metoderna GET, POST, PUT och DELETE) korrekt

**Request**

{{baseURL}}/api/books

**Förväntat resultat**

POST: ny post med status 201 Created PASS

GET/:id: hämta nya posten och status 200 OK PASS

PUT/:id: se den uppdaterade posten samt få status 200 OK PASS

DELETE: se den borttagna posten samt få status 200 OK PASS

**Faktiska resultat**

Se förväntat resultat

**Problem/Fel/Kommentarer**

Inga fel.

Testade alla HTTP-metoderna mes status 200/201 samt att fanns JSON-data i responsen.

**Uppgift 9**

*Write an automated test to check if the API correctly handles updates to existing records, ensuring that changes are saved and reflected in subsequent requests.*

**Motivering**

Testa uppdatering för att säkerställa att ändringar sparas och återspeglas i efterföljande förfrågningar.

**Request**

{{baseURL}}/api/books/’:id’

**Förväntat resultat**

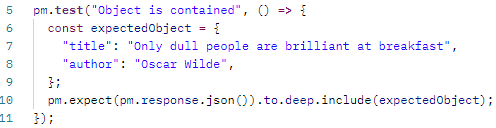
PASS 200

**Faktiska resultat**

Se förväntat resultat

**Problem/Fel/Kommentar**

Inga, men det var svårt att veta hur jag skulle testa detta. Förutom ett test för statuskod 200 gjorde jag ett som kollar att returnerat objekt innehåller uppdateringarna som gjordes.



**Uppgift 10**

*Design an automated performance test that simulates a large number of users making requests simultaneously to check the API’s performance under heavy load.*

**Motivering**

Testa API:ts prestanda genom att simulera flera förfrågningar/requests samtidigt.

**Request**

{{baseURL}}/api/books?genre=Drama

**Förväntat resultat**PASS 200 OK men med fördröjning

**Faktiska resultat**

Som förväntat resultat



**Problem/Fel/Kommentarer**

Jag testade statuskod 200, vilket jag fick PASS på. Jag gjorde också ett test för att svarstiden skulle vara under 500 msek. Det klarade API:t också. I detta test kördes 100 itereringar i ”Runner”.

**Uppgift 11**

*Create an automated test that verifies the API can recover gracefully from failures, such as database connection issues or third-party service outages, without compromising data integrity.*

**Motivering**

Testa om/hur API:t kan återhämta sig från fel, t ex databasanslutningar.

**Request**

{{baseURL}}/api/books

**Förväntat resultat**

Förväntade mig inget resultat eftersom jag inte lyckades lösa denna uppgift.

**Faktiska resultat**

Se förväntat resultat

**Problem/Fel/Kommentar**

Försökte googla och ta hjälp av chatGPT men jag fick inga vettiga tips, så denna uppgift får jag lämna olöst.

**Uppgift 12**

*Develop an automated test to handle edge cases, such as requests with missing or invalid parameters, and ensure that appropriate error messages are returned.*

**Motivering**

Testa hur API:t hanterar förfrågningar som saknar parametrar eller är ogiltiga.

**Request**

{{baseURL}}/api/books/{{invalidInput}}

**Förväntat resultat**

PASS Status code 500 Indicates server error

FAIL AssertionError: expected response to have status code 404 but got 500

**Faktiska resultat**

Se förväntat resultat

**Problem/Fel/Kommentar**

Gjorde två tester för att fånga upp error code 400 Invalid Request och error code 500 Server Error. Att det förstnämnda inte fungerar visste jag efter som jag inte har fått min try/catch att fånga det felet i min booksRouter.get() i books.js. Oavsett går koden in else-satsen med

res.status(500).json({ message: err.message });

**Uppgift 13**

*Write an automated test to verify that the API correctly implements any rate limiting or throttling mechanisms to prevent abuse or excessive use of resources.*

**Motivering**

Skriva ett automatiserat test för att verifiera att API:et implementerar alla hastighetsbegränsande eller ”strypningsmekanismer” korrekt för att förhindra missbruk eller överdriven användning av resurser.

**Request**

{{baseURL}}/api/books

**Förväntat resultat**

PASS Status code 200

**Faktiska resultat**

Stoppade innan alla requests hade gått klart. Fick PASS Status code 200 på dem som gått klart.

**Problem/Fel/Kommentarer**

Vet inte om detta är ett rätt utfört test. Jag tog hjälp av chatGPT eftersom jag inte förstod hur jag skulle lösa det. Testet tog lång tid att genomföra, och därför avbröt jag.

Jag la detta test i en separat mapp (rate limiting) så att jag kunde köra hela mappen (”Run folder”) utan att riskera att råka köra andra requests också. I "Pre-request Script" la jag kod som fördröjer med 1 sek. I "Tests" la jag ett pm.test för response status code 200.