

Testguide til Auktion Projektet

Introduktion

I denne guide vil vi præsentere vores tilgang til testning af vores softwareprojekt. Vi har benyttet en kombination af Test Driven Development (TDD) og unit tests (MSTest) i kildekoden og API tests til vores endepunkter i Postman. Dette dokument er ment som en guide til at udføre disse tests.

Testmiljøer

- Unit Testing Framework (MSTest): Vi har implementeret enhedstests direkte i kildekoden for vores CatalogService, for at teste individuelle metoder og funktioner.
- Postman: Vi har brugt Postman til at udføre API-tests. Dette inkluderer test af endpoints, verifikation af HTTP-statuskoder, og validering af JSON-responsdata.

Tests i services

- AuthService: Test af autentificering og autorisation.
- UserService: Test af brugeroprettelse, login, opdatering og sletning.
- CatalogService: Test af CRUD på Catalog items.
- BiddingService: Test af budhåndtering og -registrering.
- AuctionService: Test af auktionshåndtering og -opdatering.
- End-to-End test: Tester håndtering af en auktion fra start til slut, med brug af alle services

Testcases

For hver testgruppe har vi defineret en række testcases, der dækker forskellige scenarier og grænsetilfælde. Disse omfatter:

AuthService:

- Test af korrekt login med gyldige brugeroplysninger.
- Test af forkert login med ugyldige brugeroplysninger.
- Test af autorisation af adgang til beskyttede ressourcer.

UserService:

- Test af brugeroprettelse med gyldige oplysninger.
- Test af opdatering af brugeroplysninger.
- Test af sletning af bruger.

CatalogService:

- Test af oprettelse af katalogposter.
- Test af opdatering af katalogposter.
- Test af sletning af katalogposter.

BiddingService:

- Test af indsendelse af bud.
- Test af korrekt validering af bud.

AuktionService:

- Test af oprettelse af auktioner.
- Test af opdatering af højeste bud på en auktion.
- Test af hentning af specifik auktion.

Postman teste med scripts

Download vores to .json filer "Postman Collection - Auktionshus APIs" og "Postman Environment - Azure" som kan findes under "Aflevering" - "Tests" på vores GitHub, eller tilgå den med linket her:

<https://github.com/Hablas-Si/AuktionshusAflevering/tree/main/Tests>

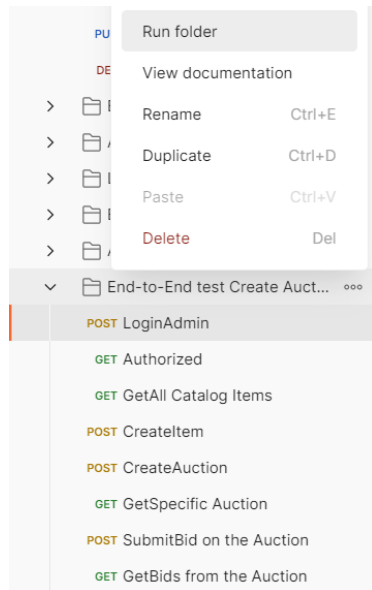
Åben Postman, og importer de to ovenstående filer under "Import"

Du har nu importeret alle vores API tests i Postman, samt de miljøvariabler vi benytter i vores opstillede API kald. Du kan nu vælge at køre dem igennem én af gangen, eller køre hele kollektionen på én gang ved at vælge hele kollektionen og trykke "run collection".

Vi har brugt environment variabler og scripts, for at gøre det nemmere at teste API endepunkterne. Disse scripts udfylder kaldenes body automatisk, men hvis du kører de enkelte kald manuelt kan du selv foretage ændringer.

En vigtig pointe: Ofte når en service, som har været inaktiv i en periode, modtager en http request, tåler det at tage et stykke tid at udføre requesten. Hvis den fejler og returnerer error: "504 Gateway Time-out", skal man blot sende requesten igen, og så burde det fungere.

Vi anbefaler at starte med vores "End-to-End test Create Auction", og enten kører dem én af gangen fra top til bund, eller vælger at køre hele mappen som kan ses på nedenstående billede:



Hvis du vil teste alle endepunkter individuelt, så skal du starte med at generere en Admin JWT-token. Dette gør du ved at åbne den første mappe i kollektionen: "AuthService". Du skal kalde på den post-request der hedder "LoginAdmin - Valid", da denne har korrekte login oplysninger til en admin bruger. Herefter er der et script som sætter den nye genererede JWT token i en environment-variabel, som bliver brugt i alle de andre requests.

Prøv nu at køre de forskellige requests, hvis du er i tvivl omkring formålet ved nogle af kaldene, kan du læse yderligere om dem ved at gå dokumentationen af kollektionen:

- Tryk på prikkerne ud fra en af metoderne
- View documentation.

Manuelle End-to-end tests i Postman

OBS: I dette afsnit anvendes *ikke* automatiserede scripts og variabler som i det ovenstående. Du skal altså selv skrive bodies, URL'er og indsætte auth-tokens.

LoginAdmin

Først skal man logge ind på en gyldig admin-bruger for at generere en JWT-token.

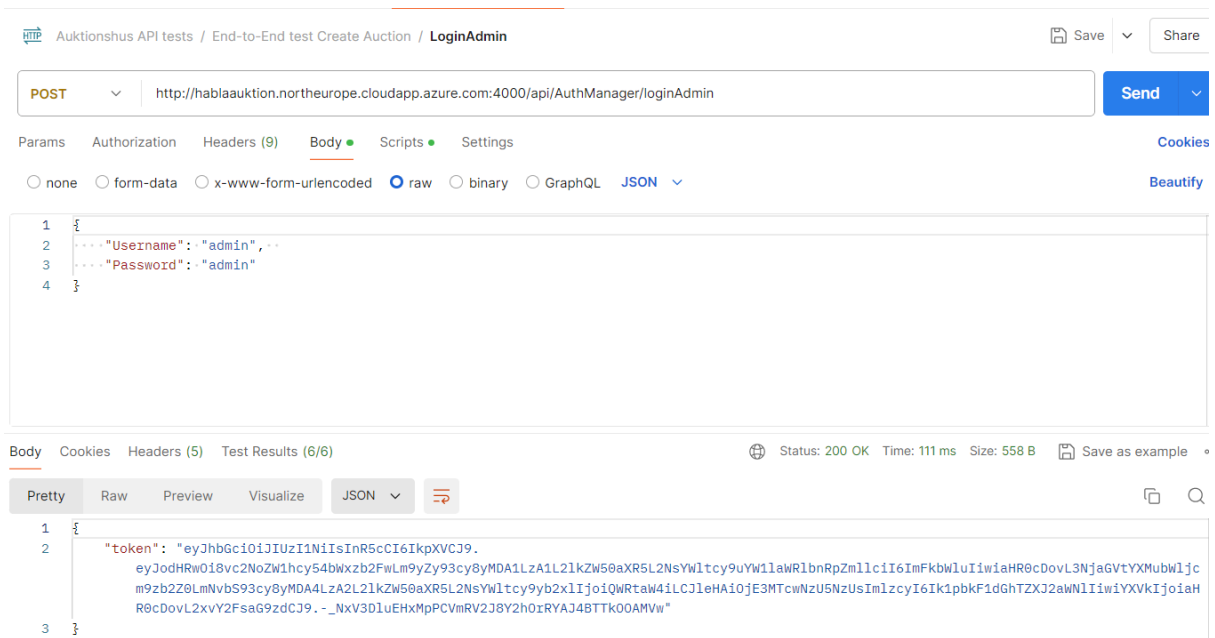
Send POST-request til:

<http://hablaauktion.northeurope.cloudapp.azure.com:4000/api/AuthManager/loginAdmin>

Body skal være "Raw" - JSON:

```
{
  "Username": "admin",
  "Password": "admin"
}
```

Send denne request, og en JWT-token bliver genereret i console-body. Denne token skal bruges for at kunne udføre de andre kald. Den skal indsættes under "Authorization" - "Auth Type": "Bearer Token".



CreateItem

Nu kan du oprette en item i catalog, så længe du har tilføjet din JWT-token fra tidligere.

Send POST-request til:

<http://hablaauktion.northeurope.cloudapp.azure.com:4000/api/Catalog>

Eksempel på Body (skal være "Raw" - JSON):

```
{
  "title": "Ny catalog item",
  "price": 129,
  "description": "En ny item som skal oprettes i catalog",
  "productAvailable": true,
  "seller": "Admin"
}
```

The screenshot shows a REST client interface with the following details:

- URL:** `http://hablaauktion.northeurope.cloudapp.azure.com:4000/api/Catalog`
- Method:** `POST`
- Body:** A JSON object: `{ "title": "Ny catalog item", "price": 129, "description": "En ny item som skal oprettes i catalog", "productAvailable": true, "seller": "Admin" }`
- Status:** 200 OK
- Time:** 5.78 s
- Size:** 356 B
- Response Body:** A JSON object: `{ "id": "11b2ce3b-de8f-4495-b788-8c902c589b01", "title": "Ny catalog item", "description": "En ny item som skal oprettes i catalog", "price": 129, "productAvailable": true, "seller": "Admin" }`

CreateAuction

Nu kan du oprette en auction til det nyoprettede item, ved at bruge det genererede itemID så længe du har tilføjet din JWT-token fra tidligere.

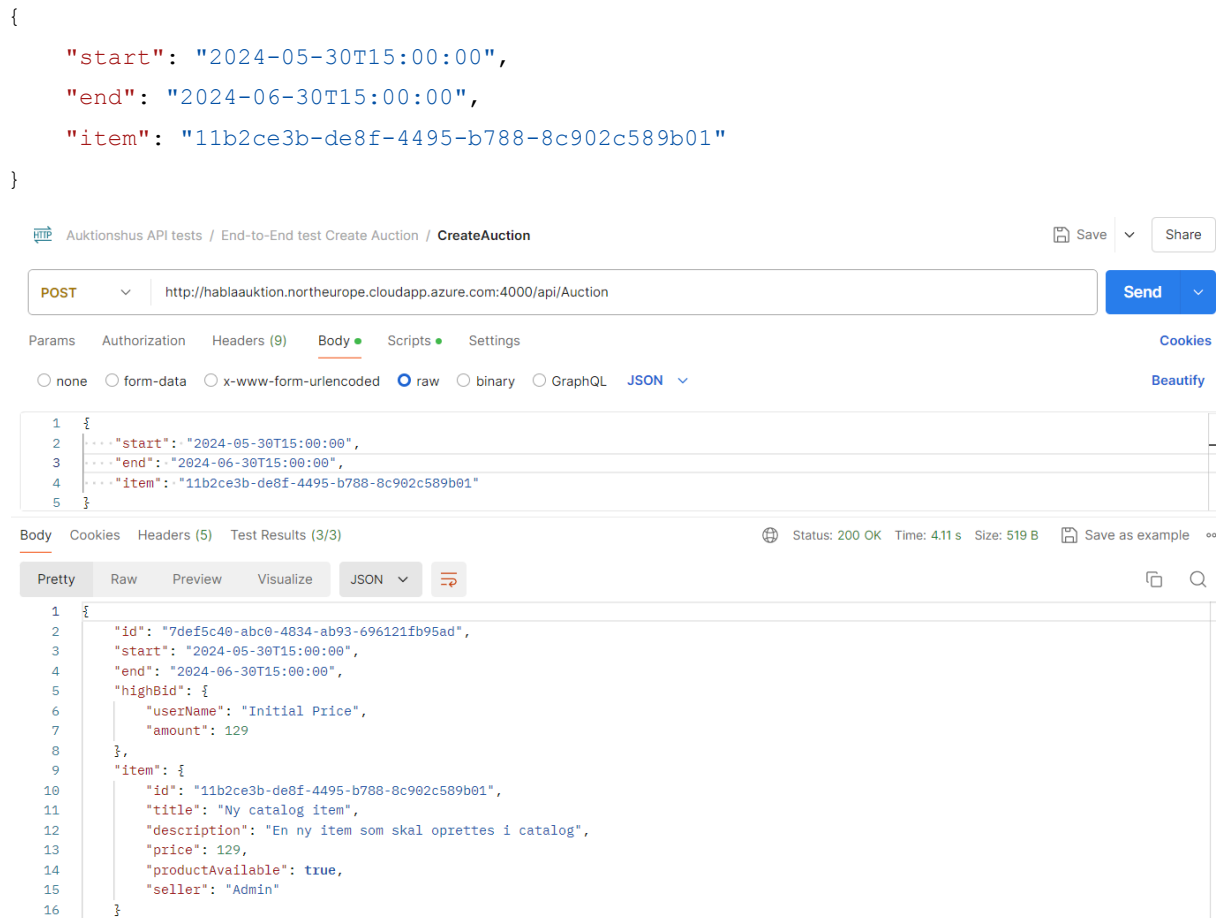
Send POST-request til:

<http://hablaauktion.northeurope.cloudapp.azure.com:4000/api/Auction>

Eksempel på Body (skal være "Raw" - JSON):

Guide for tests til Auktionsplatform for 4. Semester projekt af gruppen Hablas Si#

```
{
  "start": "2024-05-30T15:00:00",
  "end": "2024-06-30T15:00:00",
  "item": "11b2ce3b-de8f-4495-b788-8c902c589b01"
}
```



The screenshot shows a REST client interface with the following details:

- URL:** `http://hablaauktion.northeurope.cloudapp.azure.com:4000/api/Auction`
- Method:** POST
- Body:**

```
{
  "start": "2024-05-30T15:00:00",
  "end": "2024-06-30T15:00:00",
  "item": "11b2ce3b-de8f-4495-b788-8c902c589b01"
}
```
- Response:** Status: 200 OK, Time: 4.11 s, Size: 519 B. The response body is a JSON object:

```
{
  "id": "7def5c40-abc0-4834-ab93-696121fb95ad",
  "start": "2024-05-30T15:00:00",
  "end": "2024-06-30T15:00:00",
  "highBid": {
    "userName": "Initial Price",
    "amount": 129
  },
  "item": {
    "id": "11b2ce3b-de8f-4495-b788-8c902c589b01",
    "title": "Ny catalog item",
    "description": "En ny item som skal oprettes i catalog",
    "price": 129,
    "productAvailable": true,
    "seller": "Admin"
  }
}
```

SubmitBid on the Auction

Nu kan du oprette et bud på den nyoprettede auktion, ved at henvise til

Send POST-request til:

<http://hablaauktion.northeurope.cloudapp.azure.com:4000/api/Auction>

Eksempel på Body (skal være "Raw" - JSON):

```
{
  "Auction": "7def5c40-abc0-4834-ab93-696121fb95ad",
  "BidOwner": "Admin",
  "Amount": 500
}
```

Guide for tests til Auktionsplatform for 4. Semester projekt af gruppen Hablas Si#

Auktionshus API tests / End-to-End test Create Auction / **SubmitBid on the Auction** Save Share

POST http://hablaauktion.northeurope.cloudapp.azure.com:4000/api/Bidding Send

Params Authorization Headers (9) **Body** Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** Beautify

```
1 {
2   "Auction": "7def5c40-abc0-4834-ab93-696121fb95ad",
3   "BidOwner": "Admin",
4   "Amount": 500
5 }
```

Body Cookies Headers (5) Test Results (3/3) Status: 200 OK Time: 219 ms Size: 359 B Save as example

Pretty Raw Preview Visualize **JSON** Copy Search

```
1 {
2   "id": "399fc536-b2d4-457d-bb44-ec5f0b2276ed",
3   "bidOwner": "Admin",
4   "amount": 500,
5   "auction": "7def5c40-abc0-4834-ab93-696121fb95ad",
6   "accepted": true,
7   "created": "2024-05-30T15:26:31.649271Z"
8 }
```