

Alla endpoints börjar med "http://localhost:3000".

1.Som besökare vill jag kunna se sammanfattade auktionsobjekt som en lista.

get("/data/objekt/summary-list"

man gör bara en get med endpointen så får man en sammanfattande lista på auktionsobjekt

2.Som besökare vill jag kunna se detaljer för varje auktionsobjekt.

.get("/data/objekt/details"

En get med endpointen ger en detaljerad lista på auktionsobjekt

3.Som besökare vill jag kunna söka på auktioner baserat på vad jag skriver i ett sökfält.

Endpoint: /data/search/:string

Exempel: <http://localhost:3000/data/search/iphone>

4.Som besökare vill jag kunna se nuvarande bud på auktionsobjekt i listvyer

get("/data/objekt/bid-list"

en get med denna endpointen visar en lista på auktionerna med senaste buden

5.Som besökare vill jag kunna se nuvarande bud på auktionsobjekt i detaljsidor.

endpoint: /data/bud/:objekt_id

GET

exempel: <http://localhost:3000/data/bud/1>

6.Som besökare vill jag kunna registrera ett nytt konto och bli användare.

Endpoint: /data/anvandare

POST Exempel: <http://localhost:3000/data/anvandare>

```
{
  "namn": "Ruth Bader",
  "efternamn": "Ginsburg",
  "anvandarnamn": "rubin",
  "telefonummer": "+1888279916",
  "adress": "82 Queensway",
  "potskod": "43957",
  "ort": "London",
  "mail": "ruthbad@gamil.com",
  "bild": "rubagin"
}
```

7. Som användare vill jag kunna logga in.

Börja med att göra en POST i postman med endpoint "http://localhost:3000/data/login" sedan gör du en input som följande

Exempelvis:

```
{  
  "anvandarnamn": "Lilo",  
  "losenord": "lilo123"  
}
```

för att logga in.

Om värdena i posten inte existerar så skrivs följande ut

```
{  
  "status": "Wrong Username/Password"  
}
```

8.Som användare vill jag kunna lägga (högre än nuvarande) bud på auktionsobjekt på dess detaljsida.

10.Som användare ska jag inte kunna lägga bud på mina egna auktionsobjekt.

Endpoint: /data/bud

POST

Exempel: <http://localhost:3000/data/bud>

```
{  
  "objekt_id": "5",  
  "bud_pris": 99999,  
  "bud_givare": 1,  
  "bud_tid": "2022-02-10 03:49:02"  
}
```

9. Som användare vill jag kunna skapa nya auktionsobjekt.

11. Som användare vill jag att auktionsobjekt ska innehålla minst titel, beskrivning, starttid, sluttid och bild(er).

12. Som användare vill jag kunna sätta ett utgångspris på mina auktionsobjekt.

Gör en **POST** i postman med endpoint `"/data/new_auction_object"` med input som tex den här.

```
{
  "saljare": 5,
  "beskrivning": "Ett magiskt test objekt",
  "titel": "Magiskt Ägg",
  "kategori": 6,
  "start_tid": "2022-05-12 12.30.00",
  "slut_tid": "2022-05-15 12.30.00",
  "bild": "För magisk att se på!",
  "start_pris": 1000,
  "dold_slutpris": 1000000,
  "status": 1
}
```

13. Som användare vill jag kunna sätta ett dolt reservationspris på mina auktionsobjekt.

(om bud ej uppnått reservationspris när auktionen avslutas så säljs objektet inte).

Gör en **POST** i postman med endpoint `"/data/auctionista/objekt"`

för att hämta ut ett id som innehåller olika columner med olika värden,

exempel :

```
{
  "id" : 10
}
```

I detta fallet pris på objekt samt erbjudanden av objekt

Beroende på vilket id du väljer så kommer antingen objekten vara sålda till dem högsta budgivaren eller så är objekten fortfarande till salu i väntan på att få ett erbjudande som matchar användarens villkor.

14.Som besökare vill jag kunna se auktioner inom kategorier.

15.Som besökare vill jag kunna söka på auktioner inom en kategori jag valt.

Här går det bra att använda postman eller adressfältet i en webbläsare som tex chrome.

För att se eller söka på auktioner baserat på vilken kategori de tillhör så kan man använda följande endpoint **"/data/category/?"** där ? ersätts med en av följande kategorier:

cykel

mobil

djur

bil

klader&skor (Kläder & skor)

fiske

sport

mobler (Möbler)

verktyg

dator

teknik

Det går även bra med att ersätta ? med id numret **(1-11)** som varje kategori har blivit tilldelad.

16.Som besökare vill jag kunna se auktioner baserat på status (pågående, avslutade, sålda, ej sålda).

Här går det bra att använda postman eller adressfältet i en webbläsare som tex chrome.

För att se eller söka på auktioner baserat på vilken status de har så kan man använda följande endpoint **"/data/status/?"** där ? ersätts med en av följande status:

ongoing (Pågående)

end_received (Avslutad och mottagen)

end_shipped (Avslutad och skickad)

end_not_sold (Avslutad men ej såld)

Det går även bra med att ersätta ? med id numret **(1-4)** som varje status har blivit tilldelad.

17.Som användare vill jag kunna se en lista med mina egna auktionsobjekt.

Endpoint: **/data/ mina-auktioner/:id**

GET Exempel: <http://localhost:3000/data/mina-auktioner/1>

18.Som användare vill jag kunna se en lista med auktionsobjekt jag har lagt bud på.

.get("/data/:anvandare/mina_bud",

Man gör en get med endpointen så får man den sammanfattande listan på auktionsobjekt som jag lagt bud på.

20.Som användare vill jag ha en publik profilsida där namn, publika kontaktuppgift(er) & bild visas för andra att läsa.

`get("/data/anvandare/:id"`

för att komma åt en specifik profil, ersätt ":id" med en siffra. Siffran representerar en användares id.

21.Som användare vill jag att min publika profilsida innehåller information om hur många auktioner jag köpt och sålt.

Gör en POST i postman med endpoint `"/data/:anvandare/antal_kopta_objekt"`

för att hämta ut ett id som innehåller antal köpte objekt eller

`"/data/:anvandare/antal_salda_objekt"` för att hämta ut ett id som innehåller antal sålda objekt

Gör en post i Postman och skriv in vilket id du vill hämta information av, exempelvis:

```
{  
  "id" : 5  
}
```

22.Som köpare vill jag kunna ge ett betyg efter köp av ett auktionsobjekt.

23.Som säljare vill jag kunna ge ett betyg efter försäljning av ett auktionsobjekt.

endpoint: `/data/anvandare/betygsatta`

POST

exempel:

```
{  
  "anvandare_id": 5,  
  "betyg_givare": 67  
  "betyg": 4  
}
```

24.Som användare vill jag kunna se säljares betyg när jag tittar på ett auktionsobjekt.

Endpoint: `/data/saljarens-betyg/:objektId`

GET Exempel: <http://localhost:3000/data/saljarens-betyg/3>

25. Som användare vill jag kunna skicka meddelande till en säljare av ett auktionsobjekt.

Den första kontakten i en ny konversation görs alltid med denna **POST**, alla svar från båda parter görs i **PUT** metoden som förklaras i nästa del (26).

Gör en **POST** i postman med endpoint **`/data/conversation/new_conversation`** med input som tex den här.

```
{
  "sender": 3,
  "receiver": 46,
  "message": "Finns det några skador på datorn?",
  "object": 10
}
```

26. Som säljare av ett auktionsobjekt vill jag kunna svara på meddelande från användare.

Alla svar i en konversation görs med denna **PUT**, för att starta en ny konversation så används metoden för **POST** i exemplet ovan (25).

Gör en **PUT** i postman med endpoint **`/data/conversation/reply_?`** där ? ersätts med id numret som konversationen har blivit tilldelad. Den går att hitta i databasen.

Ett exempel på hur ett input ser ut ser du här.

```
{
  "message": "Nej, datorn är helt oskadd."
}
```

Pröva gärna att skriva några meddelande och kopiera sen klistra in konversationen från databasen till ett textdokument som tex anteckningar eller Google docs.