

Användardokumentation för utvecklare

Öppna Bibliotek

Stockholms stadsbibliotek 19 april 2012 API version 3.5.0

Innehåll

1 Inledning	3
1.1 Hantering av information på ÖB	3
1.2 Om dokumentet	3
2 Arkitektur	4
2.1 Struktur	4
3 Om anrop	5
3.1 Anropsmetoder	5
3.2 Resultat	5
3.3 Representation	5
3.4 Felhantering	5
4 Parametrar till objekt som finns i ÖB	7
5 Sökning i ÖB	9
5.1 Sökning med angivna index	9
5.2 Fritextsökning	9
5.3 Begränsning av antal sökresultat	9
5.4 Returnerat resultat	10
6 Speciella funktioner	11
7 Länka till bibliotek och skribenter	12
8 Roller och behörigheter	13
9 Dokumentation för vissa API-anrop	14
9.1 Hantera omdömen	14
9.2 Hantera taggar och taggningar	17
9.3 Missbrukshantering	21
9.4 Hantera bibliotek	22
9.5 Hantera användare	23
9.6 Statistik	25
9.7 Hantera databas	27
10 Generellt om söka & sortera	28
10.1 Söka	28
10.2 Sortera	28
11 Praktisk information för utvecklare	29
11.1 Utvecklarlicens	29
11.2 Administratörer	29
11.3 Buggar och tekniska problem	29
11.4 Bakåtkompatibilitet	29
12 Tredjepartsverktyg och klientbibliotek	30
12.1 Klient-bibliotek skrivet i PHP	30
12.2 Drupal-modul för betygsättning och recensering av verk	30
12.3 Drupal-modul för administration av användarproducerat material	30
12.4 Exempel på implementationer	30

1 Inledning

Målet med Öppna Bibliotek (ÖB) är att göra mervärdesinformation tekniskt, upphovsmässigt och ekonomiskt, så lättillgängliga och öppna som möjligt.

Exempel på mervärden är till exempel material som personal på biblioteken producerar såsom beskrivningar och tips av böcker, musik, film etc. Ett annat exempel är mervärden som låntagaren kan sätta på verk, till exempel att användaren kan sätta betyg, omdömen och etiketter/taggar på böcker. Detta kan komma att utvidgas med funktioner för teman och listor.

1.1 Hantering av information på ÖB

Användningen av information som delas via Öppna Bibliotek styrs av en Creative Commons-licens som godkänner återpublicering i icke-kommersiella sammanhang så länge upphovsmannen anges och texten inte bearbetas. Vissa tilläggskrav finns.

Läs mer på <http://api.oppnabibliotek.org/licens.html>

1.2 Om dokumentet

1.2.1 Dokumenthistorik

Datum	Ansvarig	Beskrivning
4/5-09	Erik Lindblad	Lagt till behörighet i metod-anropen
7/5-09	Erik Lindblad	Lagt till information om utvecklar nyckel Lagt till descriptions till missbrukshandtering Lagt till metod: Visa antal taggar
23/6	Erik Lindblad	Uppdaterat dokumentet efter telefonmöte.
22/10-09	Daniel Andersson	Version 3.3 Layout korrigerad
22/10-09	Daniel Andersson	Version 3.4 Alla ändringar accepterade
10/12 -09	Daniel Andersson	Version 3.4.1 fel ang. update-kommandot rättat
10/12 -09	Daniel Andersson	Version 3.4.2 uppladdat till Google Docs
22/09 -11	Fredrik Wollsen	Konverterad till det senaste Google Docs formatet och Layout-korrigerad
22/09 -> 28/09 - 11	Fredrik Wollsen	Allmänna förtydliganden, korrigeringar och utvidgade exempel inför lansering av version 3.5.0
14/12 - 11	Fredrik Wollsen	La till information om den nya domänstrukturen och information kring bakåtkompatibilitet. Version 3.5.0 lanseras.
April 2012	Fredrik Wollsen	Upprensning och uppdatering inför publicering av dokumentation

2 Arkitektur

ÖB använder en REST-arkitektur för alla operationer som kan göras. Detta innebär att alla resurser som man vill studera eller ändra kan komma åt genom http/https-anrop ("webben"). Från anropen returneras XML-dokument som beskriver det data som hämtats eller modifierats.

För att använda ÖBs API kan man alltså använda vilket programmeringsspråk som helst som innehåller bibliotek för http/https-anrop och XML-hantering, vilket är alla vanligt förekommande programmeringsspråk.

2.1 Struktur

Alla objekt som man vill komma åt har sin egen url-bas på ÖB. T.ex. har alla böcker url:en /books, alla recensioner url:en /descriptions, etc. Individuella objekt av en viss typ kan komma åt genom att man anger objektets id. Detta kanske enklast illustreras genom några exempel:

Exempel:

- Samtliga böcker i systemet:

<http://api.oppnabibliotek.org/books>

- Samtliga recensioner/mediabeskrivningar i systemet

<http://api.oppnabibliotek.org/descriptions>

- Boken med id 1

<http://api.oppnabibliotek.org/books/1>

- Recensionen med id 23

<http://api.oppnabibliotek.org/descriptions/23>

3 Om anrop

3.1 Anropsmetoder

Alla resurser/url:er i systemet kan betraktas eller modifieras genom någon av de fyra HTTP-anropsmetoderna GET, POST, PUT eller DELETE

Exempel:

- Visa alla böcker i systemet
GET /books
- Visa bok med ID 1 i systemet
GET /books/1
- Lägg till en bok till systemet, med bifogat data som attribut
POST /books
- Förändra bok med ID 1, med bifogat data som attribut
PUT /books/1
- Ta bort bok med ID ur systemet
DELETE /books/1

3.2 Resultat

Anrop som är korrekta och som begärt en representation av en resurs besvaras med ett XML-dokument (eller HTML-dokument i de flesta fall) som beskriver det data som hämtats. T.ex. får man ett XML-dokument som visar data för alla böcker om man går till /books. Varje bok beskrivs där, och man kan alltså lätt (m.h.a. ett XML-bibliotek) hämta ut exempelvis att det finns en bok med id 23, och sedan gå till /books/23 för att hämta mer information om just den boken. Den XML som returneras för en individuell bok innehåller även alla utgåvor och alla recensioner som hör till den boken, och man kan sedan hämta ut exempelvis alla recensionstexter för boken med id 23 genom att gå till /books/23 och plocka ut de XML-element som visar recensionstexter

3.3 Representation

Anrop att styra vilken representation man vill ha tillbaka från ett anrop, bifogar man http-headern accept: med värdet application/xml. Man får då en xml-representation tillbaka. I vissa fall går det också att få tillbaka en html-vy, men det är xml-representationen som är den kompletta och korrekta. Html-vyn får man om man använder en webbläsare för att skriva in adresserna. Använder man en webbläsare och ändå vill se XML-representationen, kan man avsluta varje adress med .xml, d.v.s. /books.xml, /books/1.xml, etc.

En hel del nyckelanrop finns även som Atom-flöde (RSS) läs mer under "Speciella funktioner" nedan.

3.4 Felhantering

Anrop som gått fel signaleras främst genom returkoder på de http-anrop som skett, enligt nedan.

200-serien - innebär att allt är OK.

- 200 OK

Kod som signalerar att allt gått bra

- 201 Created

Returneras när man begärt att ett objekt skapas och det har gått bra.

400-serien - innebär felaktigt anrop

- 400 Bad Request

De parametrar som skickats till anropet är inte korrekta

- 401 Unauthorized

Den användare som skickat anropet har antingen skickat fel användarnamn/lösenord, eller har inte tillåtelse att utföra det begärda anropet.

- 404 Not Found

Den resurs som anropet utförts på finns inte.

500-serien - innebär ett oväntat fel på server-sidan

- 500 Internal Server Error

Detta fel signalerar att något gått snett på server-sidan som klienten inte kan reda ut, t.ex. orsakat av en bugg.

4 Parametrar till objekt som finns i ÖB

Det finns en parameter som måste skickas med vid ALLA anrop av formatet xml. Det är utvecklar nyckeln (dev_key) för biblioteket som den anropande användaren tillhör. Utan denna parameter kommer anropet till ÖB alltid att misslyckas.

Exempel (ett anrop som skulle kunna komma från exempelvis Stockholms Stadsbibliotek):

http://api.oppnabibliotek.org/books/search.xml?dev_key=ssb123

Denna utvecklar nyckel är obligatorisk men har utelämnats i alla exempel i det här dokumentet för att göra det mer lättläst.

På alla objekt som finns utexporterade kan man göra en viss utsökning genom att ange värden på attribut i objektet, enligt exempel nedan. De attribut som kan användas är alla attribut som ett visst objekt har (se listan på dessa attribut under "Sökning med angivna index", samt i Appendix 2).

I alla utsökningar gäller att filtrering sätts som ett attribut inom klamrar på det objekt man söker på, d.v.s. listar man taggings så anger man library_id som tagging[library_id]=...

Ytterligare ett exempel, om det står:

objekt: assessment

Med en parameter "isbn", så ska parametern skickas som sådan:
assessment[isbn]=9999999999

Exempel:

[http://api.oppnabibliotek.org/descriptions?description\[user_id\]=6](http://api.oppnabibliotek.org/descriptions?description[user_id]=6)

[http://api.oppnabibliotek.org/editions?edition\[published\]=0](http://api.oppnabibliotek.org/editions?edition[published]=0)

Datum

På följande objekt: assessment, book, description, edition och tagging kan man ange ett date_from och ett date_to, för att filtrera på en viss period.

Exempel: /taggings.xml?tagging[date_from]=2009-05-28

Search.xml

För sök-funktionen på books (/books/search.xml) gäller att alla attribut man söker på läggs till övriga parametrar (i valfri ordning).

Exempel:

<http://api.oppnabibliotek.org/books/search.xml?authorlastname=andersson>

Limit, Order

Dessutom kan man använda "order" (plus "reverse", "limit" och "offset"). Dessa sätts INTE inom klamrar.

Exempel:

- Alla böcker (men med default-antal som visas satt till 50)

<http://api.oppnabibliotek.org/books>

- Alla böcker (visa 100 stycken)

<http://api.oppnabibliotek.org/books?limit=100>

- Alla beskrivningar, ordnade i omvänd tidsordning (d.v.s. senast först)

http://api.oppnabibliotek.org/descriptions?order=updated_at&reverse=true

För samtliga url:er med parametrar gäller att den första parametern föregås av frågetecken och alla följande parametrar med och-tecken.

Exempel: <http://api.oppnabibliotek.org/books/search.xml?authorlastname=strindberg&authorfirstname=august>

För alla parametrar som berör hur resultatet ska presenteras (limit, order, offset och reverse) gäller att dessa parametrar läggs till övriga parametrar (i valfri ordning).

Exempel:

<http://api.oppnabibliotek.org/books/search.xml?authorfirstname=jan&limit=2>

Det går att lista användare och verk utifrån aktivitet med följande sorteringsparametrar:

activity_assessments antal omdömen

activity_grades betygsmedelvärde

activity_taggings antal etiketter

Dessa går också att kombinera med datumutsökning.

Exempel:

Lista de 10 användare som har satt flest omdömen:

http://api.oppnabibliotek.org/users.xml?order=activity_assessments&limit=10

Lista de 10 användare som har satt högst betyg, skapade den senaste veckan:

[http://api.oppnabibliotek.org/users.xml?order=activity_grades&user\[date_from\]=2009-05-25&limit=10](http://api.oppnabibliotek.org/users.xml?order=activity_grades&user[date_from]=2009-05-25&limit=10)

Lista de 10 böcker som har flest omdömen:

http://api.oppnabibliotek.org/books.xml?order=activity_assessments&limit=10

Filtrering på frekvens av taggar. Det går inte att välja ut de taggar som bara använts en gång. Däremot går det att sortera listningen av taggar efter hur mycket de används: /tags.xml?order=byuse.

Listningar av omdömen för ett verk kan inte sorteras på användarid eller utgåva.

5 Sökning i ÖB

Utöver den allmänna sökfunktionen som beskrivits ovan finns tre utökade sökningar kopplat till verk och dessas utgåvor och beskrivningar: sökning med specificerade index, fritextsökning med mediebeskrivningstexten utesluten, och fritextsökning med mediebeskrivningstexten inkluderad. Samtliga dessa tre sökningar utgår från adressen <http://api.oppnabibliotek.org/books/search>

5.1 Sökning med angivna index

Flera index kan anges. Endast träffar som uppfyller samtliga angivna kriterier returneras.

Möjliga sök-index är fördelade enligt följande:

- **verk**: title, booktitle_part1, booktitle_part2, authorfirstname, authorlastname, group_inst, signumname, keyword, sb_keyword, agegroupname, targetgroupname, reserved
- **utgåva**: year, isbn, recordnr, recordcompany, translator, illustrator, auxcreator, image, mediatype, mediatypecode, published, manual
- **mediebeskrivning**: libraryname, text, username, userfirstname, userlastname

Exempel:

- <http://api.oppnabibliotek.org/books/search?authorlastname=Strindberg>
- <http://api.oppnabibliotek.org/books/search?authorlastname=Strindberg&authorfirstname=August>
- <http://api.oppnabibliotek.org/books/search?isbn=9120071116>
- <http://api.oppnabibliotek.org/books/search?authorlastname=Strindberg&username=kary65>

5.2 Fritextsökning

Skillnaden mellan de båda fritextsökningarna är att mediebeskrivningstexten antingen inkluderas eller inte. Fritextsökningar indikeras med index freequery1 (utan mediebeskrivningstext) och freequery2 (med texten). Om man använder någon av fritextsökningarna kan man inte kombinera dem med något annat index.

Exempel:

- <http://api.oppnabibliotek.org/books/search?freequery1=Strindberg>
- <http://api.oppnabibliotek.org/books/search?freequery2=Jesper>
- <http://api.oppnabibliotek.org/books/search?freequery2=kvinnor:Strindberg>

*(I detta fall betyder ":"-tecknet logiskt "OCH", d.v.s. resultat som innehåller både **kvinnor** och **Strindberg** returneras.)*

5.3 Begränsning av antal sökresultat

Resultat från alla sökningar är begränsat till 10 verk om man inte anger en limit-parameter. Det går att söka helt utan att ange något sökresultat. Alla verk returnas då, men antalet och utsnittet begränsas av parametrarna limit och offset (se nedan).

Exempel:

<http://api.oppnabibliotek.org/books/search?authorlastname=lindgren&limit=20>

Sortering av resultat

Sökresultaten kan ordnas efter titel, författarnamn, utgivningsår, och recensionsdatum, genom att ange parametern order och argumenten "title", "author", "year" respektive "date". Ordningen kan kombineras med parametern "reverse" för att vända ordningen på sökresultatet.

Ordningen fungerar också tillsammans med parametrarna limit och offset, vilket gör det möjligt att dela upp en sökning som ger ett stort antal träffar i många olika delar, och hämta en del i taget. (T.ex. hämta tio i taget och låta användaren navigera sig framåt i resultatet).

Exempel:

- <http://api.oppnabibliotek.org/books/search?authorlastname=lindgren&limit=20&order=date&reverse=true>
- <http://api.oppnabibliotek.org/books/search?keywords=deckare&limit=10&offset=10>

5.4 Returnerat resultat

Resultaten från fritextsökningarna returnerar en "djup" version av ingående verk som hittats, d.v.s. varje matchande verk finns med i retur-xml:en, och varje verk har alla utgåvor och alla mediebeskrivningstexter (oavsett om dessa matchade eller ej).

Resultatet från index-sökningen filtreras däremot enligt följande:

- Sökterm kopplad till verk: Om det finns söktermer som är kopplade till verk, returneras endast de verk (books) som matchar, inkluderande utgåvor och mediebeskrivningar.
- Sökterm kopplad till utgåva: Om det finns söktermer som är kopplade till utgåva, returneras de verk som inkluderar matchande utgåvor, men med de utgåvor som inte matchar bortfiltrerade.
- Sökterm kopplad till mediabeskrivning: Om det finns söktermer som är kopplade till mediebeskrivning, returneras de verk som inkluderar matchande mediebeskrivningar, inkluderande utgåvor och de mediebeskrivningar som matchar.
- Kombinationen av flera index på olika nivåer gör att retur-xml:en kan filtreras på flera nivåer, d.v.s. xml-trädet har vissa utgåvor och mediebeskrivningar bortfiltrerade.
- Tomma objekt returneras inte, d.v.s. en utgåva som inte har någon mediebeskrivning kvar efter filtrering visas inte, och ett verk som inte har några utgåvor kvar efter filtrering returneras inte.

6 Speciella funktioner

OPAC-länk

För de bibliotek som har en dynamisk söksträng inlagd kan man anropa funktionen dynurl så här: GET http://api.oppnabibliotek.org/libraries/1/dynurl.xml?edition_id=1. Detta är alltså ett anrop till biblioteket med id 1, där man ber att få länken till utgåvan med id 1. Resultatet som returneras innehåller den dynamiska söksträngen ifylld med fält från den aktuella utgåvan. Ett exempel på resultat från detta anrop: <dynurl>http://www.gotlib.goteborg.se/search*swe/i?SEARCH=9100102172</dynurl>
(OBS dynURL har ingen HTML-vy - bara XML)

All authors

Man kan få ut en lista på alla författare genom följande anrop: GET <http://api.oppnabibliotek.org/books/authors.xml>
(OBS - har ingen HTML-vy - bara XML)

Atom/RSS

Alla intressanta objekt kan kommas åt som atom-feeds. En atom-länk tar samma parametrar som övriga anrop, så om man t.ex. vill prenumerera på mediebeskrivningar kan man använda följande länk: http://api.oppnabibliotek.org/descriptions.atom?order=updated_at&reverse=true

By username

Om man enbart känner till en användares användarnamn, och inte dennes (interna) id, kan man anropa metoden users/byusername med användarnamnet som parameter. Givet att man har rättigheter att se informationen, får man då tillbaka ett XML-dokument som beskriver användarens attribut.

By bookproperty

Om man vill ha en listning av mediebeskrivningar, filterat på ett attribut som sitter på verksnivå (t.ex. boktitel), kan man använda metoden descriptions/bybookproperty. Resultatet sorteras i omvänd datumordning för mediebeskrivningarna.

Statistikfunktion för bibliotek

Metoden libraries/bydescriptions kan anropas för att få en statistik över antalet mediebeskrivningar per bibliotek.

Användningsfrekvens för nyckelord

Metoden keywords/byuse kan anropas för att få en listning av nyckelord, ordnade efter användningsfrekvens.

Tvättning av recensionstexter

Om en skribent klistrar in en recensionstext från exempelvis Microsoft Word, kan mycket html-markup följa med i form av Word-specifika html-taggar. En tvättning av denna text görs på API-sidan, så att enbart en begränsad mängd taggar sparas i texten. De taggar som tillåts vara kvar är: <a href>, ,
, <i>, <p> och <hr>. Material som sparas till Öppna Bibliotek skall inte innehålla otillåtna taggar men denna "tvättning" finns som extra säkerhet.

7 Länka till bibliotek och skribenter

Länkning av biblioteks namn skall ske om biblioteket har en URL i < libraryinfolink > så skall namnet på biblioteket länkas till denna adress.

Detsamma gäller även skribentens namn <user> om information finns i <dynamicinfolink> så skall skribentens namn länkas.

Skribentlänkning har två alternativ;

· Absolut individuell länk: Varje enskild user/skribent kan ha en egen absolut URL i sin <dynamicinfolink>, denna skall användas för länkning förutsatt att det inte finns en dynamisk URL angivet för det bibliotek man tillhör <userinfolink>.

· Dynamisk skribentlänk: Om det finns ett värde för biblioteket i <userinfolink> så skall istället denna gälla och kombineras* med det värde som finns angiven för varje skribent i <dynamicinfolink>.

*Med kombineras menas "läggas till efter". dvs <userinfolink><dynamicinfolink> vilka tillsammans genererar en fullständig fungerande URL till skribenten ifråga.

För dig som även skapar redaktionellt material till Öppna Bibliotek: <dynamicinfolink> kan även administreras på boktips.net och kallas där "Informationssida" dock saknas i skrivande stund (080813) möjlighet att lägga in uppgifter om <userinfolink> på boktips.net, stöd i ÖppnaBiblioteks APier finns däremot.

För mer information om utformningen av länkningen se även <http://api.oppnabibliotek.org/licens.html>

8 Roller och behörigheter

Följande roller och behörigheter gäller för ÖB: (inom hakparanteser motsvarande role_id)

Roll	Behörighet
Allmänhet (reader [1])	Alla med tillgång till Internet kan läsa publicerad mervärdesinformation på någon biblioteks-sajt.
Låntagare (member [5], writer [2])	Person med lånekort hos ett med ÖB anslutet bibliotek kan skapa eget användarkonto/alias, logga in och skapa/redigera/ta bort sin info via sitt biblioteks-sajt. Informationen får delas vilket uttryckligen måste godkännas av respektive användare. Respektive bibliotek/organisation ansvarar för att detta efterlevs. En Låntagare med writer-roll kan bidra med mediebeskrivningar.
Lokal administratör i ÖB (local admin [3])	Personal hos ett med ÖB anslutet bibliotek kan få behörighet att redigera/ta bort den info som det egna bibliotekets låntagare har skapat, hantera missbruksanmälan, lägga till/redigera/ta bort andra lokala administratörer och låntagaranvändare för det egna biblioteket. Det måste alltid finnas en lokal administratörer per bibliotek.
Central administratör för ÖB (admin [4])	Utsedd person med behörighet att i ÖB skapa/redigera/ta bort bibliotek, samt samma rättigheter som Lokal administratör men gällande för all information.

Utöver roller och behörigheter ställs för vissa anrop även krav på ägandeskap. Detta innebär exempelvis att en användare inte kan redigera eller ta bort andra användares betyg och omdömen, utan bara sina egna.

9 Dokumentation för vissa API-anrop

- Generellt för de beskrivna användningsfallen gäller att Öppna Bibliotek besvarar anrop med information om anropet har lyckats eller inte.
- För de anrop där det är tillämpligt ska paging parametrar (limit, offset) och listningsparametrar (order, reverse) definieras.
- Alla parametrar förväntas skickas direkt i anrops-url:ens querystring
- ÖB är i skrivande stund (2011-09-22) väldigt noga med att parametrarna skickas exakt som förväntat. Utvecklaren kan inte förvänta sig ett felmeddelande som talar om vad som saknas eller är felaktigt. Läs källkoden för antingen ÖB eller ett befintligt klient-bibliotek för att få mer insikt i formaten.
- **Verk** består alltid av utgåvor och har i ÖB följande attribut: Författare, Titel, ISBN och Libris-id, Signum.
- **Användare** (låntagare) ska i ÖB ha följande attribut: Biblioteks-benämning, "patron number" och aktuellt alias.
- **Biblioteks-benämning** är en unik benämning på det specifika biblioteket i ÖB som skapas av den Centrala administratören när biblioteket integreras med ÖB. Förslagsvis används förkortningar som inte innehåller "ääö" och mellanslag.
- **Utvecklarnyckel** är en **obligatorisk** parameter i **alla** anrop (beskrivet i kap 4). Eftersom den är gemensam för alla anrop är den utelämnad i nedanstående lista för att göra den mer lättläst.

9.1 Hantera omdömen

9.1.1 Skapa omdöme

Skapar ett omdöme om ett verk eller en utgåva
behörighet: admin, local admin, member, writer

http-metod: POST

url: <http://api.oppnabibliotek.org/assessments>

objekt: assessment

obligatoriska parametrar: se kommentar nedan

valfria parametrar:

- grade – string
- comment_header – string
- comment_text – string
- book_id – integer
- edition_id – integer
- isbn – string
- published – boolean (0 eller 1)

kommentar:

- book_id, edition_id eller isbn är obligatoriskt
- edition_id eller isbn anger book_id indirekt om omdömet gäller en utgåva

retur-xml (exempel):

```
<assessment>
  <book-id type="integer">122</book-id>
  <comment-header></comment-header>
  <comment-text></comment-text>
  <created-at type="datetime">2011-09-23T03:50:38Z</created-at>
  <edition-id type="integer">122</edition-id>
  <grade type="integer">1</grade>
  <id type="integer">1</id>
  <published type="boolean">false</published>
  <updated-at type="datetime">2011-09-23T03:50:38Z</updated-at>
```

```
<user-id type="integer">585</user-id>
</assessment>
```

Xpath för retur-xml:

föväntad retur-status: 201

retur header: location - <http://api.oppnabibliotek.org/assessments/<id>>

retur-xml (exempel vid fel):

```
<errors>
  <error>Edition has already been assessed by this user.</error>
</errors>
```

sidoeffekter:

9.1.2 Ändra omdöme

Ändrar ett omdöme

behörighet: admin, local admin, member, writer

http-metod: PUT

url: <http://api.oppnabibliotek.org/assessments/<id>>

objekt: assessment

obligatoriska parametrar:

valfria parametrar:

- grade – string
- comment_header – string
- comment_text – string
- published – boolean (0 eller 1)

kommentar:

retur-xml (exempel):

Xpath för retur-xml:

föväntad retur-status: 200

retur header:

sidoeffekter:

9.1.3 Ta bort omdöme

Tar bort ett omdöme

behörighet: admin, local admin, member, writer

http-metod: DELETE

url: <http://api.oppnabibliotek.org/assessments/<id>>

objekt: assessment

obligatoriska parametrar:

valfria parametrar:

kommentar:

retur-xml (exempel):

Xpath för retur-xml:

föväntad retur-status: 200

retur header:

sidoeffekter:

9.1.4 Visa omdömen för ett verk, utgåva eller en användare

Visar omdömen för ett verk, utgåva eller en användare

behörighet: admin, local admin, member, writer

http-metod: GET

url:

- <http://api.oppnabibliotek.org/books/<id>/assessments>
- <http://api.oppnabibliotek.org/editions/<id>/assessments>
- <http://api.oppnabibliotek.org/users/<id>/assessments>

objekt: assessment

obligatoriska parametrar:

valfria parametrar:

kommentar:

retur-xml (exempel):

```
<assessments type="array">
  <hitcount>1</hitcount>
  <average_grade>4.0</average_grade>
  <assessment>
    <id>3</id>
    <user_id>585</user_id>
    <username>testuser_la</username>
    <book_id>12341</book_id>
    <edition_id>12901</edition_id>
    <grade>4</grade>
    <comment_header>Hyfsad!</comment_header>
    <comment_text>Lite seg...</comment_text>
    <published>true</published>
    <book>
      <id>12341</id>
      <title>Jespers memoarer</title>
      <authorfirstname>Martin</authorfirstname>
      <authorlastname>Olsson</authorlastname>
      <signum />
    </book>
    <edition>
      <id>12901</id>
      <isbn>93544224</isbn>
    </edition>
  </assessment>
</assessments>
```

Xpath för retur-xml:

- hit count: /assessments/hitcount
- average grade: /assessments/average_grade
- assessment id: /assessments/assessment[index]/id
- grade: /assessments/assessment[index]/grade
- comment header: /assessments/assessment[index]/comment_header
- comment text: /assessments/assessment[index]/comment_text
- user id: /assessments/assessment[index]/user_id
- user name: /assessments/assessment[index]/username
- book id: /assessments/assessment[index]/book/id
- author firstname: /assessments/assessment[index]/book/authorfirstname
- author lastname: /assessments/assessment[index]/book/authorlastname
- signum: /assessments/assessment[index]/book/signum
- edition id: /assessments/assessment[index]/edition/id
- isbn: /assessments/assessment[index]/edition/isbn
- libris id: /assessments/assessment[index]/edition/libris_id
- media type: /assessments/assessment[index]/edition/mediatype

förväntad retur-status: 200

retur header:

sidoeffekter:

9.2 Hantera taggar och taggningar

Notera skillnaden mellan tagg och taggning:

Tagg är en etikett, som t.ex. "deckare". Den kan användas för att beskriva flera olika verk/utgåvor.

Taggning är det att en viss användare har satt en viss etikett på ett visst verk. Alltså själva kopplingen mellan användare, tagg och verk/utgåva.

9.2.1 Skapa taggning

Skapar en taggning på ett verk eller en utgåva
behörighet: admin, local admin, member, writer

http-metod: POST

url: <http://api.oppnabibliotek.org/taggings>

objekt: tagging

obligatoriska parametrar: se kommentar nedan

valfria parametrar:

- tag_name – string
- tag_id – integer
- book_id – integer
- edition_id – integer
- isbn – string
- published – boolean (0 eller 1)

kommentar:

- book_id, edition_id eller isbn är obligatoriskt
- edition_id eller isbn anger book_id indirekt om taggen gäller en utgåva
- tag_name eller tag_id är obligatoriskt

retur-xml (exempel):

Xpath för retur-xml:

förväntad retur-status: 201

retur header: location – <http://api.oppnabibliotek.org/taggings/<id>>

sidoeffekter:

- taggar som inte finns skapas upp med det namn som är bifogat i anropet

9.2.2 Ta bort taggning

Tar bort en taggning från ett verk eller en utgåva

behörighet: admin, local admin, member, writer

http-metod: DELETE

url: <http://api.oppnabibliotek.org/taggings/<id>>

objekt: tagging

obligatoriska parametrar:

valfria parametrar:

kommentar:

retur-xml (exempel):

Xpath för retur-xml:

förväntad retur-status: 200

retur header:

sidoeffekter:

9.2.3 Visa taggningar för en användare

Visar taggningar som en viss användare har gjort

behörighet: admin, local admin, member, writer

http-metod: GET

url: <http://api.oppnabibliotek.org/users/<id>/taggings>

objekt: tagging

obligatoriska parametrar:

valfria parametrar:

kommentar:

retur-xml (exempel):

```
<taggings type="array">
<hitcount>2</hitcount>
<tagging>
<id>1</id>
<user_id>585</user_id>
<username>testuser_la</username>
<book_id>12342</book_id>
<edition_id>12898</edition_id>
<published>true</published>
<tag>
<id>1</id>
<name>fantasy</name>
</tag>
<book>
<id>12342</id>
<title>Eriks bok</title>
<authorfirstname>Martin</authorfirstname>
<authorlastname>Olsson</authorlastname>
<signum />
</book>
<edition>
<id>12898</id>
<isbn>12924</isbn>
</edition>
</tagging>
</taggings>
```

Xpath för retur-xml:

- hit count: /taggings/hitcount
- tagging id: /taggings /tagging [index]/id
- name: /taggings/tagging[index]/name
- user id: /taggings /tagging [index]/user_id
- username: /taggings /tagging [index]/username
- book id: /taggings /tagging [index]/book/id
- author firstname: /taggings /tagging [index]/book/authorfirstname
- author lastname: /taggings /tagging [index]/book/authorlastname
- signum: /taggings /tagging [index]/book/signum
- edition id: /taggings /tagging [index]/edition/id
- isbn: /taggings /tagging [index]/edition/isbn
- libris id: /taggings/tagging[index]/edition/libris_id
- media type: /taggings/tagging[index]/edition/mediatype

förväntad retur-status: 200

retur header:

sidoeffekter:

9.2.4 Visa taggar för ett verk eller en utgåva

Visar taggar för ett verk eller en utgåva

behörighet: admin, local admin, member, writer

http-metod: GET

Öppna Bibliotek
oppnabibliotek.se
kontakt@oppnabibliotek.se

url:

- <http://api.oppnabibliotek.org/books/<id>/tags>
- <http://api.oppnabibliotek.org/editions/<id>/tags>

objekt: tag

obligatoriska parametrar:

valfria parametrar:

kommentar:

retur-xml (exempel):

```
<?xml version="1.0" encoding="UTF-8" ?>
<tags type="array">
  <tag>
    <id type="integer">1</id>
    <count type="NilClass">2</count>
    <name>fantasy</name>
  </tag>
  <tag>
    <id type="integer">2</id>
    <count type="NilClass">2</count>
    <name>deckare</name>
  </tag>
</tags>
```

Xpath för retur-xml:

- name: /tags/tag [index]/name
- count: /tags/tag[index]/count
- tag id: /tags/tag[index]/id

förväntad retur-status: 200

retur header:

sidoeffekter:

9.2.5 Visa alla verk och utgåvor med en viss tag

Visar alla verk och utgåvor med en viss tag

behörighet: admin, local admin, member, writer

http-metod: GET

url:

- <http://api.oppnabibliotek.org/tags/<id>>
- [http://api.oppnabibliotek.org/tags?tag\[name\]=<name>](http://api.oppnabibliotek.org/tags?tag[name]=<name>)

objekt: tag

obligatoriska parametrar:

valfria parametrar:

kommentar:

retur-xml (exempel):

```
<tags type="array">
  <tag>
    <created-at type="datetime">2009-04-07T09:21:17+02:00</created-at>
    <id type="integer">2</id>
    <name>deckare</name>
    <updated-at type="datetime">2009-04-07T09:21:17+02:00</updated-at>
    <editions type="array">
      <edition>
        <auxcreator nil="true"></auxcreator>
        <book-id type="integer">12342</book-id>
        <created-at type="datetime">2009-04-07T09:25:04+02:00</created-at>
        <id type="integer">12898</id>
        <illustrator nil="true"></illustrator>
```

```

<imageurl nil="true"></imageurl>
<isbn>12924</isbn>
<manual type="boolean" nil="true"></manual>
<mediatype nil="true"></mediatype>
<mediatypecode nil="true"></mediatypecode>
<published type="boolean" nil="true"></published>
<recordcompany nil="true"></recordcompany>
<recordnr type="integer" nil="true"></recordnr>
<ssb-key nil="true"></ssb-key>
<temp-bookid type="integer" nil="true"></temp-bookid>
<translator nil="true"></translator>
<updated-at type="datetime">2009-04-07T09:25:04+02:00</updated-at>
<year type="integer" nil="true"></year>
</edition>
</editions>
<books type="array">
  <book>
    <agegroup-id type="integer" nil="true"></agegroup-id>
    <authorfirstname>Martin</authorfirstname>
    <authorlastname>Olsson</authorlastname>
    <booktitle-part1 nil="true"></booktitle-part1>
    <booktitle-part2 nil="true"></booktitle-part2>
    <created-at type="datetime">2009-04-07T09:16:09+02:00</created-at>
    <group-inst nil="true"></group-inst>
    <id type="integer">12342</id>
    <reserved type="boolean" nil="true"></reserved>
    <signum-id type="integer" nil="true"></signum-id>
    <targetgroup-id type="integer" nil="true"></targetgroup-id>
    <temp-bookid type="integer" nil="true"></temp-bookid>
    <temp-signumtext nil="true"></temp-signumtext>
    <title>Eriks bok</title>
    <updated-at type="datetime">2009-04-07T09:16:09+02:00</updated-at>
  </book>
</books>
</tag>
</tags>

```

Xpath för retur-xml:

- title: /tags/tag/books[index]/title
- author first name: /tags/tag/books[index]/authorfirstname
- author last name: /tags/tag/books[index]/authorlastname
- signum: /tags/tag/books[index]/signum
- isbn: /tags/tag/editions[index]/isbn
- libris id: /tags/tag/editions[index]/libris_id
- media type: tags/tag/editions[index]/mediatype

förväntad retur-status: 200

retur header:

sidoeffekter:

9.2.6 Ändra tag

Ändrar en tag

behörighet: admin, local admin

http-metod: PUT

url: <http://api.oppnabibliotek.org/tags/<id>>

Öppna Bibliotek
oppnabibliotek.se
kontakt@oppnabibliotek.se

objekt: tag
obligatoriska parametrar:
· name – string
valfria parametrar:
kommentar:
retur-xml (exempel):
Xpath för retur-xml:
förväntad retur-status: 200
retur header:
sidoeffekter:

9.2.7 Ta bort tag

Tar bort en tag
behörighet: admin, local admin
http-metod: DELETE
url: <http://api.oppnabibliotek.org/tags/<id>>
objekt: tag
obligatoriska parametrar:
valfria parametrar:
kommentar:
retur-xml (exempel):
Xpath för retur-xml:
förväntad retur-status: 200
retur header:
sidoeffekter:

9.3 Missbrukshantering

9.3.1 Missbruksanmäla tag/omdöme eller beskrivning

Skapar upp en missbruksanmälan samt skickar ett mail till den anmälade användarens bibliotek och till biblioteket som taggen/omdömet/beskrivningen skapades på.

behörighet: admin, local admin, member, writer
http-metod: POST
url: <http://api.oppnabibliotek.org/reports/reportabuse>
objekt: report
obligatoriska parametrar:
· assessment_id/tag_id/description_id – integer
valfria parametrar:
· subject – string
· message - string
kommentar:
retur-xml (exempel):
Xpath för retur-xml:
förväntad retur-status: 201
retur header:
Location: <http://api.oppnabibliotek.org/reports/<id>>
sidoeffekter:

9.3.2 Notifiera missbrukaren

Skickar ett mail till den person som har skapat den anmälda taggen/ömdömet/beskrivningen

behörighet: admin, local admin
http-metod: POST

url: <http://api.oppnabibliotek.org/reports/notifyabuser>

objekt: report

obligatoriska parametrar:

· assessment_id/tag_id/description_id – integer

valfria parametrar:

· subject – string

· message - string

kommentar:

retur-xml (exempel):

Xpath för retur-xml:

förväntad retur-status: 200

retur header:

sidoeffekter:

9.3.3 Svartlista tag/omdöme/beskrivning

Svartlistar en tag/omdöme, antingen globalt eller för specifika bibliotek
behörighet: admin, local admin

http-metod: POST

url: <http://api.oppnabibliotek.org/blacklistings>

objekt: blacklisting

obligatoriska parametrar:

· assessment_id/tag_id/description_id – integer

se även kommentar

valfria parametrar:

· library_id - integer

· global – boolean (0 eller 1)

kommentar:

library_id eller global är obligatoriskt

retur-xml (exempel):

Xpath för retur-xml:

förväntad retur-status: 201

retur header:

Location: <http://api.oppnabibliotek.org/blacklisting/<id>>

sidoeffekter:

9.4 Hantera bibliotek

9.4.1 Skapa bibliotek

Skapar ett bibliotek

behörighet: admin

http-metod: POST

url: <http://api.oppnabibliotek.org/libraries>

objekt: library

obligatoriska parametrar:

valfria parametrar:

· name – string

· county_id - integer

· infolink – string

· userinfolink – string

· bookinfolink – string

· abuse_email - string

kommentar:

retur-xml (exempel):

Xpath för retur-xml:
förväntad retur-status: 201
retur header:
Location: <http://api.oppnabibliotek.org/libraries/<id>>
sidoeffekter:

9.4.2 Ändra bibliotek

Ändrar ett bibliotek
behörighet: admin, local admin
http-metod: PUT
url: <http://api.oppnabibliotek.org/libraries/<id>>
objekt: library
obligatoriska parametrar:
valfria parametrar:
· name – string
· county_id - integer
· infolink – string
· userinfolink – string
· bookinfolink – string
· abuse_email - string
kommentar:
retur-xml (exempel):
Xpath för retur-xml:
förväntad retur-status: 200
retur header:
sidoeffekter:

9.4.3 Ta bort bibliotek

Tar bort ett bibliotek
behörighet: admin
http-metod: DELETE
url: <http://api.oppnabibliotek.org/libraries/<id>>
objekt: library
obligatoriska parametrar:
valfria parametrar:
kommentar:
retur-xml (exempel):
Xpath för retur-xml:
förväntad retur-status: 200
retur header:
sidoeffekter:

9.5 Hantera användare

9.5.1 Skapa användare

Skapar en användare
behörighet: admin, local admin
http-metod: POST
url: <http://api.oppnabibliotek.org/users>
objekt: user
obligatoriska parametrar:
valfria parametrar:
· username – string

- firstname - string
- lastname – string
- password – string
- role_ids – string/array (se kommentar nedan)
- email – string
- library_id - integer
- dynamicinfolink – string

kommentar:

role_ids bör skickas med som en sträng av komma-separerade värden, ex user[role_ids]=3,4
 Detta skapar en användare med roll-id 3 och 4.

Fram till och med version 3.5.0 kunde man även skicka med en array som en parameter i url:en enligt följande syntax:

[http://api.oppnabibliotek.org/users?username=testuser&user\[role_ids\]\[\]=3&user\[role_ids\]\[\]=4](http://api.oppnabibliotek.org/users?username=testuser&user[role_ids][]=3&user[role_ids][]=4)

retur-xml (exempel):

Xpath för retur-xml:

föväntad retur-status: 201

retur header:

Location: <http://api.oppnabibliotek.org/users/<id>>

sidoeffekter:

9.5.2 Ändra användare

Ändrar en användare

behörighet: admin, local admin

http-metod: PUT

url: <http://api.oppnabibliotek.org/users/<id>>

objekt: user

obligatoriska parametrar:

valfria parametrar:

- username – string
- firstname - string
- lastname – string
- role_ids – string/array (se kommentar nedan)
- password - string
- email – string
- library_id - integer
- dynamicinfolink – string

kommentar:

role_ids bör skickas med som en sträng av komma-separerade värden, ex user[role_ids]=3,4
 Detta skapar en användare med roll-id 3 och 4.

Fram till och med version 3.5.0 kunde man även skicka med en array som en parameter i url:en enligt följande syntax:

[http://api.oppnabibliotek.org/users?username=testuser&user\[role_ids\]\[\]=3&user\[role_ids\]\[\]=4](http://api.oppnabibliotek.org/users?username=testuser&user[role_ids][]=3&user[role_ids][]=4)

retur-xml (exempel):

Xpath för retur-xml:

föväntad retur-status: 200

retur header:

sidoeffekter:

9.5.3 Ta bort användare

Tar bort en användare

behörighet: admin, local admin

http-metod: DELETE

url: <http://api.oppnabibliotek.org/users/<id>>

objekt: user

obligatoriska parametrar:

valfria parametrar:

kommentar:

retur-xml (exempel):

Xpath för retur-xml:

förväntad retur-status: 200

retur header:

sidoeffekter:

curl-exempel:

```
curl -XDELETE 'https://api.oppnabibliotek.org/users/<id>.xml?dev_key=ssb123' --insecure --user 'adminusername:adminpassword'
```

9.6 Statistik

9.6.1 Visa antal omdömen

Visar antal omdömen som har gjorts.

behörighet: admin, local admin

http-metod: GET

url: <http://api.oppnabibliotek.org/assessments/count>

objekt: assessment

obligatoriska parametrar:

valfria parametrar:

· library_id - integer

· county_id – integer

· date_from – date (yyyymmdd)

· date_to – date (yyyymmdd)

kommentar: local admins kan få statistik för deras eget bibliotek genom att ange 'own' som library_id

retur-xml (exempel):

```
<assessments>
  <count>
    <total>3</total>
    <published>3</published>
    <blacklisted>0</blacklisted>
  </count>
</assessments>
```

Xpath för retur-xml:

· total: assessments/count/total

· published: assessments/count/published

· blacklisted: assessments/count/blacklisted

förväntad retur-status: 200

retur header:

sidoeffekter:

9.6.2 Visa antal taggningar

Visar antal taggningar som gjorts

behörighet: admin, local admin

http-metod: GET

url: <http://api.oppnabibliotek.org/taggings/count>

objekt: tagging

obligatoriska parametrar:

valfria parametrar:

· library_id - integer

- county_id – integer
- date_from – date (yyyymmdd)
- date_to – date (yyyymmdd)

kommentar: local admins kan få statistik för deras eget bibliotek genom att ange 'own' som library_id

retur-xml (exempel):

```
taggings>
  <count>
    <total>4</total>
  </count>
</taggings>
```

Xpath för retur-xml:

- total: taggings/count/total

förväntad retur-status: 200

retur header:

sidoeffekter:

9.6.3 Visa antal taggar

Visar antal taggar som finns
behörighet: admin, local admin

http-metod: GET

url: <http://api.oppnabibliotek.org/tags/count>

objekt: tags

obligatoriska parametrar:

valfria parametrar:

kommentar: OBS! denna metod är inte korrekt implementerad i rails-lösningen

retur-xml (exempel):

```
<tags>
  <count>
    <total>3</total>
    <blacklisted>0</blacklisted>
  </count>
</tags>
```

Xpath för retur-xml:

- total: tags/count/total
- blacklisted: tags/count/blacklisted

förväntad retur-status: 200

retur header:

sidoeffekter:

9.6.4 Visa antal descriptions

Visar antal descriptions som gjorts
behörighet: admin, local admin

http-metod: GET

url: <http://api.oppnabibliotek.org/descriptions/count>

objekt: description

obligatoriska parametrar:

valfria parametrar:

- library_id - integer
- county_id – integer
- date_from – date (yyyymmdd)
- date_to – date (yyyymmdd)

kommentar: local admins kan få statistik för deras eget bibliotek genom att ange 'own' som library_id

retur-xml (exempel):

```
taggings>
  <count>
    <total>4</total>
    <blacklisted>0</blacklisted>
  </count>
</taggings>
```

Xpath för retur-xml:

· total: taggings/count/total

förväntad retur-status: 200

retur header:

sidoeffekter:

9.6.5 Visa antal missbruksanmälningar

Visar antal missbruksanmälningar som gjorts

behörighet: admin, local admin

http-metod: GET

url: <http://api.oppnabibliotek.org/reports/count>

objekt: report

obligatoriska parametrar:

valfria parametrar:

· date_from – date (yyyymmdd)

· date_to – date (yyyymmdd)

kommentar: det saknas filter av inkommande och utgående reports

retur-xml (exempel):

```
taggings>
  <count>
    <total>4</total>
  </count>
</taggings>
```

Xpath för retur-xml:

· total: taggings/count/total

förväntad retur-status: 200

retur header:

sidoeffekter:

9.7 Hantera databas

9.7.1 Hämta filtrerad dump av databasen

Hämtar en filtrerad dump av databasen. Filen produceras varje natt.

behörighet: admin, local admin

http-metod: GET

url: <http://api.oppnabibliotek.org/users/dbdump>

10 Generellt om söka & sortera

10.1 Söka

Man ska kunna söka efter alla användare, verk, betyg, omdömen, etiketter, alla verk för ett signum.

Man ska kunna filtrera på bibliotek, betyg, frekvens (t ex etiketter med bara en förekomst), publicerat/opublicerat, tidsintervall.

Man ska kunna söka ut användare som är mest aktiva. Till exempel:

- Lista de 10 användare som har satt mest betyg, etiketter eller omdömen/kommentarer.
- Lista de 10 användare som har satt högst betyg, skapade den senaste veckan.

10.2 Sortera

Informationen i ett sökresultat ska kunna sorteras enligt:

- Verk sorteras efter bästa/sämsta betyg, flest betyg, senast inlagda betyg, flest mervärden, författare (alfabetiskt), titel (alfabetiskt).
- Omdöme/Kommentar för ett verk ska kunna sorteras utifrån alias, utgåva, publiceringsdatum.
- Etiketter ska kunna sorteras i alfabetisk ordning, efter frekvens, och publiceringsdatum.

11 Praktisk information för utvecklare

11.1 Utvecklarlicens

Alla bibliotek, projekt och produkter som vill hämta data från ÖB skall i varje anrop till tjänsten identifiera sig med en unik nyckel som skapas av admin efter t ex mailkontakt med kontaktperson på ÖB.

11.2 Administratörer

Nycklar tilldelas på biblioteksnivå - en per bibliotek - av en ÖB-administratör. Kontakta kontakt@opnabibliotek.se för mer info.

11.3 Buggar och tekniska problem

Det höga versionnumret till trots (3.5.0) så är API:erna ännu ej färdigtestade mer än i den utsträckning som de används i dagsläget. Se punkt 12.4

Kontakta kontakt@opnabibliotek.se angående frågor om felkorrigering och generell vidareutveckling.

11.4 Bakåtkompatibilitet

Innan version 3.5.0 så skedde alla anrop mot www.opnabibliotek.org. Från och med version 3.5.0 så sker alla anrop via api.opnabibliotek.org

För att säkra bakåtkompatibilitet så används följande domänstruktur:

Alla versioner (X.Y.Z) kan nås via vX_Y_Z.api.opnabibliotek.org, exempelvis v3_5_0.api.opnabibliotek.org

Den senaste i serien X.Y kommer alltid att nås via vX_Y.api.opnabibliotek.org, exempelvis v3_5.api.opnabibliotek.org

Den senaste i serien X kommer alltid att nås via vX.api.opnabibliotek.org, exempelvis v3.api.opnabibliotek.org

Den senaste stabila versionen kommer alltid att nås via api.opnabibliotek.org

Nästa api-uppdatering (3.5.1) kommer alltså kunna nås på v3_5_1.api.opnabibliotek.org, v3_5.api.opnabibliotek.org, v3.api.opnabibliotek.org samt api.opnabibliotek.org fram tills dess att ytterligare en uppdatering släpps.

12 Tredjepartsverktyg och klientbibliotek

12.1 Klient-bibliotek skrivet i PHP

En nästintill komplett implementation av API:t finns som klasser skrivna i PHP. Kontakta Virtuella Enheten på Stockholms Stadsbibliotek för mer information.

12.2 Drupal-modul för betygsättning och recensering av verk

Version 1.0 är färdigutvecklad och används på <http://biblioteket.stockholm.se>

12.3 Drupal-modul för administration användarproducerat material

Delvis utvecklad och ska ev komma att sättas i drift i redaktionella delen av <http://biblioteket.stockholm.se>

12.4 Exempel på implementationer

<http://biblioteket.stockholm.se> - Se exempelvis sökresultatlistningarna, fullposten, sidan "Mina betygsatta böcker" mm.

<http://boktips.net> - Hela sidan bygger på information som hämtas och skrivs till Öppna Biblioteks API:er