

WebAPI

Bibliotek-Systemer As

WebAPI

Bibliotek-Systemer As

Innholdsfortegnelse

Register	1
1. Bibliofils WebAPI	2
1.1. Autentisering	2
1.2. Uthenting av data	4
1.2.1. OAI-PMH	4
1.2.2. Z39.50	5
1.2.3. Search/Retrieve via URL (SRU)	5
1.2.4. RSS	7
1.2.5. KatalogKrydder	7
1.2.6. REST-servicer	9
1.3. Endring av data	14
1.3.1. REST-servicer	14
1.4. Test APIet	19

tabelloversikt

1.1. metadataPrefix	4
1.2. SRU recordSchema	5

Eksempeloversikt

1.1. OAuth	3
1.2. SRU	6
1.3. KatalogKrydder	7
1.4. KatalogKrydder som json	8
1.5. <i>items</i> og visningsmaler	9
1.6. GET Request for items	9
1.7. GET Request for copies	10
1.8. getDepartments	11
1.9. translateBarcode	11
1.10. getBranches	12
1.11. userinfo	12
1.12. requests	13
1.13. getLnr	13
1.14. bibauth	14
1.15. addReservation	15
1.16. removeReservations	15
1.17. getLoans	16
1.18. renewLoans	18
1.19. Obskoder	18

Register

A

Autentisering, 2

D

Dublin Core, 4, 5

F

Format, 9

H

HTTPS, 2

J

JSON, 9

K

Katalogkrydder, 7

M

MarcXchange, 4

O

OAI-PMH, 4

Oauth, 2

R

REST-service

 endring av data, 14

 uthenting av data, 9

RSS, 7

S

Sikkerhet, 2

SRU, 5

Z

Z39.50, 5

Kapittel 1. Bibliofils WebAPI

Bibliofils WebAPI (programmeringsgrensesnitt) er todelt. Hoveddelen inneholder hovedsaklig alle funksjoner som kan brukes for å hente ut informasjon fra databasen. Dette inkluderer blant annet søking og bestandinformasjon. Den andre delen gir tilgang til funksjoner som gjør endringer i databasen. Endringer i databasen stiller større krav til autentisering.

WebAPI-et har en forholdsvis liten kostnad som skal dekke installasjon og (videre)utvikling av grensesnittene. Man kan velge å bare kjøpe hoveddelen hvis det er kun den man ønsker å bruke. Send gjerne en forespørsel til bestill@bibsyst.no for et uforpliktende tilbud.

WebAPI-et er ment å brukes av utviklere, men vi har også funksjoner som enklere kan integreres i nettsidene til biblioteket. Vi har portalvennlige versjoner av Websøket og MappaMi samt mulighet for integrasjon av en innloggingsboks for lånerne i nettsidene til biblioteket. Dette er ikke inkludert i WebAPI-et, men er dokumentert på *Bibliofils hjemmesider* [<http://bibliofil.no/kundeinfo/labben>].

WebAPI-et består av velkjent webteknologi. Enkelte funksjoner baserer seg på internasjonale standarder og bruker primært XML for utveksling av data, men vi har også Bibliofil-spesifikke funksjoner som bruker rene REST-grensesnitt med svar i JSON. Disse er spesielt egnet for bruk i Javascript direkte i nettsidene.

Vi forventer at biblioteket har en databehandlingsavtale for de delene som krever det med eventuelle eksterne aktører slik at disse har fått tilgang til å hente ut og behandle lånerdata. Bibliotek-Systemer As kan ikke ta ansvar for eventuell misbruk som skjer igjennom tilgangen som er gitt til tredjepartsaktører som er godkjent av biblioteket.

WebAPI-et er ikke skrevet i stein. Vi vil videreutvikle det i henhold til nye ønsker og krav som kommer fra kunder og deres samarbeidspartnere. Noe vil vi kunne gjøre på veldig kort tid (og som en del av standardløsningen), mens større endringer vil kreve mer utviklingstid. Kom gjerne med ønsker for vurdering.

Viktig

Ekseplene i dokumentasjonen er ment å vise hvordan URL-ene bygges opp og hva man kan forvente å få i svar. For REST-servicene som krever innlogging vil de ikke kunne testes uten gyldig innlogging. For å teste må man gå mot Bibliofil-tjeneren til biblioteket man integrerer mot, hvilket betyr at man bytter ut "faretbib" med maskinnavnet til bibliotekets tjener.

Kontaktinformasjon. For et uforpliktende tilbud: bestill@bibsyst.no. For teknisk informasjon/spørsmål/feilmelding/ønsker: webapi@bibsyst.no

1.1. Autentisering

REST-servicene har to sikkerhetsnivåer. Begge krever bruk av HTTPS. Uthenting av ikke-konfidensiell informasjon krever autentisering vha *Basic Authentication*, mens lånerinformasjon og endringer i databasen krever at man har autentisert seg vha *Oauth 2.0* (<http://oauth.net/>).

For Basic Authentication vil vi opprette en global bruker for hver implementatør (per bibliotek) som brukes når man kaller de ikke-konfidensielle REST-servicene. NB! Disse kan ikke kalles direkte ifra nettleseren da dette vil eksponere innloggingsinformasjonen som skal behandles konfidensielt av implementatør.

Oauth 2.0 autentiseringen fungerer slik at låneren blir videresendt fra eksternt side/applikasjon til Bibliofils innlogging (med informasjon om retur URL) hvor låneren logger inn med sitt lånernummer og pin/passord. Når innloggingen er gjort blir låneren videresendt tilbake til nettsida/applikasjonen med en innloggingstoken i URLen. Dette tokenet brukes så ved kall til REST-servicene. Tokenet kan sendes med i cookie eller i en POST. Vi tillater ikke at tokenet sendes med i URLen (v/GET) annet enn for testing da dette vil eksponere tokenet bl.a. i loggfilene til webtjeneren.

Innloggingssida til Bibliofil vil ta seg av å gi låneren nytt passord/pin hvis glempt og vil også støtte 2-faktor autentisering på sikt. Tredjepartsaktører vil ikke få tilgang til lånerens passord/pin ved bruk av denne løsningen.

Det er viktig at REST-servicene tar høyde for at feilsituasjoner kan oppstå hvis f.eks. tokenet har blitt gjort ugyldig. REST-servicene vil i såfall returnere **status=false**.

Vi vil legge inn en id og passord for integratør som brukes for å identifisere hvem som gjør kallene til REST-servicene. I tillegg vil vi registrere en URL for å begrense hvilke URLer låneren kan videresendes til fra OAuth-innloggingen.

Eksempel 1.1. OAuth

https://faretbib.bib.no/cgi-bin/oauthlogin kalles med følgende argumenter:

```
response_type=token
client_id=<clientid>
redirect_uri=<URI>
```

client_id for DesignFirmaX for FaretBib er:

```
5c57d72d57565a545925559e55ff6d68fff412f78a8991aceb51eb5ccca0feacba71cba5a3
```

redirect_uri skal være en URI-formatert URI tilbake til app. NB! Denne må være registrert i OAuth-oppsettet på Bibliofil-tjeneren. Den trenger ikke være komplett, men nok til at den ikke kan misbrukes. Under testingen er den satt til "https://*".

Når låneren har logget inn vil *redirect_uri* bli kalt med *#token* som må brukes for å aksessere webservicene første gang.

URLen som Appen sender låneren vil se slik ut:

```
https://faretbib.bib.no/cgi-bin/oauthlogin?response_type=token&client_id=5c57d72d57
```

Etter innlogging vil låneren redirigeres til (etter vellykka innlogging):

```
https://designfirmamax.no/MyExternalApp#02d38fb41a88d90a159e74343243243947346234732ae
```

Her har man 3 valg:

- a. Send med tokenet i HTTP-headeren:

```
Authorization: Bearer 02d38fb41a88d90a159e74343243243947346234732ae0d812d922eccf0
```

- b. Sett en cookie med tokenet:

```
oa_lnr_token=02d38fb41a88d90a159e74343243243947346234732ae0d812d922eccf0b10006fa0
```

- c. Send med i URL. NB! Dette skal kun brukes til testing da tokenet blir lagret bl.a. i Apache-loggen.

`https://.....?oa_lnr_token=02d38fb41a88d90a159e74343243243947346234732ae0d812d922`

Når dette kallet er gjort første gang etter innlogging vil vi opprette cookiene:

```
oa_lnr_expires
oa_lnr_token
oa_lnr_user
```

Det er dermed ikke nødvendig å sende med/sette token etter første kall.

Tokenet er i utgangspunktet gyldig i 6 timer og refreshes automatisk hver gang man gjør et kall som krever OAuth-autentisering. Timeout kan justeres etter behov.

Webservicene vil returnere **status=false, message="Unauthorized login"** hvis token er ugyldig.

Utlogging kan gjøres med *redirect_uri* og *logout=1*:

`https://faretbib.bib.no/cgi-bin/oauthlogin?response_type=token&client_id=<client_id>`

1.2. Uthenting av data

Dette er den viktigste delen av WebAPI-et og den som de fleste vil benytte seg av. Flere av funksjonene er allerede tilgjengelig på mange bibliotek, så sjekk gjerne om dere allerede har det dere trenger.

1.2.1. OAI-PMH

OAI-PMH (<https://www.openarchives.org/OAI/openarchivesprotocol.html>) er en standard internasjonal innhøstingsprotokoll som primært brukes av Nasjonalbiblioteket for innhøsting av katalogpostene til bibliotekene. Dette er protokollen som ligger i bunnen av *Biblioteksøk* og målet til Nasjonalbiblioteket er at alle folkebibliotek skal ha dette etterhvert. Bibliofil OAI-PMH-tjenere støtter *deleted records* og *OAI-PMH sets*.

Følgende *metadataPrefix* er tilgjengelig:

Tabell 1.1. metadataPrefix

Prefix	Forklaring
dc	Dublin Core
marcexchange	Normarc i MarcXchange
bibliofilmarc	Full Bibliofilpost med \$3 i MarcXchange
bibliofilmarcnoholdings	Full Bibliofilpost med \$3 i MarcXchange, men uten beholdningsdata
noholdings	Samme som MarcXchange, men uten beholdningsdata

Oversikt over tilgjengelige OAI-PMH-tjenere finnes her: <http://www.bibsys.no/oaiservere.html>. Her er også informasjon om OAI-PMH-repositoryet er allment tilgjengelig eller om det er begrenset til spesifikke IP-adresser.

Hvis biblioteket har OAI-PMH vil det være tilgjengelig på <https://maskinnavn.bib.no/cgi-bin/oai>. Se **?verb=Identify** for mer informasjon om repositoryet. "*maskinnavn*" er hostnavnet på Bibliofil-tjeneren.

Da OAI-PMH er en ren innhøstingsprotokoll er den lite brukt utenfor Biblioteksøk. Vi henviser derfor til *Open Archives Initiative* [<https://www.openarchives.org/pmh/>] for mer informasjon om denne protokollen.

1.2.2. Z39.50

Z39.50 er en binær søkeprotokoll som brukes mye på det norske bibliotekmarkedet, men vi anbefaler at man ikke lager nye løsninger basert på den så fremt man ikke må. Dette fordi videreutviklingen av Z39.50 på det norske markedet har stoppet opp og biblioteksystemleverandørene driver kun med vedlikehold og feilretting for denne protokollen.

1.2.3. Search/Retrieve via URL (SRU)

Søking har vi gjort tilgjengelig igjennom en XML-basert internasjonal søkeprotokoll. Bibliofil SRU-tjener er i henhold til SRU-standarden v1.2 (<http://www.loc.gov/standards/sru/>) samt norsk profil (som er definert på <http://norzig.no>).

SRU-tjeneren støtter følgende *recordSchema*:

Tabell 1.2. SRU recordSchema

recordSchema	Forklaring
dc	Dublin Core
zeerex	Explain Schema
normarc	Normarc i MarcXchange
bibliofilmarc	Full Bibliofilpost med \$3 i MarcXchange
bibliofilmarcnoholdings	Full Bibliofilpost med \$3 i MarcXchange, men uten beholdningsdata
noholdings	Samme som MarcXchange, men uten beholdningsdata
bsdc	Dublin Core, med ekstra Bibliofil-spesifikke tagger

Vi anbefaler at man bruker **dc** eller **bsdc** hvis man ikke er ute etter komplette katalogposter eller beholdningsdata i 850-taggene.

Oversikt over tilgjengelige SRU-tjenere ligger her: <http://www.bibsys.no/produkter/bibliofil/z3950.php>. Alle som har kjøpt Z39.50 har også SRU da vi ser på det som en videreutvikling/erstatning for Z39.50.

Tilgjengelige søkeindekser kan fås ved å kalle <https://maskinnavn.bib.no/cgi-bin/sru>. De som ligger i context settet *bibliofil* er Bibliofil-spesifikke, mens de andre er enten fra SRU-standarden eller Norzig-profilen.

Bibliofil SRU-tjener har en utvidelse for å få støtte for fasetter. Ved å sende med **extraRequestData=bibliofil:fasetter:1** vil fasetter legges med i svaret i *extraResponseData*. Tallet 1 indikerer fasettnivået. 1 betyr at man skal maks returnere 1 treff per type fasett. Man står fritt til å øke den etter behov, men desto høyere, desto større (og tregere) svar må forventes. Det er viktig å være klar over at det ligger en kostnad i tidsbruk ved å be om fasetter, da dette betyr at SRU-tjeneren må løpe igjennom alle treffene og hente ut fasettinformasjonen. Hvis man har store trefflister er man ikke garantert at tjeneren klarer å svare innenfor fornuftig tid.

Det er mulig å be om andre typer sorteringer enn standard, men de krever en gjennomløping av treffene og vil derfor medføre til at søkene går tregere. Vi støtter sortering på creator, title, date, spatial. Eks: *sortKeys=date,,false,false,highValue* (sorterer på utgiverår, ingen schema, *!ascending*, *!caseSensitive*, *missingValue=highValue*). Dette er defaultverdiene, så man får samme resultat med *sortKeys=date*.

SRU-tjeneren støtter også *scan* i tillegg til *searchRetrieve*. F.eks. **operation=scan&scanClause=dc.title** vil returnere indekserte tittel-termer fra databasen (fra emneregisteret).

SRU-tjeneren støtter kun basisprofilen for CQL med noen få unntak. Trunkering (høyre) kan brukes, men det kreves minst 2 tegn for at det skal være gyldig. Forespørselen vil bli automatisk avbrutt hvis søket tar mer enn 60 sekunder.

Eksempel 1.2. SRU

GET Request:

```
https://kongbib.bib.no/cgi-bin/sru?operation=searchRetrieve&recordSchema=dc&query=n
```

Resposn:

```
<?xml version="1.0" encoding="utf-8"?>
<SRU:searchRetrieveResponse xmlns:SRU="http://www.loc.gov/zing/sru/"
xmlns:dc="info:sru/schema/1/dc-v1.1"
xmlns:normarc="info:lc/xmlns/marcxchange-v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:DIAG="http://www.loc.gov/zing/sru/diagnostics/">
  <SRU:version>1.2</SRU:version>
  <SRU:numberOfRecords>80</SRU:numberOfRecords>
  <SRU:resultSetId>833</SRU:resultSetId>
  <SRU:nextRecordPosition>11</SRU:nextRecordPosition>
  <SRU:records>
    ... treffliste ...

  </SRU:records>
</SRU:searchRetrieveResponse>
```

Svaret inneholder antall treff, en *resultSetId* som kan brukes for å få de neste treffene og hvilken posisjon det neste treffet (som ikke er inkludert i svaret) vil ha. Selve treffene er ikke tatt med i dette eksempelet.

Treffene vil se slik ut ved bruk av *dc*:

```
<SRU:record>
  <SRU:recordSchema>
    info:srw/schema/1/dc-v1.1</SRU:recordSchema>
  <SRU:recordPacking>xml</SRU:recordPacking>
  <SRU:recordData>
    <dc>
      <dc:title>Arme riddere</dc:title>
      <dc:description type="abstract">Vettskremt og blodig
våkner Oscar Svendsen med en hagle i hendene i det som en
gang var en anstendig strippebule ved Svinesund. Rundt
ham ligger åtte lik, og foran ham står politietterforsker
Solør og peker på ham med en pistol. For Solør fremstår
det klart at han er den skyldige selv om Oscar hardnakket
bedyrer sin uskyld. Nølende begynner Oscar å fortelle den
utrolige historien om fire karer som sammen fikk tolv
rette i tipping og nesten to millioner kroner i fanget.
Pengene skulle vise seg å være vanskelig å dele på
fire.</dc:description>
      <dc:publisher>Nordisk film</dc:publisher>
      <dc:date>2012</dc:date>
      <dc:format id="material">video</dc:format>
      <dc:identifiser id="localid">94529</dc:identifiser>
      <dc:identifiser id="bibliofilid">8235056</dc:identifiser>
      <dc:language>nob</dc:language>
    </dc>
  </SRU:recordData>
</SRU:record>
```

```
</SRU:recordData>  
<SRU:recordPosition>1</SRU:recordPosition>  
</SRU:record>
```

maximumRecords (antall treff man får returnert i hvert svar) er standard 10, men kan økes opp til maksimalt 500 ved å sende den med i URL.

CCL-søk i SRU:

<https://kongbib.bib.no/cgi-bin/sru?operation=searchRetrieve&query=bibliofil.ccl=%22>

1.2.4. RSS

Alle Bibliofil-bibliotek har tilgjengelig RSS-strømmer for ting som nye titler, mest utlånte, etc. En enkel RSS kan lages ved å sende med *ccl*=argumentet til */cgi-bin/rss*. Det er også mulig å sende med *bibxml=1* for å få med enkelte katalogspesifikke tagger samt noe KatalogKrydder hvis det er tilgjengelig.

RSS er beskrevet i Statistikkhåndboken: <http://www.bibsys.no/kundeinfo/dokumentasjon/web/tkstat/rss.html>.

1.2.5. KatalogKrydder

Tilgang til KatalogKrydder er begrenset med IP-adresse og kan i utgangspunktet ikke benyttes direkte fra nettleseren. Tilgang kan kjøpes separat fra WebAPI-et hvis det er ønskelig.

API-et mot KatalogKrydder er XML-basert. Kall mot *krydderxml* består av en GET som inneholder Bibliofil-ID (en Bibliofil-spesifikk identifikator som brukes for å unikt identifisere verk) for å finne krydder for et spesifikt verk. Det finnes krydder for de fleste dokumenttyper, ikke bare for bøker. Krydder legges inn av bibliotekene selv i en dagnadsinnsats. Typer krydder som finnes er forsidebilder, beskrivelser, innholdsfortegnelser, anmeldelser (av lånere), stjernekest, etc.

Svarformat er enten *xml* (default) eller *json*: **format=json**

Eksempel 1.3. KatalogKrydder

Request:

<https://krydder.bib.no/cgi-bin/krydderxml?bibid=64286>

Resposn:

```
<?xml version='1.0' encoding='utf-8'?>  
<krydder xmlns="http://krydder.bib.no/cgi-bin/krydderxml">  
  <item>  
    <bibliofilid>64286</bibliofilid>  
    <kryddertyper>bilde beskrivelse</kryddertyper>  
    <krydder_bilde>  
      <type>forside</type>  
      <bilddata>  
        <størrelse>256x</størrelse>  
        <url>https://krydder.bib.no/0395/64286.bilde.1327560326.n.jpg</url>  
      </bilddata>
```

```

<bilddata>
  <størrelse>ekstrastor</størrelse>
  <url>https://krydder.bib.no/0395/64286.bilde.1327560326.x.jpg</url>
</bilddata>
<bilddata>
  <størrelse>medium</størrelse>
  <url>https://krydder.bib.no/0395/64286.bilde.1327560326.m.jpg</url>
</bilddata>
<bilddata>
  <størrelse>karusell</størrelse>
  <url>https://krydder.bib.no/0395/64286.bilde.1327560326.k.jpg</url>
</bilddata>
<bilddata>
  <størrelse>liten</størrelse>
  <url>https://krydder.bib.no/0395/64286.bilde.1327560326.l.jpg</url>
</bilddata>
<bilddata>
  <størrelse>stor</størrelse>
  <url>https://krydder.bib.no/0395/64286.bilde.1327560326.s.jpg</url>
</bilddata>
</krydder_bilde>
<krydder_beskrivelse>
  <type>forlagstekst</type>
  <url>http://krydder.bibsys.no/0395/64286.beskrivelse.1327560327.xml</url>
  <tekst>Forlaget skriver om denne boka:<br> "Paul Auster
har skrevet en glassklar og vakker bok, en bok med nesten
nifs gjennomsiktighet og åpenhet" - slik presenterer Jan
Kjærstad New York-trilogien i Bokspeilet, Bokklubben nye
bøkers medlemsblad. Det er tre bøker sm her kjem i eitt band,
<em>Glasbyen, Gjengangarar og Det stengde
rommet</em>. I den danske avisa Eksta Bladet skriv Poul
Borum om trilogien: "Det er kriminalromaner, der samtidig
leger med fiktionen og har noget væsentligt at sige. Auster
bruger virtuost alle klicheerne fra Raymond Chandler osv. og
blander dem med Borges og Poe (og Lewis Carroll og Cervantes
og meget mere), og mixer alle disse elementer til en
forbløffende og forrygende labyrintisk fortelling med vittig
leg med identiteter."<br> Den amerikanske forfatteren
Paul Auster er fødd i 1947, universitetsutdanna litterat, har
også vore sjømann og budd i Frankrike ei tid. No bur han
Brooklyn, New York. Han understrekar gjerne at han er
jøde.</tekst>
  </krydder_beskrivelse>
</item>
</krydder>

```

Eksempel 1.4. KatalogKrydder som json

Request:

<https://krydder.bib.no/cgi-bin/krydderxml?bibid=64286&format=json>

Respons:

```
{
```

```

"status"      : "OK",
"statusmelding" : "",
"64286"      : {
  "kryddertyper"      : ["bilde", "beskrivelse"],
  "krydder_bilde"    : [{
    "epoc"      : "1327560326",
    "type"      : "forside",
    "størrelser" : ["k", "l", "m", "n", "s", "x"],
    "url"       : "https://krydder.bib.no/0395/64286.bilde.1327560326.m.jp
  }],
  "krydder_beskrivelse" : [{
    "epoc" : "1327560327",
    "type" : "forlagstekst",
    "url"  : "http://krydder.bibsys.no/0395/64286.beskrivelse.1327560327.
    "tekst" : "Forlaget skriver om denne boka:<br>\n\"Paul Auster har skrev
  }]
}
}

```

For en oversikt over hva som er tilgjengelig og en enkel hjelpetekst kan man kalle KrydderXML direkte uten argumenter: <https://krydder.bib.no/cgi-bin/krydderxml>.

1.2.6. REST-serviceser

WebAPI-et har noen funksjoner tilgjengelig igjennom REST-kall som returnerer JSON-kode som enkelt kan behandles i f.eks. Javascript. Noen av disse er kun tilgjengelig over HTTPS og krever autentisering.

1.2.6.1. items&copies

Full beholdningsdata med enkel tittelinformasjon og eventuelt KatalogKrydder hvis dette er tilgjengelig. Krever ikke autentisering (ingen lånerdata).

items brukes for å hente ut katalogdata og lenker til eksemplarene. *copies* gir eksemplarinformasjon for tittelen.

Identifikatoren er det lokale tittelnummeret i basen (i eksempelet 325234). Identifikatoren kan bestå av flere tittelnumre separert med komma.

Innholdet i svaret fra *items* bestemmes av maler i Bibliofils websøk og kan tilpasses der.

Eksempel 1.5. *items* og visningsmaler

https://tbgbib.bib.no/cgi-bin/rest_service/items/1.0/data/325234,54231?postformat=k

Eksemplet ber om formatmalen *kort*. Tilgjengelige maler kan variere fra bibliotek til bibliotek. I utgangspunktet har alle følgende formater: *default*, *kortformat*, *forhand*, *isbd*, *wap*, *tidsskrift*, *bilde*, *bilde-ekstra*, *bilde-deich*, *dikt*, *kort*, *DC*, *nettressurs*, *originalspreek*, *lydbib-komplett*, *lydbib*. Noen av disse er spesike for enkeltbibliotek, og kan være endret eller slettet av andre.

Nye visningsformat kan defineres i innstillingene for websøket (Fellesinnstillinger → Websøk4 → websøk4.tcl → Visning → Rediger visningsformater (webpostformat.par)).

Eksempel 1.6. GET Request for *items*

https://tbgbib.bib.no/cgi-bin/rest_service/items/1.0/data/325234

Respons:

```
{
  "325234":{
    "Tittel": ["Dying to tell"],
    "Forfatter": ["Goddard, Robert , Engelsk 1954-"],
    "Språk": ["engelsk"],
    "Utgitt": ["London : Bantam , 2001"],
    "Omfang": ["345 s."],
    "Opplysninger": ["Engelsk tekst", "%brefr2-TbgBib-1347369284952", "LOCAL-ID", "0000"],
    "Sjanger": ["Krim"],
    "ISBN": ["0-593-04762-1", "(h.)", "155.00"],
    "Hylleplass": ["82 G"],
    "bibliofilid":2142206,
    "copies":"https://tbgbib.bib.no/cgi-bin/rest_service/copies/1.0/data/325234?fields=full&exfields=full",
    "duration":232,"time":"20160622-1340-29 +0200"
  }
}
```

Eksempel 1.7. GET Request for copies

https://tbgbib.bib.no/cgi-bin/rest_service/copies/1.0/data/325234?fields=full&exfields=full

Respons:

```
{
  "elements":
  { "325234":{
    "bibliofilid":2142206,
    "total":0,
    "available":0,
    "holds":0,
    "copies":[]
  }
  },
  "length":1,"updatetime":"2016-06-22 1348-14:+0200","duration":90,"apiver":"1.0-beta"
}
```

Hvis man ikke tar med *fields* og *exfields* vil man få en kort variant med kun totaltall.

Det er mulig å ta ut både *items* og *copies* i en request slik: https://larbib.bib.no/cgi-bin/rest_service/items/1.0/data/25231?fields=full&holdings=1

Mer informasjon kan få ved å ta opp REST-tjenesten uten argumenter: https://larbib.bib.no/cgi-bin/rest_service/copies.

Viktig

For å få ut JSONP, må det sendes med en vilkårlig streng i argumentet *jsoncallback*. Den verdien blir da lagt inn som et funksjons-kall rundt json-koden. Dette er nødvendig dersom dataene skal bli hentet ut via javascript fra et annet domene.

Autentisering: Ingen. Anonyme data.

getDepartments

Henter ut liste over avdelinger på biblioteket. Default returneres henteavdelingene hvis biblioteket har satt dette opp. Bruk *type=alle* for å få alle avdelinger.

Eksempel 1.8. getDepartments

Request:

`https://faretbib.bib.no/rest_service/webapi_getDepartments/1.0/data`

Respons:

```
{
  "status"      : true,
  "serverdate"  : "2016-04-28T13:24:03",
  "departments" : [{
    "code" : "fhb",
    "name" : "Faret Hovedbiblioteket"
  }, {
    "code" : "fhbv",
    "name" : "Faret Hovedbiblioteket voksen"
  }]
}
```

Autentisering: Krever gyldig innlogging vha Basic Authentication.

translateBarcode

Oversettelse av strekkode/RFID til tittelnummer og eksemplarnummer.

Parameter: *barcodeId* // Strekkode/RFID

Return: Et objekt som inneholder *unique_material_id*, *local_id* and *bibliofil_id* for *barcodeId*.

Eksempel 1.9. translateBarcode

`https://faretbib.bib.no/rest_service/webapi_translateBarcode/1.0/data?barcodeId=`

```
{
  "status"      : true,
  "message"     : "",
  "serverdate"  : "2016-04-15T11:17:50",
  "material"    : {
    "bibliofilid" : 131371,
    "localId"     : 19,
    "loanId"      : "ix:19,7"
  }
}
```

Autentisering: Krever gyldig innlogging vha Basic Authentication.

getBranches

Henter ut bibliotekinformasjon.

Funksjon: `getBranches`

Parametere: Ingen

Returnerer: Array med bibliotek. Som regel er dette bare ett, men det kan teoretisk være flere.

```
branch object:
{
  code: string, // Unik forkortelse for biblioteket, oftest er dette "main".
  name: string, // Bibliotekets fulle navn
  longitude: float|null,
  latitude: float|null
}
```

Eksempel 1.10. getBranches

https://faretbib.bib.no/rest_service/webapi_getBranches/1.0/data

```
{
  "status"      : true,
  "message"     : "",
  "serverdate" : "2016-04-15T11:12:13",
  "branch"      : [{
    "code"       : "main",
    "name"       : "Kongsberg bibliotek",
    "latitude"   : "59.664926",
    "longitude"  : "9.645268"
  }]
}
```

Autentisering: Krever gyldig innlogging vha Basic Authentication.

userinfo

Henter ut informasjon om autentisert låner.

Funksjon: userinfo

Eksempel 1.11. userinfo

Eksempel: https://faretbib.bib.no/cgi-bin/rest_service/oawebapi_userinfo/1.0/data

```
{
  "status"      : true,
  "message"     : "",
  "serverdate" : "2016-08-09T15:48:08",
  "userinfo"    : [{
    "navn"       : "Nordmann, Ola",
    "adresse1"   : "Kongeveien 99",
    "adresse2"   : "",
    "postnr"    : "0202",
    "poststed"  : "OSLO"
  }]
}
```

Autentisering: Krever OAuth 2.0 autentisering.

requests

Henter ut reserveringer.

Eksempel 1.12. requests

https://faretbib.bib.no/cgi-bin/rest_service/oawebapi_requests/1.0/data

Respons:

```
{
  "status"      : true,
  "message"     : "",
  "serverdate" : "2016-08-09T15:38:13",
  "requests"   : [{
    "idx"       : "res:4188,102",
    "res_dat"   : "2016-08-08",
    "res_hentavd" : "mbhv",
    "res_hentavdeling" : "Moss hovedbibliotek",
    "ordord"    : "Heyerdahl, Thor",
    "tittel"    : "Påskeøya : en gåte blir løst",
    "material"  : "Bok",
    "qpos"      : 2
  }, {
    "idx"       : "res:42675,101",
    "res_dat"   : "2016-09-22",
    "res_hentavd" : "mbhv",
    "res_hentavdeling" : "Moss hovedbibliotek",
    "ordord"    : "McCourt, Frank",
    "tittel"    : "Engelen på det sjuende trinn : en irsk barndom",
    "material"  : "Bok",
    "qpos"      : 13
  }
  ]
}
```

Autentisering: Krever OAuth 2.0 autentisering.

getLnr

Henter ut lånernummer basert på e-postadresse og/eller fødselsnummer. Vi anbefaler at man bruker begge (i samme kall) for å få bedre treffsikkerhet.

Det er også mulig å kalle getLnr med *lnr=lånernummer* for å finne alder/kjønn på en gitt låner.

Parametere:

```
fnr // fødselsnummer
epost // e-postadresse
lnr // lånernummer
```

Returnerer: Et array som inneholder lånernummeret. Hvis låneren har nasjonalt lånernummer vil man få dette, ellers det lokale lånernummeret. I tillegg vil man få alder (alder er "0" på de som mangler fødselsdato) og kjønn (x=ikke satt, u=ukjent, m=mann, k=kvinne).

Eksempel 1.13. getLnr

`https://faretbib.bib.no/rest_service/webapi_getLnr/1.0/data?epost=roger@bibsyst.`

```
{
  "status"      : true,
  "message"     : "",
  "serverdate" : "2016-09-15T15:50:32",
  "lnr"        : "N000099999"
  "alder"      : "43",
  "kjønn"      : "m"
}
```

Status vil være true hvis forespørselen gikk igjennom. *lnr* vil være "" hvis oppslaget feilet (fant ikke låneren).

NB! Hvis man bruker fødselsnummer (*fnr*) MÅ man sende dette med i en POST. Fødselsnummeret er sensitivt nok til at det ikke bør havne i loggen til webtjeneren.

Autentisering: Krever Basic Authentication

bibauth

Brukes til å autentisere en låner med lånernummer og PIN.

Parametere:

lnr
pin

Returnerer info om kombinasjon *lnr/pin* er korrekt. *lnr* kan også være lånerens e-postadresse. Denne vil ikke bli godkjent hvis denne e-postadressen finnes på flere enn en låner.

Eksempel 1.14. bibauth

`https://host.bib.no/cgi-bin/rest_service/bibauth/1.0/data?lnr=N121212345&pin=134`

```
{
  "success" : true,
  "message" : "",
  "timeout" : 60
}
```

success vil være true dersom kombinasjonen stemmer.

Hvis det gjøres 3 forsøk på autentisering som feiler så kan det ikke gjøres andre autentiseringer på denne låneren før det antall sekunder som står i verdien *timeout* er passert.

Autentisering: Krever oppsett med filtrering på IP-adresse.

1.3. Endring av data

1.3.1. REST-servicer

Autentisering: Krever OAuth 2.0 autentisering.

addReservation

Legg inn en bestilling/reservering.

Parametre:

reservationPickUpBranch Angir hvor låneren ønsker å hente det reserverte materialet.

Eksempel 1.15. addReservation

Request:

`https://faretbib.bib.no/rest_service/oawebapi_addReservation/1.0/data?reservation`

`https://faretbib.bib.no/rest_service/oawebapi_addReservation/1.0/data?reservation`

Respos:

```
{
  "status"      : true,
  "message"     : "",
  "serverdate" : "2016-04-29T13:25:39"
}
```

For e-bøker (eBokBib) vil en slik bestilling bli utført umiddelbart/automatisk hvis det er ledige eksemplarer som låneren har tilgang til.

removeReservations

Fjern en eller flere reserveringer.

Eksempel 1.16. removeReservations

Request: `https://faretbib.bib.no/rest_service/oawebapi_removeReservations/1.0/da`

Respos:

```
{
  "status"      : true,
  "message"     : "",
  "serverdate" : "2016-04-29T13:28:04",
  "detailed_status" : [{
    "reservationId" : 7980,
    "status"        : true,
    "message"       : ""
  }]
}
```

getLoans

Liste over den innloggede lånerens lån.

Eksempel 1.17. getLoans

https://faretbib.bib.no/rest_service/oawebapi_getLoans/1.0/data

Respons:

User loan object:

```
{
  loanId: string, // Til bruk ved fornyelser
  localId: string,
  dueDate: string, // ISO 8601
  loanDate: string, // ISO 8601
  loanBranch: string, // Hvor lånet er gjort
  renewedToday: bool,
  loanIsRenewable: bool,
  loanIsFastTrack: bool,
  readyForPickup: bool,
  pickupReference: string,
  pickupBranch: string,
  title: string,
  authors: string[],

  ...
}
```

https://faretbib.bib.no/rest_service/oawebapi_getLoans/1.0/data

```
"status"      : true,
"message"     : "",
"serverdate"  : "2016-04-15T11:01:26",
"loans"       : [{
  "loanId"      : "ix:101,2",
  "authors"     : "Såtvedt, Olav",
  "title"       : "Giftskandalen",
  "dueDate"     : "2016-02-01",
  "isbn"        : "8210029940",
  "material"    : "Bok",
  "localId"     : 101,
  "bibliofilid" : 34963,
  "loanDate"    : "",
  "renewedToday" : false,
  "loanBranch"  : "",
  "loanIsRenewable" : true,
  "readyForPickup" : true,
  "loanIsFastTrack" : false,
  "pickupReference" : "3\ 10",
  "pickupBranch" : "khbv"
}, {
  "loanId"      : "ix:202,1",
  "authors"     : "",
  "title"       : "Godt sagt : sitater, aforismer, visdomsord, ordspråk",
  "dueDate"     : "2016-05-13",
```

```
    "isbn"           : "8202117623",
    "material"      : "Bok",
    "localId"       : 202,
    "bibliofilid"   : 312848,
    "loanDate"      : "",
    "renewedToday"  : true,
    "loanBranch"    : "",
    "loanIsRenewable" : true
    "readyForPickup" : false,
    "loanIsFastTrack" : false
    "pickupReference" : "",
    "pickupBranch"  : ""
  }, {
    "loanId"         : "ix:1009,10",
    "authors"        : "Golding, William",
    "title"          : "Fluenes herre",
    "dueDate"        : "2016-04-28",
    "isbn"           : "8205170304",
    "material"       : "Bok",
    "localId"        : 1009,
    "bibliofilid"    : 83207,
    "loanDate"       : "2016-03-31",
    "renewedToday"   : false,
    "loanBranch"     : "khbv",
    "loanIsRenewable" : true
    "readyForPickup" : true,
    "loanIsFastTrack" : false
    "pickupReference" : "1\ / 10",
    "pickupBranch"   : "khbv"
  }, {
    "loanId"         : "ix:5003,3",
    "authors"        : "Ludlum, Robert",
    "title"          : "Gåten Jason Bourne : 2",
    "dueDate"        : "2016-05-13",
    "isbn"           : "82-590-0344-9",
    "material"       : "Bok",
    "localId"        : 5003,
    "bibliofilid"    : 1125048,
    "loanDate"       : "2016-03-31",
    "renewedToday"   : true,
    "loanBranch"     : "khbv",
    "loanIsRenewable" : true
    "readyForPickup" : false,
    "loanIsFastTrack" : false
    "pickupReference" : "",
    "pickupBranch"   : ""
  }
}
}
```

renewLoans
Forny lån.

Parametre:

loanIds (loanIds skal være en liste med *loanId* fra *getLoans* separert med semikolon)

Returnerer: Et array med de lånene som er fornyet, evt. med status som forklarer problemer. Mulige problemer er: "Not on loan", "Not renewable", "Maximum loan period achieved", "Requested by another patron", "ILL not renewable".

Eksempel 1.18. renewLoans

https://faretbib.bib.no/rest_service/oawebapi_renewLoans/1.0/data?loanIds=ix:5003

```
{
  "status"      : true,
  "message"     : "",
  "serverdate"  : "2016-04-15T11:02:47",
  "renewstatus" : [{
    "loanId"      : "ix:5003,3",
    "renewstatus" : true,
    "renewmessage" : "",
    "authors"     : "Ludlum, Robert",
    "title"       : "Gåten Jason Bourne : 2",
    "dueDate"     : "2016-05-13",
    "isbn"        : "82-590-0344-9",
    "material"    : "Bok",
    "localId"     : 5003,
    "bibliofilid" : 1125048,
    "loanDate"    : "2016-03-31",
    "renewedToday" : true,
    "loanBranch"  : "khbv",
    "loanIsRenewable" : true
  }, {
    "loanId"      : "ix:1007,2",
    "renewstatus" : false,
    "renewmessage" : "Not on loan"
  }, {
    "loanId"      : "ix:202,432",
    "renewstatus" : false,
    "renewmessage" : "loanId not found"
  }
  ]
}
```

obskode

Setter og fjerner obskoder på lånere. (Om obskoder, se http://bibliofil.no/kundeinfo/dokumentasjon/web/ba2/obs_kd.html).

Parametre:

lnr // lånernummer

obs // mellomromseparert liste med obskoder som skal legges til eller slettes.

Eksempel 1.19. Obskoder

https://faretbib.bib.no/rest_service/webapi_obskode/1.0/data?lnr=10&obs=D

```
{
  "status"      : true,
  "message"     : "",
  "serverdate" : "2016-09-15T16:23:55",
  "obs"        : "D"
}
```

Svaret i *obs* vil være aktive obskoder på låneren (uten mellomrom). For å fjerne obskoder brukes minus (Eks: **-D**). Obskode "D" brukes for å gi låner tilgang til døgnåpent bibliotek. Obskode "B" brukes for å blokkere låner og bør sjekkes før man eventuelt setter "D".

Autentisering: Krever Basic Authentication

1.4. Test APllet

```
Server: https://bstestbib.bib.no
Lånernummer: 56789
Pinkode: 1234
Client: TestInc
OAuth client_id: "129c172ce37eddc6e955ec9502f2c453f3079f4bdb0c6e5bfd081a017115fc99"
Basic Auth: TestInc / testpass123
```

Låner 56789 har et utlån, en reservering, obskode "D" og en melding på låner.