# Konsepter og prinsipper

## Utforming av tjenester

Mandatet til prosjektgruppen var å etablere CRUD tjenester (Create, Read, Update, Delete) for NOARK5 standarden. Både tjenestene og datastrukturer er modellert i UML.

De aller fleste objekter i NOARK trenger operasjoner/tjenester for å opprette objekt, finne objekter, oppdatere objekter og i noen spesielle tilfeller slette objekter. I noen av kravene i NOARK er det også beskrevet egne tjenester som skal kunne utføres.

Det er valgt å spesifisere REST for tjenestene. Prinsippene og eksempler følger under, og ytterligere detaljer kan en finne i vedlegg 3.

### REST tjenestene

For REST er HATEOAS prinsipper fulgt slik at en klient skal fra en hoved-URL kunne navigere og oppdage selv alle mulig tjenester som kjernen tilbyr.

Dette gjøres med ressurslenker og relasjonslenker som inneholder beskrivelse av ressursen med eksempler på forespørsler, resultat og statuskoder.



Under følger eksempler fra tjenestene.

#### Oppkobling og ressurslenker

Oppkobling skjer mot en hoved-URL og er den eneste ressursen klienten trenger å vite for å starte interaksjon. Resten av endepunkter oppdages av klienten via relasjonsnøkler som beskriver hva ressursen kan brukes til.

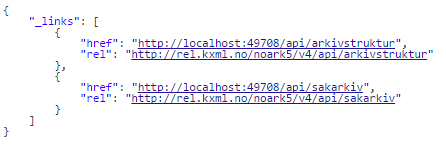
**Request**

GET <http://localhost:49708/api>

Accept: application/vnd.noark5-v4+json

**Response**

Content-Type: application/vnd.noark5-v4+json



Eksempelet viser at denne arkivkjernen støtter arkivstruktur (http://rel.kxml.no/noark5/v4/api/arkivstruktur) og sakarkiv (<http://rel.kxml.no/noark5/v4/api/sakarkiv>). Ved å følge **href** til disse relasjonsnøkler vil tilgjengelige ressurser innen disse områder annonseres på samme måte.

**Resultatkoder**

|  |  |  |
| --- | --- | --- |
| Statuskode | Beskrivelse |  |
| 200 | OK |  |
| 400 | BadRequest - ugyldig forespørsel | |  | | --- | |  | |
| 403 | Forbidden - ingen tilgang | |  | | --- | |  | |
| 404 | NotFound - ikke funnet | |  | | --- | |  | |
| 501 | NotImplemented - ikke implementert |  |

Alternativt som XML

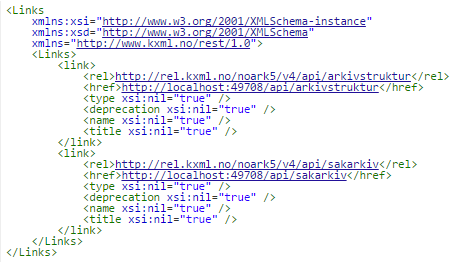
**Request**

GET <http://localhost:49708/api>

Accept: application/vnd.noark5-v4+xml

**Response**

Content-Type: application/vnd.noark5-v4+xml



­­­**href** kan være hva som helst og trenger ikke følge noe fast mønster for oppbygning av url. Mens **rel** (relasjonsnøkkelen) har faste verdier som beskriver hva ressursen kan brukes til. Denne kan klienten også åpne for å vise beskrivelse, eksempel på bruk, statuskoder og annet som er relevant for denne relasjonsnøkkelen.

**Relasjonsnøkler på rotnivå**

|  |  |
| --- | --- |
| Relasjonsnøkkel (rel) | Beskrivelse |
| <http://rel.kxml.no/noark5/v4/api/arkivstruktur> | Arkivkjerne støtter konformitetsnivå 1 arkivstruktur |
| <http://rel.kxml.no/noark5/v4/api/sakarkiv> | Arkivkjerne støtter konformitetsnivå for sakarkiv (2a) |

Relasjonsnøkler under de forskjellige konformitetsnivå listes ut i kapittel 7 sammen med beskrivelse av klasser.

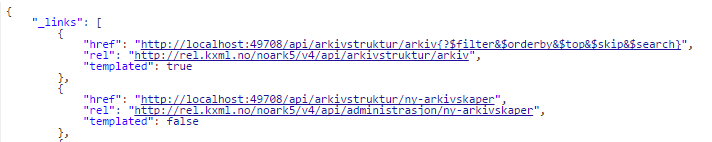
**Spesielle relasjonsnøkler**

|  |  |
| --- | --- |
| Relasjonsnøkkel (rel) | Beskrivelse |
| [self](http://rel.kxml.no/noark5/v4/api/arkivstruktur) | Brukes for å identifisere en ressurs, og kan brukes til oppdatering og sletting. |
| next | Brukes for å angi neste side ved serverstyrt resultatoppdeling |

Ressurser bør kun gjøres tilgjengelig i API når pålogget bruker har tilgang til disse. Hvis en bruker ikke har tilgang til å avslutte en mappe så bør ikke relasjonsnøkkel for dette annonseres i API for å gjøre det lettere å navigere til aktuelle funksjoner.

#### Finne objekter (Read)

For filter skal syntaks fra oData standarden (<http://docs.oasis-open.org/odata/odata/v4.0/os/part2-url-conventions/odata-v4.0-os-part2-url-conventions.html#_Toc372793790> ) benyttes. De ressurser som støtter filter skal annonserer dette under **\_links** med **templated=true** og parametre som kan brukes til dette i **href**. Typiske parametre er **$filter**, **$top**, **$skip** og **$orderby**. Alle lister med data bør støtte søk og filtrering.



Figur 1 anonsering av templated link for søk etter arkiv

Filter parametre som skal støttes er:

* $filter
* $top
* $skip
* $search
* $orderby

**Nivå på filter**

* Nivå basis (påkrevd):
  + Filter på direkte felter.
  + Filter på en-til-en gruppe relasjoner (blant annet kodelister)
* Nivå utvidet:
  + Filter på en-til-mange relasjoner (vha. 'any' og 'all' odata funksjonene)

##### **Filtrering**

Filtrering støttes med $filter parameter. Nedenfor følger en del eksempeler på ulike filtreringer med de innebygde odata-operatorene. Flere filtre kan kombineres med operatorene **and** og **or**.

Begynner med

**Syntaks:** startswith(feltnavn, ‘tekst’)

**Eksempel:**

/api/arkivstruktur/mappe/?$filter=startswith(tittel, 'allergisk testmappe')

Er lik

Syntaks: feltnavn eq verdi

**Eksempel:**

/api/arkivstruktur/mappe/?$filter=systemID eq '1'

Inneholder

**Syntaks:** substringof(‘tekst’, feltnavn)

**Eksempel:**

/api/arkivstruktur/mappe/?$filter=substringof('test', tittel)

Større enn

**Syntaks:** feltnavn gt verdi

**Eksempel:**

/api/arkivstruktur/registrering/?$filter=year(oppdatertDato) gt 2012

/api/sakarkiv/saksmappe?$filter=saksdato gt DateTime'2017-02-15'

Mindre enn

**Syntaks:** feltnavn lt verdi

**Eksempel:**

/api/sakarkiv/saksmappe?$filter=saksdato lt DateTime'2017-02-15'

Større enn eller lik

**Syntaks:** feltnavn ge verdi

**Eksempel:**

/api/sakarkiv/saksmappe?$filter=saksdato ge DateTime'2017-02-15'

Mindre enn eller lik

**Syntaks:** feltnavn le verdi

**Eksempel:**

/api/sakarkiv/saksmappe?$filter=saksdato le DateTime'2017-02-15'

Og

**Syntaks:** uttrykk and utrykk

**Eksempel:**

/api/sakarkiv/saksmappe/?$filter=saksdato gt DateTime'2017-02-10' and saksdato lt DateTime'2017-02-12'

Eller

**Syntaks:** uttrykk or utrykk

**Eksempel:**

/api/sakarkiv/saksmappe/?$filter=year(saksdato) gt 2014 or year(opprettetDato) gt 2014

**Flere eksempler på filtrering**

De to første mappene med test i tittelen

/api/arkivstruktur/mappe/?$top=2&$filter=substringof('test',tittel)

Mapper med graderingskode B

/api/arkivstruktur/mappe/?$filter=gradering/graderingskode/kode eq 'B'

Mapper med merknader som har merknadstype B

|  |  |  |
| --- | --- | --- |
| Eksempel | Forklaring | Nivå |
| ../arkivstruktur/mappe/?$expand=merknad&$filter=merknad/any(m: m/merknadstype/kode eq 'B') | Mapper med merknader som har merknadstype B | utvidet |
| [../arkivdel/1235/mappe?$top=2&$filter=contains(‘testmappe’, tittel) eq true](http://n5test.kxml.no/api/arkivstruktur/Arkivdel/1235/mappe?$top=2&$filter=tittel%20eq%20%E2%80%98testmappe%E2%80%99) $orderby=oppdatertDato desc | De to første mapper hvor testmappe er en del av tittel sortert synkende på oppdatertDato | basis |
| ../api/arkivstruktur/Mappe?$filter=klasse/klasseID eq '12/2' and klasse/klassifikasjonssystem/klassifikasjonstype/kode eq 'GBNR' | Mappe med klassering på eiendom | utvidet |
| ../api/arkivstruktur/Mappe?$filter=klasse/klasseID eq '12345678901' and klasse/klassifikasjonssystem/klassifikasjonstype/kode eq 'PNR' | Mappe med klassering på fødselsnr | utvidet |
| ../api/arkivstruktur/Mappe?$filter=klasse/klasseID eq '123456789' and klasse/klassifikasjonssystem/klassifikasjonstype/kode eq 'ORG' | Mappe med klassering på organisasjonsnr | utvidet |
| ../api/sakarkiv/Saksmappe/?$filter=sakspart/any(s:s/Default.SakspartPersonType/foedselsnummer eq '12334566') | Saksmapper med sakspart(SakspartPerson) med gitt fødselsnr | utvidet |
| ../api/sakarkiv/Saksmappe/?$filter=sakspart/any(s:s/Default.SakspartEnhetType/organisasjonsnummer eq '12334566') | Sakspart med organisasjonsnr | utvidet |
| ..api/sakarkiv/journalpost/?$filter=korrespondansepart/any(s:s/Default.KorrespondansepartPersonType/foedselsnummer eq '12334566') | Korrespondansepart med fødselsnummer | utvidet |
| ..api/arkivstruktur/mappe/?$filter=nasjonalidentifikator/any(i: i/Default.BygningType/byggidentifikator/bygningsNummer eq '12345678') | Nasjonal identifikator med bygningsnr | utvidet |

**Søk**

$search brukes for generelt søk. Arkivkjernen bestemmer hvordan denne er implementert med hensyn på hvilke felter den inkluderer i søk og om for eksempel innhold i dokumenter er med.

Eksempel på hvordan syntaks for et søk i et arkiv kan se ut:

/api/arkivstruktur/arkiv?$search='test'

**Sortering**

$orderby brukes for å angi sortering av resultat etter gitte felter.

**Resultatoppdeling (Paginering)**

På klientsiden kan $top og $skip brukes sammen for å angi hvilken side av søkeresultatet en ønsker returnert. $top gir antallet som skal returneres, og $skip gir antallet en skal hoppe over og ikke inkludere i resultatet.

Serverstyrt resultatoppdeling kan settes av arkivkjernen med PageSize. Pagesize setter max antall som kan returneres fra arkivkjerne og kjerne må returnere en next link som gir neste siden.

**Filter på underobjekter**

Any eller All brukes for å filtrere på navigerbare objekter. Det kan være begrensninger på hvor mange nivå/dybde en arkivkjerne støtter.

|  |  |
| --- | --- |
| localhost:49708/api/sakarkiv/saksmappe?$filter=nasjonalidentifikator/any(i: i/Default.BygningType/byggidentifikator/bygningsNummer eq '12345678') |  |
| localhost:49708/api/sakarkiv/saksmappe?$filter=nasjonalidentifikator/any(i: i/Default.BygningType/byggidentifikator/bygningsNummer eq '12345678') |  |

**Resultat med underobjekter**

$expand brukes for å inkludere underobjekter i resultat. Det kan være begrensninger på hvor mange nivå en arkivkjerne støtter. Som standard skal ikke underobjekter returneres hvis dette ikke spesifiseres med $expand. Hvor mange nivåer som støttes settes opp i kjernen med MaxExpansionDepth.

**Filter og tilgangsstyring**

Ved søk skal arkivkjernen ta hensyn til tilgangsrettigheter slik at brukere ikke får uautorisert tilgang til informasjon. Er informasjonen unntatt offentlighet, skjermet eller gradert så skal ikke uautoriserte brukere få tilgang til dette. Dette kan bety at en bruker har lov til å registrere et objekt, men ikke rettigheter til å vise dette etterpå.

Resultatkoder ved navigering/søk

|  |  |  |
| --- | --- | --- |
| Statuskode | Beskrivelse |  |
| 200 | OK |  |
| 400 | BadRequest - ugyldig forespørsel | |  | | --- | |  | |
| 403 | Forbidden - ingen tilgang | |  | | --- | |  | |
| 404 | NotFound - ikke funnet | |  | | --- | |  | |
| 500 | InternalServerError – generell feil på server |  |
| 501 | NotImplemented - ikke implementert |  |

#### Opprette objekter (Create)

For å opprette objekter må først ressurslenke finnes basert på relasjonsnøkkel.

For eksempel kan en opprette mapper på arkivdel, og da vil \_Links under en arkivdel inneholde relasjonsnøkkelen rel=»<http://rel.kxml.no/noark5/v4/api/arkivstruktur/ny-mappe/>» om bruker har lov til å opprette mapper på denne arkivdelen. Den aktuelle ressurslenke kan være <http://n5test.kxml.no/api/arkivstruktur/Arkivdel/12345/ny-mappe>. Denne kan brukes til både GET og POST forespørsel.

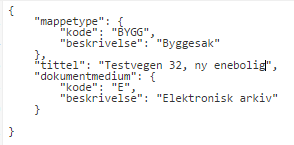
GET-forespørselen forhåndsutfyller en lovlig objektstruktur og gir relasjonslenker til aktuelle kodelister. En slik forespørsel oppretter ikke noe objekt og returverdien refererer ikke heller til et objekt i databasen, og er derfor uten «self»-relasjon.



Klienten sender en POST forespørsel med en lovlig objektstruktur til gitt url. Responsen gir statuskode 201 Created om objektet ble opprettet korrekt og komplett objekt samt location header for lese eller endre url.

POST til <http://n5test.kxml.no/api/arkivstruktur/Arkivdel/12345/ny-mappe>

Content-Type: application/vnd.noark5-v4+json



**Resultat**

201 Opprettet

Location →

http://localhost:49708/api/arkivstruktur/Mappe/a043d07b-9641-44ad-85d8-056730bc89c8



Figur 2 respons fra opprett mappe (eksempel avkortet for liste over links)

Resultatkoder ved oppretting av objekt

|  |  |  |
| --- | --- | --- |
| Statuskode | Beskrivelse |  |
| 200 | OK |  |
| 201 | Created - opprettet |  |
| 400 | BadRequest - ugyldig forespørsel | |  | | --- | |  | |
| 403 | Forbidden - ingen tilgang | |  | | --- | |  | |
| 404 | NotFound - ikke funnet | |  | | --- | |  | |
| 409 | Conflict – objektet kan være endret av andre |  |
| 500 | InternalServerError – generell feil på server |  |
| 501 | NotImplemented - ikke implementert |  |

Heleide objekter(komposisjoner) kan opprettes sammen med hovedobjektet og inngår i dens lovlige objektstruktur. For eksempel merknad på en mappe kan registreres sammen med registreringen av mappe.

#### Preutfylling av objekt

Ved å bruke GET på for eksempel ny-mappe (<http://rel.kxml.no/noark5/v4/api/arkivstruktur/ny-mappe/>) så kan arkivkjerne preutfylle og foreslå vanlige data for et objekt basert på pålogget bruker samt annonsere hvor diverse lovlige koder kan hentes fra slik som mappetype og dokumentmedium.



#### Oppdatere objekter (Update)

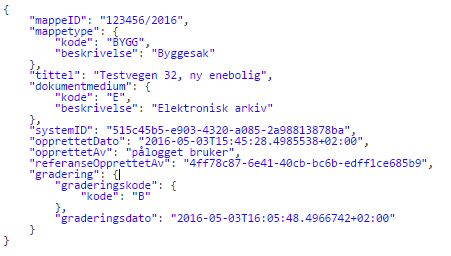
Alle ressurser kan med sin relasjonslenke rel=»self» og ressurslenke (href) benytte denne til oppdatering.

For oppdatering sender klienten en PUT forespørsel med alle data for en lovlig objektstruktur. Alle egenskaper må være med, med unntak av underobjekter som har en mange relasjon (0..\* eller 1..\*) i oppdatering av et objekt. Underobjekter må oppdateres separat med sine resurslenker.

For å hindre at data blir oppdatert samtidig av forskjellige brukere og overskrevet med gamle data så må kjernen sjekke innkomne objekt og lagret objekt. ETag (<http://en.wikipedia.org/wiki/HTTP_ETag> ) skal benyttes for å støtte «optimistic concurrency control». Om det oppstår konflikt så kan resultatkode 409 benyttes. Da må klient hente opp ny versjon fra arkivkjerne og gjøre fletting av data mellom server og klient. For å redusere risikoen for konflikt bør derfor klienten alltid hente en fersk utgave av objektet med en GET-forespørsel og deretter oppdatere opbjektet med en PUT-forespørsel.

PUT til http://n5test.kxml.no/api/arkivstruktur/Mappe/a043d07b-9641-44ad-85d8-056730bc89c8

Content-Type: application/vnd.noark5-v4+json



**Resultat**

200 OK

Location →

http://localhost:49708/api/arkivstruktur/Mappe/a043d07b-9641-44ad-85d8-056730bc89c8



Figur 3 respons fra oppdatering av mappe med graderingsinformasjon (eksempel avkortet ved links liste)

Resultatkoder ved oppdatering av objekt

|  |  |  |
| --- | --- | --- |
| Statuskode | Beskrivelse |  |
| 200 | OK |  |
| 400 | BadRequest - ugyldig forespørsel | |  | | --- | |  | |
| 403 | Forbidden - ingen tilgang | |  | | --- | |  | |
| 404 | NotFound - ikke funnet | |  | | --- | |  | |
| 409 | Conflict – objektet kan være endret av andre |  |
| 500 | InternalServerError – generell feil på server |  |
| 501 | NotImplemented - ikke implementert |  |
| 409 | Conflict – objektet kan være endret av andre |  |

#### Utvid objekter til andre typer

Noen objekter kan utvides fra sin basistype til en annen subtype. Dette gjelder for eksempel Mappe og Saksmappe. Dette annonseres ved hjelp av **utvid-til-xx** metodene.

Ved uthenting av en mappe vil du få følgende relasjon tilbake:

{

"rel": "http://rel.kxml.no/noark5/v4/api/sakarkiv/utvid-til-saksmappe",

"href": "http://n5test.kxml.no/api/sakarkiv/Saksmappe/1/utvid-til-saksmappe",

"templated": false

}

Ved å kjøre PUT-forespørsel på angitt href med tilhørende felter som er påkrevd for saksmappe så

skal objektet utvides til å bli en saksmappe.

**PUT** [**http://n5test.kxml.no/api/sakarkiv/Saksmappe/1/utvid-til-saksmappe**](http://n5test.kxml.no/api/sakarkiv/Saksmappe/1/utvid-til-saksmappe)

Content-Type: application/vnd.noark5-v4+json

{

"saksansvarlig": "Arne",

"saksdato": "2017-12-08T00:00:00",

"saksstatus": { "kode": "R", "beskrivelse": "Opprettet av saksbehandler"}

}

Respons skal være den nye saksmappen. Merk at **self** nå peker på saksmappe og ikke mappe.

{

"saksdato": "2017-12-08T00:00:00",

"saksansvarlig": "Henning",

"saksstatus": {

"kode": "R",

"beskrivelse": "Opprettet av saksbehandler"

},

"mappeID": "1/2014",

"tittel": "klok testmappe 1",

"offentligTittel": "Dette er en offentlig tittel \*\*\*\*",

"gradering": {

"graderingskode": {

"kode": "B"

},

"graderingsdato": "2017-12-08T15:32:10.739027+01:00",

"\_links": []

},

/// Resten av objektet utelatt

"\_links": [

{

"rel": "self",

"href": "http://n5test.kxml.no/api/sakarkiv/saksmappe/1",

"templated": false

},

}

Resultatkoder ved utvidelse av objekt

|  |  |  |
| --- | --- | --- |
| Statuskode | Beskrivelse |  |
| 200 | OK |  |
| 400 | BadRequest - ugyldig forespørsel | |  | | --- | |  | |

Resultatkode 400 leveres dersom id til eksterende mappe er ugyldig eller det mangler påkrevde felter.

#### Oppdatere referanser mellom objekter

Relasjoner kan angis ved tildelte attributter eller via plassering på gitt url. For eksempel ny mappe knyttes til arkivdel ved at url til ny mappe også inneholder hvilke arkivdel denne skal opprettes på. Egne attributter kan for eksempel være referanseForeldremappe for å lage undermapper.

Mer generelt kan klienter benytte href for rel=self for aktuelle objekter sammen med $ref parameter for å slette, endre eller opprette referanser mellom objekter.

**For å opprette ny referanse**

Her opprettes ny referanse mellom registrering og dokumentbeskrivelse.

POST http://localhost:49708/api/arkivstruktur/registrering/cf8e1d0d-e94d-4d07-b5ed-46ba2df0465e/dokumentbeskrivelse/$ref?$id=http://localhost:49708/api/arkivstruktur/Dokumentbeskrivelse/1fa94a89-3550-470b-a220-92dd4d709044

Resultatkode 204 – No Content

**For å oppdatere/flytte referanse**

Her flyttes mappen fra en arkivdel til en annen.

PUT http://localhost:49708/api/arkivstruktur/mappe/cf8e1d0d-e94d-4d07-b5ed-46ba2df0465e/arkivdel/$ref

Body:

http://localhost:49708/api/arkivstruktur/arkivdel/092e497a-a528-4121-8f22-fbc78fa6c930

Resultatkode 204 – No Content

**For å slette referanser fra en liste**

Ved sletting av referanser i en liste skal $id-parameteren benyttes. Her slettes referansen til dokumentbeskrivelse fra registrering.

DELETE http://localhost:49708/api/arkivstruktur/registrering/cf8e1d0d-e94d-4d07-b5ed-46ba2df0465e/dokumentbeskrivelse/$ref?$id=http://localhost:49708/api/arkivstruktur/Dokumentbeskrivelse/092e497a-a528-4121-8f22-fbc78fa6c930

Resultatkode 204 – No Content

**For å slette en enkelt-referanse**

Ved sletting av en enkelt-referanse så skal ikke $id-parameteren benyttes. Her slettes referansen til registrering fra dokumentbeskrivelse.

DELETE http://localhost:49708/api/arkivstruktur/dokumentbeskrivelse/092e497a-a528-4121-8f22-fbc78fa6c930/registrering/$ref

Resultatkode 204 – No Content

Resultatkoder ved oppdatering av referanser til objekt

|  |  |  |
| --- | --- | --- |
| Statuskode | Beskrivelse |  |
| 200 | OK |  |
| 204 | NoContent |  |
| 400 | BadRequest - ugyldig forespørsel | |  | | --- | |  | |
| 403 | Forbidden - ingen tilgang | |  | | --- | |  | |
| 404 | NotFound - ikke funnet | |  | | --- | |  | |
| 409 | Conflict - objektet kan være endret av andre | |  | | --- | |  | |
| 500 | InternalServerError – generell feil på server |  |
| 501 | NotImplemented - ikke implementert |  |

#### Slette objekter (Delete)

Klienten sender en DELETE forespørsel på aktuell ressurs(url). Alle ressurslenker med rel=»self» kan potensielt slettes om bruker har nødvendige rettigheter. Respons gir statuskode 204 om ressursen er korrekt slettet.

Et viktig krav i Noark 5 er at arkiverte elektroniske dokumenter ikke skal kunne slettes. Kontrollert sletting skal bare kunne foretas av autoriserte brukere i forbindelse med kassasjon

Sletting av Arkivdel – Kan også gjøres med oppdatering av arkivdel hvor det legges inn informasjon om sletting.

Sletting av Dokumentbeskrivelse – her er flere type sletting så må gjøres med oppdatering?

De som ikke har sletting datatype, hvordan skal de merkes/fjernes?Krav om logging nok?Skal ikke være mulig å slette uansett hvor mye rettigheter en bruker har? – Arkivverket må avklare dette



Resultatkoder ved sletting av objekt

|  |  |  |
| --- | --- | --- |
| Statuskode | Beskrivelse |  |
| 200 | OK |  |
| 204 | NoContent – slettet ok |  |
| 400 | BadRequest - ugyldig forespørsel | |  | | --- | |  | |
| 403 | Forbidden - ingen tilgang | |  | | --- | |  | |
| 404 | NotFound - ikke funnet | |  | | --- | |  | |
| 409 | Conflict - objektet kan være endret av andre | |  | | --- | |  | |
| 500 | InternalServerError – generell feil på server |  |
| 501 | NotImplemented - ikke implementert |  |

#### Overføringsformat

Innholdstyper (Content-Type) som skal brukes:

|  |
| --- |
| Innholdstype (Content-Type) |
| application/vnd.noark5-v4+json |
| application/vnd.noark5-v4+xml |

Overføringsformat skal være i henhold til følgende skjema for begge innholdstyper:

* <http://skjema.kxml.no/arkivverket/noark5/v4.0/>

Datoformat skal være angitt ihht <http://www.w3.org/TR/NOTE-datetime>

#### Hente og overføre filer

Ved navigering til dokumentobjekt så kan selve filen også åpnes ved å følge referanseDokumentfil eller href til relasjonsnøkkel http://rel.kxml.no/noark5/v4/arkivstruktur/fil

GET <http://localhost:49708/api/arkivstruktur/Dokumentobjekt/a895c8ed-c15a-43f6-86de-86a626433785>



GET <http://localhost:49708/api/arkivstruktur/Dokumentobjekt/a895c8ed-c15a-43f6-86de-86a626433785/referanseFil>

Gir Content-type=filens mime type feks “application/pdf” og filen streames til klient

**Overføre små filer**

For å overføre en ny fil brukes POST til href til rel=http://rel.kxml.no/noark5/v4/api/arkivstruktur/fil med headere for content-type og content-length.

POST <http://localhost:49708/api/arkivstruktur/Dokumentobjekt/a895c8ed-c15a-43f6-86de-86a626433785/referanseFil>  
Content-Type: application/pdf  
Content-Length:111111

Pdf data

**Overføre små filer**

For store filer(over 150MB) så kan filen overføres i bolker. Prosessen for å overføre store filer er inspirert av APIet til Google Drive, <https://developers.google.com/drive/v3/web/resumable-upload>.

For å starte en opplastingssesjon:

1. Send en POST til href til rel=http://rel.kxml.no/noark5/v4/api/arkivstruktur/fil

Headeren Content-Length settes til 0  
Headeren X-Upload-Content-Type settes til filens MIME-type  
Headeren X-Upload-Content-Length settes til filens totalstørrelse

1. Responsen du mottar vil inneholde en Location-Header som inneholder en sesjons-URI som skal benyttes i en PUT-forespørsel for å overføre filen i bolker.
2. Deretter overføres filen med en PUT-forespørsel. Responsen fra serveren inneholder en Range-header, hvor øvre verdi benyttes som start verdi i Content-Range i neste oversending.

Headeren Content-Range settes for å angi hvor mye av filen som blir oversendt.

1. Når siste overføring er gjort så returneres statuskode 201 Created.

Komplett eksempel

Opprett sesjon:

POST <http://localhost:49708/api/arkivstruktur/Dokumentobjekt/a895c8ed-c15a-43f6-86de-86a626433785/referanseFil>  
Content-Length:0  
X-Upload-Content-Type: image/jpeg  
X-Upload-Content-Length: 2000000

Respons: 200 OK

Location: <http://localhost:49708/api/arkivstruktur/Dokumentobjekt/a895c8ed-c15a-43f6-86de-86a626433785/referanseFil?filsesjon=abc1234567>

Last opp første del:

PUT <http://localhost:49708/api/arkivstruktur/Dokumentobjekt/a895c8ed-c15a-43f6-86de-86a626433785/referanseFil?filsesjon=abc1234567>  
Content-Length: 524288  
Content-Type: image/jpeg  
Content-Range: bytes 0-524287/2000000

Respons: 200 OK

Location: <http://localhost:49708/api/arkivstruktur/Dokumentobjekt/a895c8ed-c15a-43f6-86de-86a626433785/referanseFil?filsesjon=abc1234567>  
Range: bytes=0-524287

Last opp siste del:

PUT <http://localhost:49708/api/arkivstruktur/Dokumentobjekt/a895c8ed-c15a-43f6-86de-86a626433785/referanseFil?filsesjon=abc1234567>  
Content-Length: 524288  
Content-Type: image/jpeg  
Content-Range: bytes 524287-2000000/2000000

Respons: 201 Created

Resultatkoder for opplasting av filer

|  |  |  |
| --- | --- | --- |
| Statuskode | Beskrivelse |  |
| 200 | OK |  |
| 201 | Created - opprettet |  |
| 204 | NoContent – slettet ok |  |
| 400 | BadRequest - ugyldig forespørsel | |  | | --- | |  | |
| 403 | Forbidden - ingen tilgang | |  | | --- | |  | |
| 404 | NotFound - ikke funnet | |  | | --- | |  | |
| 409 | Conflict - objektet kan være endret av andre | |  | | --- | |  | |
| 415 | UnsupportedMediaType – filtypen støttes ikke |  |
| 500 | InternalServerError – generell feil på server |  |
| 501 | NotImplemented - ikke implementert |  |
| 503 | ServiceUnavailable – tjeneste utilgjengelig |  |

## Validering av data

For de fleste objekter i NOARK5 så er det knyttet forskjellige krav til hva som er lovlige verdier og strukturer. Disse kravene må implementeres i tjenestegrensesnitt/arkivkjerne som forretningsregler og sørge for at data er konsistente.

Restriksjoner som er dokumentert under hvert objekt i informasjonsmodellen skal valideres av kjernen. For eksempel hvis en mappe er avsluttet så skal det ikke være mulig å registrere flere registreringer på denne (jfr krav 5.4.7).

## Identifikatorer

SystemID brukes som entydig identifikator for alle objekter

SystemID tildeles av kjernen og skal være konsistente over tid. Arkivkjernen må sørge for at dette blir en unik og persistent identifikator på tvers av andre system. Den skal kunne brukes til å identifisere og referere til objekter liggende i andre filer eller databaser.

## Utvidelsesmuligheter

Virksomhetsspesifikke metadata kan brukes for å legge ved mer data på enkelte objekter i kjernen.

Søk i virksomhetsspesifikke data dekkes ikke av NOARK 5 tjenestegrensesnitt, men den enkelte arkivleverandør kan tilby tjenester som tilbyr søk i virksomhetsspesifikke data