

Nemus API – Filkomplettering

Komplettering med filer

Detta dokument beskriver hur filkomplettering används i Nemus API och är avsett för systemleverantörer som integrerar mot API:et.

Filkomplettering (komplettering med filer) används för att skicka in filer kopplade till ett befintligt ärende, till exempel som svar på ett utgående dokument från Nemus.

Versionshantering

API-kontraktet för filkomplettering underhålls löpande.
Förändringar som påverkar integrationer kommuniceras i god tid.

1. Översikt

Filkomplettering används för att skicka in filer kopplade till ett **befintligt ärende** i Nemus.

Till skillnad från tidigare inlämningsätt, som är XSD-baserade och XML-endast, är filkomplettering **OpenAPI-baserad** och stöder både **JSON och XML**.

Kontraktet består av två nivåer:

1. Ett **yttre kontrakt** (OpenAPI) som beskriver hur filkompletteringen ska struktureras.
2. Ett **inre schema per handling**, angivet med `schemaName` och `schemaVersion`, som styr vad handlingen får innehålla.

2. Endpoint

```
POST /arenden/{arendeId}/filkomplettering
```

Path-parameter

Namn	Typ	Beskrivning
<code>arendeId</code>	UUID (GUID)	Nemus ärende-ID som dokumenten ska kopplas till

3. Format och content-types

Request

```
application/json  
application/xml
```

Response

```
application/json
application/xml
```

API:t använder content negotiation. Samma format som används i requesten returneras i response, inklusive vid fel.

4. Yttre kontrakt – Filkomplettering

Request body representerar en filkomplettering som innehåller en eller flera handlingar.

Struktur (översikt)

```
ArendeFilkomplettering
├─ handlingar[] (0..n)
│  ├─ schemaName (string, required)
│  ├─ schemaVersion (int, optional)
│  ├─ svarPaDokumentId (uuid, optional)
│  └─ filer[] (0..n)
│     ├─ ursprungsFilnamn
│     ├─ filTyp
│     ├─ dokumentTyp
│     └─ filData (Base64)
└─ content (fri struktur, beroende på schema)
```

Det yttre kontraktet validerar **inte** innehållet i `content` eller värden i dokumentmetadata. Det styrs i stället av det inre schema som anges per handling.

5. Inre schema för handlingar

Varje handling måste ange vilket **inre schema** som gäller via:

```
handlingar[].schemaName
handlingar[].schemaVersion
```

Det inre schemat avgör:

- om `content` ska användas eller inte
 - vilka dokumenttyper som är tillåtna för filer
 - vilka filtyper som är giltiga
-

5.1 Tillgängliga scheman (initialt)

I första versionen stöds ett inre schema:

schemaName	schemaVersion	Beskrivning
filinskickObjekttillsyn	1	Filinlämning för objekttillsyn

5.2 Särskilda regler för `filinskickObjekttillsyn` version 1

För detta schema gäller följande:

- content ska **inte** skickas (utelämnas eller sätts till null)
- Endast fältet filer används
- Varje fil måste ange:
 - dokumentTyp
 - filTyp
- Tillåtna värden för dessa fält är fastställda av det inre schemat (filinskickObjekttillsyn_1.schema.json).

Tillåtna filTyp :

pdf

Exempel på tillåtna dokumentTyp :
(Listan nedan är inte fullständig.)

svarpakommunikation
hansynsredovisning
karta
fullmakt
fotodokumentation
traktdirektiv
inventeringsprotokoll
beslutannanmyndighet

5.3 Exempel – JSON (filinskickObjekttillsyn v1)

```
{
  "handlingar": [
    {
      "schemaName": "filinskickObjekttillsyn",
      "schemaVersion": 1,
      "filer": [
        {
          "ursprungsFilnamn": "omradeskarta.pdf",
          "filTyp": "pdf",
          "dokumentTyp": "karta",
          "filData": "JVBERi0xLjQKJ..."
        },
        {
          "ursprungsFilnamn": "fotografier.pdf",
          "filTyp": "pdf",
          "dokumentTyp": "fotodokumentation",
          "filData": "JVBERi0xLjQKJ..."
        }
      ]
    }
  ]
}
```

Notera att content inte används för detta schema.

5.4 Exempel – XML (filinskickObjekttillsyn v1)

```
<ArendeFilkomplettering>
  <handlingar>
    <schemaName>filinskickObjekttillsyn</schemaName>
    <schemaVersion>1</schemaVersion>
    <filer>
```

```
<fil>
  <ursprungsFilnamn>omradeskarta.pdf</ursprungsFilnamn>
  <filTyp>pdf</filTyp>
  <dokumentTyp>karta</dokumentTyp>
  <filData>JVBERi0xLjQKJ...</filData>
</fil>
<fil>
  <ursprungsFilnamn>fotografier.pdf</ursprungsFilnamn>
  <filTyp>pdf</filTyp>
  <dokumentTyp>fotodokumentation</dokumentTyp>
  <filData>JVBERi0xLjQKJ...</filData>
</fil>
</filer>
</handlingar>
</ArendeFilkomplettering>
```

Bilagor

Följande filer tillhandahålls som komplement till detta dokument:

- **filinskickObjekttillsyn_1.schema.json**

JSON Schema som beskriver tillåtna metadata för filer vid användning av `schemaName = filinskickObjekttillsyn`, version 1. Schemat används som referens för tillåtna värden, inte som ett fristående API-kontrakt.

- **utgaendeDokument_4.xsd**

XML Schema för utgående dokument vid hämtning av dokument från Nemus med `schemaVersion = 4`. Detta schema gäller endast för dokument från Skogsstyrelsen, inte för filkomplettering.

6. Responses

6.1 202 Accepted

Filkompletteringen är mottagen och registrerad.

- Response body innehåller ett kvitto med identifierare och status
- Format: samma som request (JSON eller XML)

6.2 400 Bad Request

Filkompletteringen kunde inte tolkas eller uppfyllde inte förväntad struktur.

- Format: samma som request (JSON eller XML)
- Response body följer standardtypen `ProblemDetails`
- Eventuella valideringsfel återfinns i `extensions.errors`

7. Utgående dokument – schemaVersion i URI

När dokument hämtas från Nemus används ett **versionsstyrt XML-schema** som anges via query-parameter.

Endpoint

GET /arenden/{arendeId}/dokument?schemaVersion={version}

Tillgängliga versioner

schemaVersion	Schema	Kommentar
2	utgaendedokument_2.xsd	Grundschema
3	utgaendedokument_3.xsd	Kan inkludera geometrier för beslut
4	utgaendedokument_4.xsd	Ny version som inkluderar information för kompletteringar

Viktigt att notera

schemaVersion för **utgående dokument**:

- styr **endast formatet på svaret från Nemus**
- är **oberoende** av schemaVersion som används vid filkomplettering

7.1 Svar på utgående dokument (svarsinformation)

I version 4 av schemat för utgående dokument (utgaendeDokument_4.xsd) kan ett dokument innehålla sektionen **svarsinformation**.

Denna information används för att styra **hur och om mottagaren ska svara** på det utgående dokumentet genom filkomplettering.

7.1.1 Vad är svarsinformation?

svarsinformation innehåller metadata som beskriver:

- vilket dokument svaret ska kopplas till
- hur länge det är möjligt att svara
- vilket schema som ska användas vid inlämning av svaret

Informationen är avsedd att konsumeras av systemet som hämtar dokumentet och används vid efterföljande filkomplettering.

7.1.2 Fält i svarsinformation

Fält	Beskrivning
dokumentId	Identifierare (GUID) för det utgående dokumentet. Ska användas som referens vid svar.
stopdatum	(valfritt) Senaste datum/tidpunkt då svaret accepteras.
svarSchema	(valfritt) Namn på schema som ska användas vid svar.

Fält	Beskrivning
medKod	(valfritt) Kod som kan användas vid svar via webbgränssnitt.

7.1.3 Hur svarsinformation används vid filkomplettering

När ett utgående dokument innehåller `svarsinformation` ska ett svar skickas in via filkomplettering (`POST /arenden/{arendeId}/filkomplettering`).

Svarsinformation fungerar som länken mellan ett utgående dokument och efterföljande filkomplettering.

Vid filkompletteringen gäller följande:

- Fältet `svarPaDokumentId` i handlingen ska sättas till värdet från `svarsinformation.dokumentId`
- Det schema som anges i `svarsinformation.svarSchema` anger vilket **inre schema** (`schemaName` och `schemaVersion`) som ska användas
- Eventuellt `stoppdatum` anger senast tillåtna tidpunkt för inlämning

7.1.4 Initialt supported svarschema

I första versionen gäller:

svarSchema	schemaName	schemaVersion
filinskickObjekttillsyn_1.schema.json	filinskickObjekttillsyn	1

För detta schema gäller samma regler som beskrivs i avsnittet *Inre schema för handlingar*:

- `content` ska inte användas
- Endast filer skickas med
- Filmetadata måste följa schemat

7.1.5 Exempel – utgående dokument med svarsinformation (konceptuellt)

```
<svarsinformation>
  <dokumentId>b23cfd4-7b4d-4f82-9e92-4ad91a0c9f3d</dokumentId>
  <svartyp>BegardKomplettering</svartyp>
  <stoppdatum>2026-06-30T23:59:59</stoppdatum>
  <svarSchema>filinskickObjekttillsyn_1.schema.json</svarSchema>
</svarsinformation>
```

7.1.6 Exempel – svar via filkomplettering (JSON)

```
{
  "handlingar": [
```

```
{
  "schemaName": "filinskickObjekttillsyn",
  "schemaVersion": 1,
  "svarPaDokumentId": "b23cfd4-7b4d-4f82-9e92-4ad91a0c9f3d",
  "filer": [
    {
      "ursprungsFilnamn": "komplettering.pdf",
      "filTyp": "pdf",
      "dokumentTyp": "svarpakommunikation",
      "filData": "JVBERi0xLjQKJ..."
    }
  ]
}
```

I detta exempel:

- Svaret är kopplat till ett utgående dokument via `svarPaDokumentId`
- Inre schema följer `filinskickObjekttillsyn` version 1
- Inget `content` används

7.1.7 Viktiga anmärkningar

- Avsaknad av `svarsinformation` innebär att inget strukturerat svar förväntas via filkomplettering
- `svarsinformation` gäller endast för det individuella utgående dokumentet
- `schemaVersion` för utgående dokument är oberoende av `schemaVersion` som används vid filkomplettering

8. Stabilitet och förändringar

De scheman och kontrakt som används för filkomplettering utgör det aktuella API-kontraktet.

Strukturen betraktas som tillräckligt stabil för att påbörja implementation. Eventuella justeringar förväntas vara mindre och kan exempelvis avse:

- förtydliganden i dokumentation
- tillägg av nya tillåtna värden

Grundläggande struktur, endpoint och filhantering ändras endast i undantagsfall och kommuniceras i god tid.

9. Sammanfattning

- Filkomplettering är OpenAPI-baserad och stöder JSON och XML
- Innehållet styrs av ett yttre kontrakt och ett inre schema per handling
- Initialt stöds ett inre schema: `filinskickObjekttillsyn` version 1
- För detta schema används endast `filer`, inte `content`
- Utgående dokument är fortsatt XSD-baserade och versionsstyrda via URI

Vid frågor eller behov av förtydliganden, kontakta supportnemus@skogsstyrelsen.se.