

# API-flyt for Asynkrone signeringsoppdrag

Dette integrasjonsmønsteret passer for tjenesteeiere som ønsker å opprette signeringsoppdrag i signeringstjenesten som et ledd i en flyt som ikke starter med at sluttbruker befinner seg på tjenesteeiers nettsider. Signeringsseremonien gjennomføres av sluttbruker i Signeringsportalen, og tjenesteeier vil deretter asynkront kunne polle på status og hente ned det signerte dokumentet.

Dette scenariet er også utviklet med tanke på å støtte prosesser der det er behov for å innhente signaturer fra flere enn én sluttbruker på et dokument.

Relevante typer for denne delen av APIet finnes i filen `schema/xsd/portal.xsd`.



Flytskjema for det asynkrone scenariet: *skjemaet viser flyten fra tjenesteeier sender inn oppdrag, starter polling, via at sluttbruker(e) signerer oppdragene, og tjenesteeier får svar på polling og kan laste ned signert versjon. Dersom du ikke sender et oppdrag til mer enn en bruker (multiundertegner) kan du se bort i fra den første "steg 4"-seksjonen. Heltrukne linjer viser brukerflyt, mens stiplede linjer viser API-kall*

## Steg 1: Opprette signeringsoppdraget

Flyten begynner ved at tjenesteeier gjør et bak-kanal-kall mot APIene for å opprette signeringsoppdraget. Dette kallet gjøres som ett multipart-request, der den ene delen er dokumentpakken og den andre delen er metadata.

- Kallet gjøres som en `HTTP POST` mot ressursen `<rot-URL>/portal/signature-jobs`
- Dokumentpakken legges med multipart-kallet med mediatypen `application/octet-stream`. Se tidligere kapittel for mer informasjon om dokumentpakken.
- Metadataene som skal sendes med i dette kallet er definert av elementet `portal-signature-job-request`. Disse legges med multipart-kallet med mediatypen `application/xml`.

Følgende er et eksempel på metadata for et asynkront signeringsoppdrag:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<portal-signature-job-request xmlns="http://signering.posten.no/schema/v1">
  <reference>123-ABC</reference>
  <polling-queue>custom-queue</polling-queue>
</portal-signature-job-request>
```

Følgende er et eksempel på `manifest.xml` fra dokumentpakken for et signeringsoppdrag som skal signeres av fire undertegnere:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<portal-signature-job-manifest xmlns="http://signering.posten.no/schema/v1">
  <signers>
    <signer order="1">
      <personal-identification-number>12345678910</personal-identification-number>
      <signature-type>ADVANCED_ELECTRONIC_SIGNATURE</signature-type>
      <notifications>
        <!-- Override contact information to be used for notifications -->
        <email address="signer1@example.com" />
        <sms number="00000000" />
      </notifications>
    </signer>
    <signer order="2">
      <personal-identification-number>10987654321</personal-identification-number>
      <signature-type>AUTHENTICATED_ELECTRONIC_SIGNATURE</signature-type>
      <notifications>
        <email address="signer2@example.com" />
      </notifications>
    </signer>
    <signer order="2">
      <personal-identification-number>01013300001</personal-identification-number>
      <signature-type>AUTHENTICATED_ELECTRONIC_SIGNATURE</signature-type>
      <notifications-using-lookup>
        <!-- Try to send notifications in both e-mail and SMS using lookup -->
        <email/>
        <sms/>
      </notifications-using-lookup>
    </signer>
    <signer order="3">
      <personal-identification-number>02038412546</personal-identification-number>
      <signature-type>AUTHENTICATED_ELECTRONIC_SIGNATURE</signature-type>
      <notifications-using-lookup>
        <email/>
      </notifications-using-lookup>
    </signer>
  </signers>
  <sender>
    <organization-number>123456789</organization-number>
  </sender>
  <document href="document.pdf" mime="application/pdf">
    <title>Tittel</title>
    <nonsensitive-title>Sensitiv tittel</nonsensitive-title>
    <description>Melding til undertegner</description>
  </document>
  <required-authentication>4</required-authentication>
  <availability>
    <activation-time>2016-02-10T12:00:00+01:00</activation-time>
    <available-seconds>864000</available-seconds>
  </availability>
</portal-signature-job-manifest>
```

```
</availability>
<identifier-in-signed-documents>PERSONAL_IDENTIFICATION_NUMBER_AND_NAME</identi
</portal-signature-job-manifest>
```

## Undertegners kontaktinformasjon

Et signeringsoppdrags undertegner(e) kan adresseres på kontaktinformasjon (e-postadresse og/eller mobilnummer) for virksomheter som ikke har eller ønsker å bruke fødselsnummer.

Erstatt i så fall `personal-identification-number` med `identified-by-contact-information` og legg ved kontakinformasjonen i `notifications` -elementet. NB: dette kan *ikke* kombineres med `notifications-using-lookup` og er således ikke tilgjengelig for offentlige avsendere.

```
<signer>
  <identified-by-contact-information/>
  <notifications>
    <email address="email@example.com"/>
  </notifications>
</signer>
```

Les mer om adressering uten fødselsnummer i [den funksjonelle dokumentasjonen](#).

## Andre attributer

`order` -attributtet på `signer` brukes til å angi rekkefølgen på undertegnerne. I eksempelet over vil oppdraget først bli tilgjengelig for undertegnerne med `order="2"` når undertegnere med `order="1"` har signert, og for undertegneren med `order="3"` når begge de med `order="2"` har signert.

Som for synkrone oppdrag kan man også inkludere feltet `on-behalf-of` under `signer`. Det har samme semantikk for asynkrone som for synkrone oppdrag, bortsett fra at standardverdien er `OTHER` dersom avsender selv angir undertegners kontaktinformasjon. For asynkrone oppdrag på vegne av offentlige avsendere vil verdien av feltet alltid kunne utledes fra varslingsinnstillingene, og er derfor ikke nødvendig å oppgi.

Verdien av dette feltet vil også valideres opp mot varslingsinnstillingene. Har man angitt `OTHER` kan man ikke angi `notifications-using-lookup`, ettersom man ikke kan slå opp kontaktinformasjon i Kontakt- og reservasjonsregisteret om man signerer på vegne av noen andre enn seg selv. Videre vil man for offentlige virksomheter ikke kunne angi `SELF` og samtidig overstyre kontaktinformasjon; når man undertegner signeringsoppdrag på vegne av

seg selv fra avsendere i offentlig sektor *må* kontaktinformasjon til undertegner hentes fra KRR.

For private virksomheter vil man kunne velge fritt mellom `SELF` og `OTHER`, men kan aldri angi `notifications-using-lookup` ettersom disse virksomheten ikke kan benytte KRR for kontaktinformasjonsoppslag.

`availability` brukes til å kontrollere tidsrommet et dokument er tilgjengelig for mottaker(e) for signering.

Tidspunktet angitt i `activation-time` angir når jobben aktiveres, og de første undertegnerne får tilgang til dokumentet til signering.

Tiden angitt i `available-seconds` gjelder for alle undertegnere; dvs. alle undertegnere vil ha like lang tid på seg til å signere eller avvise mottatt dokument fra det blir tilgjengelig for dem. Dette tidsrommet gjelder altså *for hvert sett med undertegnere med samme `order`*. Dersom man angir f.eks. 345600 sekunder (4 dager) vil undertegnere med `order=1` få maks 4 dager fra `activation-time` til å signere. Undertegnere med `order=2` vil få tilgjengeliggjort dokumentet umiddelbart når *alle undertegnere med `order=1` har signert*, og de vil da få maks 4 nye dager fra *tidspunktet de fikk dokumentet tilgjengelig*. En jobb utløper og stopper dersom minst 1 undertegner ikke agerer innenfor sitt tidsrom når dokumentet er tilgjengelig. Dersom man utelater `availability` vil jobben aktiveres umiddelbart, og dokumentet vil være tilgjengelig i maks 2 592 000 sekunder (30 dager) for hvert sett med `order`-grupperte undertegnere. Jobber som angir større `available-seconds` enn 7 776 000 sekunder (90 dager) blir avvist av tjenesten.

`identifier-in-signed-documents` brukes for å angi hvordan undertegneren(e) skal identifiseres i de signerte dokumentene.

Tillatte verdier er `PERSONAL_IDENTIFICATION_NUMBER_AND_NAME`, `DATE_OF_BIRTH_AND_NAME` og `NAME`, men ikke alle er gyldige for alle typer signeringsoppdrag og avsendere.

Disse begrensningene er beskrevet i [den funksjonelle dokumentasjonen](#).

---

Som respons på dette kallet vil man få en respons definert av elementet `portal-signature-job-response`. Denne responsen inneholder en ID generert av signeringstjenesten. Du må lagre denne IDen i dine systemer slik at du senere kan koble resultatene du får fra polling-mekanismen til riktig oppdrag.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<portal-signature-job-response xmlns="http://signering.posten.no/schema/v1">
  <signature-job-id>1</signature-job-id>
  <cancellation-url>https://api.signering.posten.no/api/{sender-identifier}/porta
</portal-signature-job-response>
```

## Steg 2: Polling på status

Siden dette er en asynkron flyt, så må du jevnlig spørre signeringstjenesten om det har skjedd noen endringer på noen av signeringsoppdragene for din organisasjon. Dette gjør du på tvers av alle signeringsoppdrag du har opprettet, hvis ikke ville du måtte foretatt en voldsom mengde spørringer dersom du har flere aktive signeringsoppdrag i gang samtidig, hvilket du sannsynligvis har.

For å gjøre en polling, så gjør du en `HTTP GET` mot `<rot-URL>/portal/signature-jobs`. Hvis signeringsoppdraget er lagt på en spesifikk kø, så må også query-parameteret `polling_queue` settes til navnet på køen (f.eks. `<rot-URL>/portal/signature-jobs?polling_queue=custom-queue`). Du skal ikke ha med noen request-body på dette kallet.

Responsen på dette kallet vil være én av to ting:

1. 0 oppdateringer: Dersom det ikke er noen oppdateringer på tvers av alle dine aktive signeringsoppdrag vil du få en HTTP respons med statuskode `204 No Content`.
2. Minst 1 oppdatering: Dersom det er oppdateringer på dine oppdrag, så vil du få en `200 OK` med responsbody som inneholder informasjon om oppdateringen. Denne er definert av elementet `portal-signature-job-status-change-response`.

Følgende er et eksempel på en respons der en del av signeringsoppdraget har blitt fullført:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<portal-signature-job-status-change-response xmlns="http://signering.posten.no/sche
  <signature-job-id>1</signature-job-id>
  <status>IN_PROGRESS</status>
  <confirmation-url>https://api.signering.posten.no/api/{sender-identifiser}/porta
  <signatures>
    <signature>
      <status since="2017-01-23T12:51:43+01:00">SIGNED</status>
      <personal-identification-number>12345678910</personal-identification-nu
      <xades-url>https://api.signering.posten.no/api/{sender-identifiser}/port
    </signature>
    <signature>
      <status since="2017-01-23T12:00:00+01:00">WAITING</status>
      <personal-identification-number>98765432100</personal-identification-nu
    </signature>
    <pades-url>https://api.signering.posten.no/api/{sender-identifiser}/portal/s
  </signatures>
</portal-signature-job-status-change-response>
```

## Steg 3: Last ned PAdES eller XAdES

---

I forrige steg fikk du to lenker: `xades-url` og `pades-url` . Disse kan du gjøre en `HTTP GET` på for å laste ned det signerte dokumentet i de to formatene.

XAdES-filen laster du ned pr. undertegner, mens PAdES-filen lastes ned på tvers av alle undertegnere. Denne vil inneholde signeringsinformasjon for alle undertegnere som frem til nå har signert på oppdraget. I de aller fleste tilfeller er det ikke aktuelt å laste ned denne før alle undertegnerne har statusen `SIGNED` .

Se nærmere forklaring av disse to formatene i dokumentasjonen på det synkrone scenariet.

## Steg 4: Bekrefte ferdig prosessering

---

Til slutt gjør du et `HTTP POST` -kall mot `confirmation-url` for å bekrefte at du har prosessert statusoppdateringen ferdig. Dersom statusen indikerer at oppdraget er helt ferdig, så vil denne bekreftelsen også bekrefte at du er ferdig med å prosessere hele oppdraget.

Hvis [langtidslagring](#) benyttes vil dette markere oppdraget som ferdig og lagret. I motsatt fall vil oppdraget slettes i signeringsportalen.

I tillegg vil dette kallet gjøre at du ikke lenger får informasjon om denne statusoppdateringen ved polling. Se mer informasjon om det nedenfor, i avsnittet om fler-server-scenariot.

## Mer informasjon om pollingmekanismen

---

### Hvor ofte skal du polle?

Mekanikken fungerer slik at du venter en viss periode mellom hver gang du spør om oppdateringer. Oppdateringenene vil komme på en kø, og så lenge du får en ny statusoppdatering, så kan du umiddelbart etter å ha prosessert denne igjen spørre om en oppdatering. Dersom du får beskjed om at det ikke er flere oppdateringer igjen, så skal du ikke spørre om oppdateringer før det har gått en viss periode. Når du gjør denne pollingen så vil du alltid få en HTTP-header ( `X-Next-permitted-poll-time` ) som respons som forteller deg når du kan gjøre neste polling. I produksjonsmiljøet vil neste tillatte polling-tidspunkt være om 10 minutter om køen er tom, mens for testmiljøer vil det være mellom 5 og 30 sekunder.

### Hva med et fler-server-scenario:

Signeringstjenestens pollingmekanisme er laget med tanke på at det skal være enkelt å gjøre pollingen fra flere servere uten at du skal måtte synkronisere pollingen på tvers av disse. Dersom du bruker flere servere uten synkronisering så vil du komme opp i situasjoner der en

av serverene poller før neste poll-tid, selv om en annen server har fått beskjed om dette. Det er en helt OK oppførsel, du vil da få en HTTP respons med statusen `429 Too Many Requests` tilbake, som vil inneholde headeren `X-Next-permitted-poll-time`. Så lenge du etter det kallet respekterer poll-tiden for den serveren, så vil alt fungere bra.

Statusoppdateringer du henter fra køen ved polling vil forsvinne fra køen, slik at en eventuell annen server som kommer inn ikke vil få den samme statusoppdateringen. Selv om du kaller på polling-APIet på samme tid, så er det garantert at du ikke får samme oppdatering to ganger. For å håndtere at feil kan skje enten i overføringen av statusen til deres servere eller at det kan skje feil i prosesseringen på deres side, så vil en oppdatering som hentes fra køen og ikke bekreftes dukke opp igjen på køen. Pr. i dag er det satt en venteperiode på 10 minutter før en oppdatering igjen forekommer på køen. På grunn av dette er det essensielt at prosesseringsbekrefelse sendes som beskrevet i Steg 4.