

OPPSTARTSGUIDE FOR INTEGRASJON MOT BRØNNØYSUNDREGISTRENE

Mulighet for å sende meldinger til Brønnøysundregistrene direkte fra medlemsregistre

INNHOOLD

Bakgrunn	4	Vedtak	19
Innledning	5	Innhenting av informasjon om organisasjon	19
Hvordan bruke oppstartsguiden	6	Driftsmiljø	20
Personer uten teknisk kompetanse	6	Planlagte endringer i produksjonsmiljø	20
Anbefalt teknisk kompetanse	6	Kritiske feilrettinger i produksjonsmiljø	20
Relevante teknologier og begrep	7	ID-porten	21
Beskrivelse av prosessen	8	Innledning	21
Fylle ut endringsmelding	8	Offisiell dokumentasjon	21
Signere endringsmelding	8	Tilganger	21
Innsending for behandling	9	Teknisk beskrivelse	21
Mottak av applikasjonskvittering og vedtak	9	Biblioteker	21
Applikasjonskvittering	9	Prosessflyt	22
Vedtak	9	Metadata	22
Integrasjoner	10	Altinn (Formidlingstjenesten)	23
Folkeregisteret	11	Innledning	23
Innledning	11	Offisiell dokumentasjon	23
Tilganger: Konsesjon og juridiske krav	11	Tilganger	23
Kunde i InfoTorg	11	Teknisk beskrivelse	23
Avtale om tilgang til Folkeregisteret	11	Hvorfor ikke Python?	24
Abonnementsavtale om rett til bruk av informasjon	11	Bibliotek	24
Egenerklæring	11	Altinn Broker Service Toolkit (BST)	24
Teknisk beskrivelse	12	Struktur	24
Overordnet informasjon	12	Installasjon og bruk	24
Biblioteker	12	Installere	25
Innhente personalia fra Folkeregisteret	13	Konfigurasjon	25
Offisiell dokumentasjon	13	API	25
Brønnøysundregistrene	14	Driftsmiljø	26
Innledning	14	Sertifikater og autentisering	27
Tilganger	14	Teknisk beskrivelse	27
Avtaleverk	14	Virksomhetssertifikater	27
Teknisk beskrivelse	14	Eksportere X.509-sertifikat i PEM-format	28
Tilgang til tjenestene	14	Eksportere X.509-nøkkel i PEM-format	28
Offisiell dokumentasjon	14	"Og så har me juksa litt...": Eksempler	29
Biblioteker	14	Oppslag mot Folkeregisteret	29
SOAP-tjenester	15	Oppsett av klient	29
Prekontroll	15	Foreta spørring	29
Dokumenttyper	16	SDO (frivintVerifisertMelding)	30
Oversikt	16	Recipients.xml	30
Endringsmelding	17	Manifest.xml	30
Svar fra prekontroll	17	Erfaringer fra prosjektet	31
Melding (melding.xml)	17	Tips og råd for bygging av melding	31
Signerte meldinger (frivintVerifisertMelding)	18	Nyttig om testing mot testsystemet	31
Manifest (Manifest.xml)	18	Responstid per oktober 2019	31
Mottakere (Recipients.xml)	18	Avsluttende ord	32
Applikasjonskvitteringer	19		

Oktober 2019

BAKGRUNN

Brønnøysundregistrene etablerte Frivillighetsregisteret 1. desember 2008. Formålet med registeret er å forenkle samhandlingen mellom frivillige organisasjoner og offentlige myndigheter ved å samle og gjøre tilgjengelig nøkkelinformasjon om frivillige organisasjoner ett sted. Ved lanseringen av registeret ble frivillig sektor lovet en enkel måte å registrere en samlet organisasjon på. Et hovedmål med ordningen var at frivillige organisasjoners lokalledd skulle unngå dobbeltarbeid.

Som et ledd i å innfri det løftet har det over lengre tid vært jobbet med en løsning som skal sørge for at endringsmeldinger enkelt kan sendes til Brønnøysundregistrene (Frivillighetsregisteret og Enhetsregisteret) når de oppdateres i organisasjoners medlemssystemer. Prosjektet har vært organisert i to grupper. Prosjektgruppe A har bestått av [Norges Idrettsforbund og olympiske og paralympiske komité](#), og [Brønnøysundregistrene](#), mens prosjektgruppe B har bestått av [Frivillighet Norge](#), [Brønnøysundregistrene](#), og [Landsrådet for Norges barne- og ungdomsorganisasjoner](#). Prosjektgruppe A fokuserte på et spesifikt internt medlemssystem, hadde et internt utviklingsteam og ville få løsningen i bruk raskt. Prosjektgruppe B har sett på medlemssystemer generelt og hatt som mål å utvikle en løsning som kan tas i bruk av ulike systemer.

For å teste ut løsningen i praksis valgte prosjektgruppa et medlemsregister som pilot. Valget falt på [HyperSys](#), et system som er i bruk av mange organisasjoner, har en åpen utviklingsfilosofi og eies av frivillige organisasjoner. Selskapet [Unicornis](#) utvikler og drifter HyperSys, og har hatt i oppdrag å utvikle denne oppstartsguiden med bistand fra Frivillighet Norge og Brønnøysundregistrene. Hyperion, en av organisasjonene som bruker HyperSys, har vært pilotbruker og testet løsningen underveis. HyperSys brukes av mange ulike organisasjoner innen ulike felt og med størrelser fra 200 til over 30 000 medlemmer, og i siste del av prosjektet var denne bredden godt representert i prosjektet.

Det har tatt tid å få dette prosjektet i land og få på plass oppstartsguiden, som er dokumentet du nå leser. Det er mange og sammensatte grunner til det. Mangel på både økonomiske og menneskelige ressurser, mange involverte aktører, datagrunnlag av ulik kvalitet, endrede systemkrav og uforutsette juridiske komplikasjoner er noen av dem. Det er også brukt mye tid og ressurser på brukergrensesnitt. Det har vært viktig å bruke klarspråk som samtidig ivaretar juridisk dekkende begreper.

Vi skriver nå oktober 2019 og er glade for at denne delen av prosjektet med forenkling nå avsluttes. En stor takk må rettes til Kulturdepartementet som har finansiert prosjektet og prosjektgruppene som har gjort jobben med å programmere og teste løsningen.

Vi håper mange frivillige organisasjoner og utviklere av medlemssystemer vil ha nytte av jobben som er gjort og at oppstartsguiden som du nå leser vil bli flittig brukt.

Stian Slotterøy Johnsen,
Generalsekretær
Frivillighet Norge

Lars Peder Brekk,
Direktør
Brønnøysundregistrene

Andreas-Johann Østerdal Ulvestad,
Daglig leder
Unicornis

INNLEDNING

Dokumentet du nå leser har to hovedmål:

1. Å gi dere som organisasjon oversikt over hva dere må vurdere, og hva dere må gjøre, for å ta i bruk integrasjonsløsningen i deres medlemssystem
2. Å gi en beskrivelse av tekniske krav og prosesser, som dere kan gi videre til de som utvikler deres system

Med denne oppstartsguiden i hånd, kan dere som organisasjon ta et godt informert valg om hvorvidt integrering mot Brønnøysundregistrene er noe dere ønsker å gjøre mulig i deres systemer. Ved å gjøre det, åpner dere for at meldinger enkelt kan opprettes og sendes fra deres organisasjons medlemssystem til Frivillighetsregisteret og Enhetsregisteret.

Dokumentet gir også noe informasjon til utviklerne av deres systemer om hvordan de kan gå frem for å tilby integrasjon. Den tekniske delen av denne oppstartsguiden er derfor i all hovedsak beregnet på personer med utviklingserfaring. De delene som er beregnet på mennesker med tekniske forkunnskaper er markert som "Teknisk beskrivelse".

Det er viktig at dere som organisasjon selv gjør en vurdering av hvorvidt dere ønsker å tilby løsningen og at dere har en god intern prosess på dette. Integrasjonsløsningen kan gi en enklere hverdag for lokale tillitsvalgte, som slipper å oppdatere opplysninger om organisasjonen to steder. Den vil også gjøre det lettere for flere lokallag å få organisasjonsnumre og motta grasrotmidler, og løsningen vil være et godt servicetilbud til lokalleddene. Samtidig vil den medføre noe ekstra oppfølging fra sentralledet. Når flere lokallag får organisasjonsnummer betyr det også flere personer med juridisk ansvar.

Integrasjon mot Brønnøysundregistrene medfører avtaler og integrasjon med flere leverandører, som Difi/ID-Porten, Altinn og Folkeregisteret. Leverandører som iverksetter en integrasjon, må forholde seg til de enkelte motpartenes krav til integrasjon og bære etablerings- og brukskostnader for hver enkelt av disse integrasjonene selv.

Medlemssystemene som er i bruk ute i frivilligheten, er svært ulike og er utviklet i ulike tidsaldre med ulike teknologiske valg. Oppstartsguiden er ferdigstilt i oktober 2019 og baserer seg på de tekniske kravene som er gjeldende på det tidspunktet.

Det kan komme endringer i de tekniske kravene. Framfor å skrive en komplett kravspesifikasjon i oppstartsguiden, har vi derfor vist hvor dere til enhver tid kan hente oppdatert informasjon og krav. Det vil alltid være lenkene til dokumentasjonen for de respektive komponentene/systemene som har den sist oppdaterte informasjonen. Oppstartsguiden viser dere hvordan dere skal gå frem for å komme i gang, og gjør det mulig for dere å orientere dere om hvilke komponenter/systemer integrasjonsløsningen består av.

Deler av løsningen er basert på åpen kildekode, og det er opp til hver enkelt utvikler om og hvordan de ønsker å ta koden i bruk. Noen kan for eksempel ønske å skrive sin egen kode i et annet språk og kun benytte kildekoden som referanse.

Eksempelkode er tilgjengelig på Github:

- pydsf: <https://github.com/unicornis/pydsf>
- pybreg: <https://github.com/unicornis/pybreg>
- Altinn BST: https://github.com/unicornis/altinn_broker_service_toolkit

Vi gleder oss til å se flere medlemssystemer åpne for integrering og til å se løsningen utvikle seg.

HVORDAN BRUKE OPPSTARTSGUIDEN

Kapitlene som omhandler de ulike systemene, er delt opp slik at personer uten utviklingserfaring kan lese og forstå de innledende delene i hvert delkapittel som ikke er markert som teknisk beskrivelse. Vi anbefaler likevel at dere har noe IT-kompetanse når dere leser dokumentet. Vi bruker følgende begreper på følgende måte:

- «Organisasjon» brukes om enheten som er gjenstand for endringsmeldinger (for eksempel et lokallag).
- «Foretak» og «utvikler» brukes om leverandøren som skal integrere mot Brønnøysundregistrene.

Dette dokumentet forklarer de overordnede konseptene knyttet til løsningen og hva dere som organisasjon og foretak må tenke på. For å få tilgang til kildekode, teknisk dokumentasjon og testsystemer, må dere inngå avtale med de forskjellige partene som drifter og leverer de forskjellige komponentene av integrasjonsløsningen:

- Brønnøysundregistrene (Enhetsregisteret og Frivillighetsregisteret)
- Skatteetaten (Folkeregisteret)
- Difi (ID-Porten)
- Altinn (formidlingstjenesten)

PERSONER UTEN TEKNISK KOMPETANSE

Det anbefales at personer som har ansvar for prosjektet, særlig leser seksjonene om tilganger for de ulike systemene. Her står viktig informasjon om inngåelse av avtaler og juridiske krav. Den ikke-tekniske delen av kapittelet om sertifikater bør også leses, men dersom du ikke forstår det som står der, er sjansen stor for at de som utvikler systemet, gjør det. For en overordnet forklaring av hvordan integrasjonen fungerer, anbefales kapittelet “Beskrivelse av prosessen”.

ANBEFALT TEKNISK KOMPETANSE

For de tekniske delene av dette dokumentet, anbefaler vi at leseren har utviklingserfaring.

Forfatterne jobber primært med Python, og ved prosjektets start var det tiltenkt at alle moduler og komponenter som blir lansert sammen med dette dokumentet, skulle være i Python. Visse tekniske begrensninger gjorde at integrasjonslaget for Altinn måtte gjøres i Java (detaljer finnes i det relevante kapittelet).

Det er ikke påkrevd, men anses som en fordel å ha kunnskap og kompetanse innen objektorientert programmering. De fleste eksempler som gis er i Python, men burde være leselige for personer med kompetanse i lignende språk, som for eksempel Java eller C#.

Vi legger ingen føringer eller beskrivelser for hvordan en intern dokument-, signerings- eller applikasjonsstruktur skal være for å bruke disse bibliotekene. Alle python-baserte bibliotek er agnostiske når det gjelder hvilke rammeverk de integreres i. Unntak fra dette beskrives i relevante kapitler.

Relevante teknologier og begrep

Under følger en oversikt over teknologier og begrep som er sentrale i utformingen av en infrastruktur som skal støtte innsending av endringsmeldinger til offentlige registre.

Begrep/Teknologi	Bruksområde
XML	Dataformat for informasjonen som skal utveksles
SOAP	Kommunikasjonsprotokoll mellom applikasjon og offentlige tjenester
SAML	Kommunikasjonsprotokoll for brukerautentisering og verifisering



BESKRIVELSE AV PROSESSEN

Dette kapitlet er et generelt sammendrag for å beskrive prosessen ved innsending av melding til Brønnøysundregistrene. Beskrivelsen inneholder noen tekniske termer, men selv om du ikke forstår alle detaljene, kan det allikevel være nyttig å lese for å få et bedre overblikk. Dette er ikke fullstendig eller detaljert, da alle detaljer blir tilgjengelig i en implementasjonsguide hos Brønnøysundregistrene. Av sikkerhetshensyn må det inngås en intensjonsavtale med Brønnøysundregistrene for å få tilgang til implementasjonsguiden. Formålet med intensjonsavtalen er å regulere bruk av informasjonen som dere vil få tilgang til i prosessen med å ta bruk integrasjonsløsningen.

FYLLE UT ENDRINGSMELDING

Det første steget i prosessen er å samle inn informasjonen organisasjonene ønsker å melde en endring om. Dette er en naturlig del av mange organisatoriske prosesser, slik som valg av nytt styre, vedtektsendringer m.m.

I dette steget velger organisasjonene informasjonen som skal endres i registeret, samt hvilke handlinger som skal gjøres. Selv om vi generelt sett kaller det en endringsmelding, kan det også være en nyregistrering eller en sletting av en organisasjon. Løsningen støtter følgende typer meldinger:

- Nyregistrering til Enhets- og Frivillighetsregisteret i samme melding.
- Endring i Enhetsregisteret og samtidig nyregistrering i Frivillighetsregisteret.
- Endring i Enhets- eller Frivillighetsregisteret (krever at organisasjonen er registrert i Frivillighetsregisteret).
- Endring i Enhetsregisteret og samtidig sletting fra Frivillighetsregisteret.
- Sletting fra Enhets- og Frivillighetsregisteret i samme melding.

Som det fremgår av listen, er det et krav om å være registrert, eller registrere seg i Frivillighetsregisteret for å kunne bruke løsningen. Det er med andre ord ikke en løsning for de organisasjonene som kun ønsker å forholde seg til Enhetsregisteret.

Det kan være krav om vedlegg til en endringsmelding. Dette avhenger av hvilken informasjon som meldes. Alle vedlegg skal sendes sammen med endringsmeldingen og signaturene slik at dette danner en pakke.

For at endringsmeldingen skal være korrekt utfylt, må det hentes inn informasjon om organisasjonen og dens rollehavende personer, noe som gjøres ved å kommunisere med de relevante datatjenestene hos henholdsvis Brønnøysundregistrene og Folkeregisteret.

SIGNERE ENDRINGSMELDING

Ved utfylling av en endringsmelding spesifiseres også personen som skal signere meldingen. Denne personen omtales som signatar. Hvem dette er avhenger av organisasjonens struktur og egne valg, men i mange tilfeller vil dette for eksempel være styreleder, styremedlem, eller daglig leder.

For at en signatur skal kunne innhentes, forutsetter det at signatøren er verifisert i Folkeregisteret. Så snart dette er gjort, kan dokumentet signeres. Signaturer innhentes ved å sende brukeren til ID-porten, innloggingstjenesten drevet av Direktoratet for forvaltning og IKT (Difi), som også ofte brukes ved innlogging i andre tjenester, for eksempel Altinn.

En signatur defineres som en godkjent og verifisert innlogging mot ID-porten, utført med en nivå 3-verifisering eller høyere (MinID, BankID, BankID på Mobil og tilsvarende). Når en slik innlogging er godkjent, innhentes det et teknisk dokument (kalt "SAML Assertion") fra ID-porten, som brukes for å verifisere identiteten på brukeren som gjennomfører signeringen.

En SAML Assertion er et XML-dokument som beskriver informasjon om tjenesteyter (din tjeneste), identifiseringstjeneste (ID-porten/Difi) og om brukeren som gjennomførte innloggingen. Dette dokumentet er ansett som signaturen og skal inkluderes per signatar i sluttdokumentet.

INNSENDING FOR BEHANDLING

Når endringsmeldingen er klar og signert av alle parter, skal den sendes inn for behandling hos Brønnøysundregistrene. Dette gjøres ved å generere en zip-fil med alt innholdet beskrevet i stegene over, som deretter oversendes via Formidlingstjenesten, en sentral lagringstjeneste som ligger hos Altinn.

Formidlingstjenesten er en slags Dropbox og fungerer som et mellomledd mellom aktører i det offentlige, og brukes som kanal for utveksling av dokumenter, vedtak og informasjon mellom privatpersoner, organisasjoner, selskaper og offentlige aktører.

MOTTAK AV APPLIKASJONSKVITTERING OG VEDTAK

Formidlingstjenesten brukes også som mellomledd for mottak av applikasjonskvittering og vedtak. Deres tjeneste må jevnlig sjekke om det er sendt nye filer til tjenesten, laste ned eventuelle nye filer og behandle dem på korrekt måte.

APPLIKASJONSKVITTERING

Når innsending av en endringsmelding er fullført, blir filen liggende til mottakeren (i dette tilfellet Brønnøysundregistrene sine systemer) har hentet den ned. Så snart dette er gjort, får mottaker en applikasjonskvittering som enten forteller at filen er avvist på teknisk grunnlag eller er videresendt til behandling.

Mottaket hos Brønnøysundregistrene er adskilt fra saksbehandlingssystemet, noe som innebærer at filen skal kontrolleres i flere omganger. Brukerne kan derfor oppleve å få filen avvist i neste instans selv om de først fikk en applikasjonskvittering som bekreftet at filen er sendt til behandling. I slike tilfeller vil brukerne motta to applikasjonskvitteringer. Applikasjonskvitteringene vil som hovedregel sendes med få sekunders mellomrom.

Applikasjonskvitteringer genereres automatisk etter maskinelle kontroller.

VEDTAK

Når endringsmeldingen er ferdig saksbehandlet, vil Enhetsregisteret og/eller Frivillighetsregisteret utstede et vedtak. Vedtaket kommer innpakket på samme måte som endringsmeldingen var pakket inn, sammen med en serie XML-dokumenter og vedlegg som er relevante. Denne pakken må åpnes og leses maskinelt, og vedlegg må lagres og gjøres tilgjengelig for de som har utformet endringsmeldingen. Det må også sendes varsel om at vedtak er mottatt og tilgjengelig. Når dette er gjort, anses prosessen som fullført. Ønsker brukerne å gjøre nye endringer, skal en ny endringsmelding opprettes og prosessen startes på nytt.

INTEGRASJONER

I denne delen går vi igjennom de ulike systemene man må koble seg mot for å få en fungerende løsning.

Folkeregisteret

- Verifisering av personer/fødselsnummer

Brønnøysundregistrene

- Innhenting av informasjon lagret om foreningen
- Prekontroll av endringsmelding

ID-porten

- Verifisering for signering av endringsmelding

Altinn (Formidlingstjenesten)

- Innsending av endringsmelding
- Mottak av applikasjonskvittering og vedtak

FOLKEREGETERET

INNLEDNING

Folkeregisteret er den plattformen for innhenting og verifisering av personalia som integrasjonsløsningen benytter seg av. Det er eid av Skatteetaten og drives av EVRY gjennom InfoTorg-plattformen.

Brønnøysundregistrene krever at alle personer med verv som meldes inn eller som skal signere endringsmeldinger, har verifisert fødselsnummer fra Folkeregisteret. Verifiseringen gjøres ved at organisasjonene sender inn informasjon om personen de ønsker å verifisere, og deretter får eventuelt treff og informasjon om personen hos Folkeregisteret.

I tillegg til verifisert fødselsnummer, krever Brønnøysundregistrene at organisasjonen sjekker status på person (f.eks. bosatt/utvandret/død), samt folkeregistrert adresse. Detaljerte krav står i implementasjonsguiden til Brønnøysundregistrene.

TILGANGER: KONSESJON OG JURIDISKE KRAV

For å få tilgang til løsningen er organisasjonen nødt til å få konsesjon fra Skatteetaten og bli kunde i InfoTorg.

Kunde i InfoTorg

Organisasjonene kan registrere seg som kunde i InfoTorg direkte via <https://infotorg.no>. Her kreves det at de opplyser om foretaket, med organisasjonsnummer og lignende nøkkelinformasjon.

Her fyller også organisasjonene ut søknad om tilgang til Folkeregisteret via deres integrasjonsløsning. Det vil ta rundt tre arbeidsdager å få svar på denne søknaden.

Avtale om tilgang til Folkeregisteret

Avtalen om tilgang til Folkeregisteret er et dokument som vil bli sendt av Evry og som må fylles ut og returneres. I dette dokumentet inngår en del tekniske opplysninger, som navn på systemet som skal ha tilgang, IP-adresser forespørsler vil komme fra og lignende.

Abonnementsavtale om rett til bruk av informasjon

Abonnementsavtalen er et dokument som vil bli sendt av Evry og som må fylles ut og returneres. Her må organisasjonen deklare at de har rett på tilgang til informasjonen i Folkeregisteret, gitt av konsesjonen fra Skatteetaten.

Egenerklæring

Det kreves også en egenerklæring for den tekniske plattformen. Dette dokumentet vil bli tilsendt fra Evry. Dokumentet inneholder en liste over punkter som må møtes for at løsningen skal være godkjent. Det er både tekniske og ikke-tekniske krav.

TEKNISK BESKRIVELSE

Overordnet informasjon

For å kommunisere med Folkeregisteret, brukes SOAP-protokollen¹. Kort fortalt er SOAP en serie HTTP POST-forespørsler med XML-payload i request og response. Dette kan i teorien gjøres manuelt med HTTP, men vi anbefaler på det sterkeste å bruke et ferdiglaget bibliotek for det språket dere jobber med, eller å bruke vårt bibliotek dersom dere bruker Python.

Folkeregisteret er en forholdsvis simpel SOAP-tjeneste og krever ikke avansert konfigurasjon for å oppnå forbindelse. Autentisering gjøres via brukernavn og passord i payload (se kodeeksempler), samt filtrering på IP fra EVRY sin side.

Utover dette anbefaler vi på det sterkeste at utviklere leser EVRYs offisielle dokumentasjon (lenket i slutten av kapittelet), da denne er detaljert og gir en god oversikt, samt flere eksempler både på kode og request/response-struktur i løsningen.

Biblioteker

Under følger en oversikt over biblioteker som kan brukes for å kommunisere med Folkeregisteret. Dette er på ingen måter en komplett liste, og den kan bli utvidet og forbedret over tid.

Navn	Språk	Type	Kommentar
pydsf	Python	DSF	Utformet av Unicornis og laget for dette prosjektet
suds	Python	SOAP	Eldre, men stabilt SOAP-bibliotek. Oppdateres veldig sjeldent.
CXF	Java	SOAP	Avansert og godt støttet SOAP-bibliotek utviklet av Apache.
zeep	Python	SOAP	Nyere og mer funksjonsrikt bibliotek. Utvikles aktivt, anbefales dersom man ønsker å lage egen løsning i Python.

¹ <https://www.w3.org/TR/soap/>

Innhente personalia fra Folkeregisteret

Folkeregisteret har mange endepunkter for innhenting av mange typer informasjon. Hvilke av disse endepunktene dere har tilgang til, og hvilke begrensninger som gjelder for hvert endepunkt avhenger av konsesjonen som er gitt. For innhenting av personalia anbefaler vi å bruke funksjonen **hentDetaljer**, da denne gir et mest helhetlig bilde av en person.

For de fleste konsesjoner, vil det være krav om å sende inn følgende informasjon når organisasjonene gjør oppslag i tjenesten:

- Fødselsnummer (fødselsdato og personnummer, 11 siffer), eller D-nummer (11 siffer)
- Fornavn
- Etternavn

Utover dette kan organisasjonene velge selv hvilken informasjon de ønsker å ta med. Fullstendig liste over alternativer ligger i den offisielle dokumentasjonen.

I søk på navn er det tillatt et visst avvik når organisasjonene foretar et søk, én til to bokstaver er akseptabelt (eksempelvis vil Tommas eller Thomas være gyldig for en person ved navn Tomas), men avvik utover dette vil ikke gi treff. Det støttes å søke på mellomnavn, men det er tilstrekkelig med første fornavn samt etternavn i kombinasjon. Mange brukere av forskjellige medlemsregistre opererer med flere definisjoner av navn, og noen tillater å sette fritt mellomnavn. Vi anbefaler å bruke den informasjonen dere er mest sikker på at stemmer overens med offentlige registre, altså første fornavn og ett etternavn.

For mer informasjon om hvilken informasjon som finnes i Folkeregisteret og hva de betyr, se den offisielle dokumentasjonen.

Offisiell dokumentasjon

<https://qa.infotorg.no/test/cms/site/0/page?id=77>

BRØNNØYSUNDREGISTRENE

INNLEDNING

Brønnøysundregistrene er behandlende enhet i prosessen for behandling av endringsmeldinger.

TILGANGER

Avtaleverk

Hvis dere, etter å ha satt dere inn i dette dokumentet, ønsker å gå videre med integrering mot Brønnøysundregistrene, må dere signere en intensjonsavtale for å få tilgang til mer dokumentasjon om prosesser, krav og tekniske implementasjonsdetaljer.

Før leverandører kan kommunisere med Brønnøysundregistrene sine datasystem, må det i tillegg inngås en avtale mellom leverandøren og Brønnøysundregistrene. Avtalen spesifiserer krav til medlemssystemet, krav til underliggende datasystemer og krav til leverandøren i seg selv. I tillegg til den formelle avtalen, vil leverandøren få innloggingsinformasjon utlevert fra Brønnøysundregistrene, samt at Brønnøysundregistrene må åpne for tilgang fra leverandørens systemer.

Disse avtalene fås ved å kontakte frivint@brreg.no.

TEKNISK BESKRIVELSE

Tilgang til tjenestene

Alle SOAP-tjenestene har brukernavn og passord og baserer seg på IP-whitelisting. For at organisasjonene skal få korrekte tilganger, må leverandøren sørge for at korrekte IP-er er sendt til Brønnøysundregistrene for alle miljøer organisasjonene ønsker å samhandle i. Brukernavn og passord til hvert miljø utleveres separat.

Offisiell dokumentasjon

Det finnes en stor mengde informasjon om prosesser, krav og tekniske implementasjonsdetaljer i [Brønnøysundregistrenes Implementasjonsguide \(krever tilgang\)](#).

Biblioteker

Navn	Språk	Type	Kommentar
pybrreg	Python	Brreg	Bibliotek for henting av informasjon på organisasjonen, prekontroll, innlesning og produksjon av dokumenter som sendes/mottas via Altinn.
suds	Python	SOAP	Eldre, men stabilt SOAP-bibliotek. Oppdateres veldig sjeldent.
zeep	Python	SOAP	Nyere SOAP-bibliotek
CXF	Java	SOAP	Stabilt og godt støttet
PyXB	Python	XML	Bibliotek for lesing og opprettelse av XML-dokumenter fra modeller. Brukes av pybrreg for disse funksjonene.
lxml	Python	XML	Bibliotek for behandling av XML

SOAP-tjenester

Under følger en oversikt over SOAP-tjenestene som brukes i samhandling med Brønnøysundregistrene.

Tjeneste	Bruksområde
Prekontroll	Verifisering av integrasjonERFV-melding før signatur og pakking
Frivillighetsregisteret	Informasjon om frivillige organisasjoner
Enhetsregisteret, basisdata	Basisdata om organisasjonen
Enhetsregisteret, prokura og signatur	Signaturkrav for organisasjonen
Enhetsregisteret, roller	Rolleoversikt for organisasjonen
Enhetsregisteret, registerutskrift	Registerutskrift for organisasjonen

Prekontroll

For å sikre at meldingene er av en slik kvalitet at de kan saksbehandles, tilbyr Brønnøysundregistrene en tjeneste som kalles prekontroll. Dette er en webservice som kontrollerer både format på informasjonen og forretningslogikk i meldingen før den sendes inn. Dette for å unngå at meldingen skal stoppes maskinelt på grunn av feil eller mangler i meldingen. Pybrreg har logikk for å sende inn melding til prekontroll.

Dokumenttyper

Oversikt

Her følger en oversikt over alle XML-dokumenter som brukes i prosessen. Noen typer dokumenter skal både sendes og mottas. For disse finnes det ferdig kode i `pybrreg` som kan brukes. For dokumentene som kun skal leses (gjelder primært Applikasjonskvittering og Vedtak) er det kun laget snarveier for å lese inn dokumentet til et objekt som kan brukes.

Dokumenttype	Dokumentasjon hos BR	Modul/klasse i pybrreg
<u>Endringsmelding</u>	<u>Confluence</u>	<code>brreg_report_changes.shortcuts</code>
<u>Svar fra prekontroll</u>	<u>Confluence</u>	<code>soap.service.precontrol</code>
<u>Melding</u>	<u>Confluence</u>	<code>xml.generated.melding</code> <code>models.new_inquiry.BrregNewInquiry</code>
<u>Signerte meldinger</u>	<u>Confluence</u>	<code>xml.frivintVerifisertMelding</code> <code>models.sdo.BrregSDO</code>
<u>Manifest</u>	<u>Confluence</u>	<code>xml.manifest</code> <code>models.manifest.BrregManifest</code>
<u>Mottakere</u>	<u>Confluence</u>	<code>xml.recipients</code> <code>models.recipients.Recipients</code>
<u>Applikasjonskvittering</u>	<u>Confluence</u>	<code>models.application_receipt.BrregApplicationReceipt</code>
<u>Vedtak</u>	<u>Confluence</u>	<code>models.decree.BrregDecree</code>

Endringsmelding

Endringsmeldingen bygges basert på hvilken informasjon organisasjonene ønsker å melde inn til Brønnøysundregistrene (Enhetsregisteret og/eller Frivillighetsregisteret). Endringsmelding er fellesbetegnelse på alle meldingstyper (endring av opplysninger, nyregistrering og sletting).

I endringsmeldingen ligger følgende informasjon:

- Informasjonsfelter til ett eller begge registre (adresse, styre, kontonummer osv.)
- Metainformasjon om meldingen:
 - Type melding: Nyregistrering, endring eller sletting.
 - Organisasjonsnummer eller midlertidig organisasjonsnummer, dersom det eksisterer
 - Om meldingen skal behandles i Enhetsregisteret og/eller Frivillighetsregisteret
 - Organisasjonsform
 - Meldings- og system-ID
- Liste med vedleggs-koder på vedlegg som vil bli inkludert ved innsending
- Liste med personnummer på de som skal signere meldinga

Endringsmeldingen bygges i pybrreg og konverteres til XML. Denne XML-fila sendes slik den er til prekontrollen, men når meldingen er signert, vil den inkluderes som et hashet felt i XML-filen `frivintVerifisertMelding` før innsending.

Svar fra prekontroll

Svar fra prekontrollen kommer i form av en XML som dekodes i pybrreg. Svaret inneholder en status og eventuelt en liste over sperrer (feilmeldinger) i tilfeller hvor det returneres en XML med status "Sperre".

Eventuelle sperrer skal vises til sluttbruker slik at denne kan rette opp feilene og gjennomføre prekontrollen på nytt.

En komplett oversikt over mulige feilkoder fra prekontrollen finnes i [implementasjonsguiden \(krever tilgang\)](#).

Melding (melding.xml)

Melding er hoveddokumentet i en forsendelse, og inneholder all informasjon om parter i saken som skal behandles, selve meldingen som skal kommuniseres (pakket inn i SDO) samt nødvendig referansedata.

For å bygge en melding, trenger du følgende data:

- Avsenders organisasjonsnummer
- Mottakers organisasjonsnummer
- UUID for meldingen (UUID = universally unique identifier)
- SDO i XML-format (SDO = Signed Data Object)
- Vedleggsliste (Filnavn og vedleggstype)

Når det mottas svar fra Brønnøysundregistrene kommer også disse i form av en slik XML-fil. For utgående meldinger vil innholdet i alle tilfeller være signert, for innkommende kan innholdet være en applikasjonskvittering eller et vedtak.

Signerte meldinger (frivintVerifisertMelding)

En signert melding, heretter SDO, inneholder følgende:

- En hashet versjon av meldingen som skal behandles
- En eller flere signaturer som vedkjenner seg dokumentet, også hashet

For å produsere gyldige hasher av verdiene for dette dokumentet kreves det en shared secret som sendes ut av Brreg. Signaturene er assertions som er returnert fra ID-porten.

Alle hashede verdier i dokumentet skal representeres i base64. I eksemplene (gitt i eget kapittel) håndteres dette automatisk av SDO-klassen, men dersom utvikler ønsker å beregne de hashede verdiene manuelt, er oppskriftene listet under. Mer informasjon om disse verdiene finnes i dokumentasjonen lenket i starten av kapitlet.

- 
- **SignedObject = base64(integrasjonERFV)**
 - **SignedObjectHash = base64(sha256(SignedObject))**
 - **Authentication = base64(saml:Assertion)**
 - **RecordSeal = base64(sha512(SignedObjectHash, Authentication, secret))**

NB! Når utviklere hasher disse verdiene, er det ekstremt viktig å sørge for at de har en konsekvent implementasjon når de gjør dokument -> base64 -> dokument. Dersom de ikke sørger for å gjøre eksempelvis escaping av XML-tegn i verdier inne i dokumentene korrekt, kan dette resultere i at kanonikaliseringen av dokumentet blir feil, og at valideringen av dokumentet fra Brønnøysundregistrene sin side feiler, da dokumentet ikke kan gjenopprettes til å være likt det opprinnelige som ble innsendt.

Manifest (Manifest.xml)

Manifestet beskriver innholdet i den fullstendige pakken, og er en liste over filene pakken inneholder. Dokumentet er egentlig definert av Altinn og Formidlingstjenesten, men vi beskriver det her, da det ligger pakket inn under pybrreg og skal i samme forsendelse.

Manifestet inneholder følgende:

- Tjenestekoder for forsendelser (disse er statisk satt og man trenger ikke bry seg med de)
- Senders referanse, samme UUID som Melding:id
- Avsender
- Liste over filer i forsendelsen (inkludert selve meldingen, med filnavn og sjekksum)
- En valgfri liste med alternativer (nøkkel og verdi for hvert alternativ)

Mottakere (Recipients.xml)

Mottakerfilen beskriver mottakerne av pakken. Dette vil i de aller fleste tilfeller bare være én mottaker, Brønnøysundregistrene. Hver mottaker identifiseres med organisasjonsnummer eller fødselsnummer.

Applikasjonskvitteringer

Etter innsending av en forsendelse kan organisasjonene motta én eller flere applikasjonskvitteringer. Denne kvitteringen beskriver nåværende status på innsendingen. Det gjøres en foreløpig analyse av forsendelsen umiddelbart. Dersom det er en teknisk feil med filen vil organisasjonene få rask tilbakemelding om dette, og meldingen vil bli avvist. Dersom organisasjonene mottar en applikasjonskvittering med statuskode 201 og melding “Mottatt til behandling”, er filen mulig å behandle videre, og det vil bli kjørt en rekke maskinelle kontroller før meldingen går til saksbehandler.

Dersom organisasjonene ønsker å ta vare på disse applikasjonskvitteringene i sitt system, kan det være en fordel å sørge for at de leser dem inn stigende etter sendingsdato (hver forsendelse har en sendingsdato, uavhengig av innhold), slik at de sørger for hele tiden å oppdatere status på dokumentet til den korrekte verdien.

Meldinger som sendes inn gjennom Formidlingstjenesten, går gjennom to steg. Først blir den mottatt av postmottaket til Brønnøysundregistrene, og deretter blir den prosessert videre i Brønnøysundregistrenes interne systemer (referert til som “INTEF”).

En melding kan bli akseptert av postmottaket, men avvist av INTEF. I slike tilfeller vil organisasjonene motta to applikasjonskvitteringer: én med kode 201, som bekrefter at meldingen er akseptert av postmottaket, og én som har kode mellom 300 og 399, som inneholder feilmeldingen fra INTEF. Kvittering nummer to vil som regel mottas i løpet av få minutter hvis meldingen blir avvist av INTEF.

Vedtak

Etter at saksbehandlingen er gjennomført, vil organisasjonene motta en pakke som inneholder et vedtak og vedlegg til vedtaket. For å finne overordnet status på vedtaket, må systemet lese XML-elementet, slik at organisasjonene automatisk kan oppdatere eventuelle verdier de har internt i sitt system, og sende passende varslinger til de relevante partene. Vedleggene for vedtakene kommer i PDF og for å få informasjon om innholdet i filene må dette sjekkes mot vedleggslisten i meldingsdokumentet.

Innhenting av informasjon om organisasjon

For å kunne hjelpe organisasjonen til å holde Brønnøysundregistrene og medlemssystemet oppdatert med korrekt informasjon, er det viktig at den informasjonen som er offentlig registrert på organisasjonen, blir hentet inn i medlemssystemet. Et fullstendig informasjonsbilde av en organisasjon oppnås ved å samle inn data fra to registre i Brønnøysund: Enhetsregisteret og Frivillighetsregisteret. Dette gjøres ved hjelp av et sett med webservicer som aktører som skal benytte integrasjonsløsningen for frivillig sektor, får tilgang til hos Brønnøysundregistrene. Under følger en oversikt over hvordan utviklere kan hente inn denne informasjonen via pybrreg sine klienter (pybrreg.soap.client.<klient>).

Klient	Funksjon	Input	Output
UnitRegistryServiceClient	get_base_dat_for_org	Orgnr	Basisdata (dict)
UnitRegistryServiceClient	get_contact_data_for_org	Orgnr	Kontaktinformasjon (dict)
UnitRegistryServiceClient	get_roles_for_org	Orgnr	Rolleinformasjon (models.roles.RoleData)
UnitRegistryServiceClient	verify_procuration_signature	Orgnr	Prokura signaturinfo (dict)
VolunteerRegistryServiceClient	get_volunteer_data	Orgnr	Data fra frivillighetsregisteret (dict)

Driftsmiljø

Brønnøysundregistrene sitt driftsmiljø er skilt i to områder; test og produksjon. Testmiljøet kan være litt ustabil i forbindelse med utrulling av nye versjoner, mens produksjonsmiljøet har god varsling av nedetid.

Brønnøysundregistrene og Altinn varsler planlagt nedetid, og alle integratorer må være på varslingsliste. Kontakt frivint@brreg.no for informasjon om dette.

Planlagte endringer i produksjonsmiljø

Brønnøysundregistrene har, jamfør [avtalen](#), tatt på seg en plikt for varsling av planlagte endringer til alle integratorer. Endringer varsles normalt sett med to måneders varsel.

Kritiske feilrettinger i produksjonsmiljø

Ved kritiske systemoppdateringer kan vedlikehold skje på under to måneders varsel, men aldri uten å varsle på forhånd.

ID-PORTEN

INNLEDNING

ID-porten er en fellestjeneste for autentisering mot offentlige tjenester, som drives av Direktoratet for forvaltning og IKT (Difi).

OFFISIELL DOKUMENTASJON

<https://samarbeid.difi.no/id-porten/dokumentasjon> (Krever innlogging)

https://difi.github.io/idporten-integrasjonsguide//93_hurtigguide.html (Hurtigguide, Difi@github)

TILGANGER

Dokumentasjon, driftsinformasjon og implementasjonsguider for ID-porten ligger samlet i en portal hos Difi, <https://samarbeid.difi.no/>. Gjennom denne portalen kan organisasjonene registrere en bruker, godkjenne vilkår for plattformen og laste ned nødvendige filer for test- og produksjonsmiljøer. Portalen bruker engangsenker på e-post for innlogging, så det er ikke nødvendig å registrere seg med passord.

TEKNISK BESKRIVELSE

Biblioteker

Å integrere SAML-støtte i egen applikasjon kan være en veldig komplisert jobb dersom utvikler velger å ikke bruke et ferdiglaget bibliotek. Det er også veldig tett knyttet mot det applikasjonsrammeverket utvikler bruker. Å lage et generelt python-bibliotek som støtter dette, er ikke verdt arbeidet, da flere sentrale deler av en SAML-arbeidsflyt omhandler redirects og sesjonshåndtering. Dette håndteres best gjennom ferdige rammeverk. Vi anbefaler på det sterkeste å ta i bruk en eksisterende løsning med støtte for det rammeverket utvikler arbeider med når dette skal gjøres, slik at dere sparer dere for unødig arbeid. Her vil det ikke bli gitt spesifikke eksempler. Vi ber utviklerne om å se dokumentasjonen for det valgte biblioteket, som burde inneholde en innføring i hvordan de setter opp SAML med eget rammeverk. Vi har valgt å bruke python3-saml.

Navn	Lenke	Språk	Kommentar
python3-saml	https://github.com/onelogin/python3-saml	Python	Støtter både Flask og Django, samt både python 2.7 og 3

Prosessflyt

Tjenesten bruker protokollen SAML 2.0. Vi bruker tjenesten her for å hente inn autentiseringsinformasjon om en person, som betegnes som en signatur. Når en slik innlogging er gjennomført, mottar organisasjonene en SAMLResponse, som er et kryptert XML-dokument som vi på vår side dekrypterer og verifiserer.

En innlogging foregår på følgende måte:

- Bruker velger å bli tatt til ID-porten fra vår tjeneste
- En lokal SAML-tjeneste genererer en URL for innlogging
- Brukeren sendes til nevnte URL og logger inn med ønsket metode

Dersom autentiseringen er fullført, returneres brukeren tilbake til vår tjeneste, med en SAMLResponse. Denne responsen må dekodes og lagres, og teller som en signatur for dokumentet (se [frivintVerifisertMelding](#)).

[Eksempel på dekryptert SAML Assertion \(krever tilgang\)](#)

Metadata

En sentral del av en SAML-integrasjon mot noen plattform er utveksling av metadata mellom SP (Service Provider, du som leser dokumentet) og IdP (Identity Provider, ID-porten/Difi). Metadata-filene er XML-filer som genereres hos hver part.

For SP sin del inneholder metadata noen viktige punkter:

- EntityID: En unik URI som beskriver parten. Denne er vanskelig å endre og burde være en generell representasjon av SP.
- Sertifikatinformasjon for kryptering og signering av SAML-requests
- URL for validering av SAMLResponse, peker på et endepunkt i din applikasjon
- URL for utlogging, peker på et endepunkt i din applikasjon
- Diverse kontaktinformasjon om organisasjonen

PS: Feltet NameIDFormat har flere mulige verdier, ID-porten støtter 'transient' og 'persistent', men anbefaler førstnevnte.

[Eksempel på metadata-fil \(Difi@Github\)](#)

ALTINN (FORMIDLINGSTJENESTEN)

INNLEDNING

Formidlingstjenesten er delen av infrastrukturen som sørger for kommunikasjonen mellom medlemsregisteret og Brønnøysundregistrene. Formidlingstjenesten er utviklet og driftes av Accenture, og eies av Altinn.

For å kunne sende og motta filer gjennom formidlingstjenesten er det nødvendig at foretaket som representerer medlemsregisteret (typisk selskapet som utvikler og/eller drifter systemet) setter opp brukere og tilganger, samt laster opp et sertifikat i Altinn. Dette gjøres via Altinns nettsider i det aktuelle miljøet.

Når utvikler arbeider mot testmiljøet brukes fiktive organisasjoner som er satt opp på forhånd. Det er kun i produksjon at utvikler bruker faktiske opplysninger om organisasjonen. Oppsett av disse organisasjonene gjøres av Brønnøysundregistrene, og utvikler mottar innloggingsinformasjon for dette fra dem.

OFFISIELL DOKUMENTASJON

Mer informasjon om hvordan organisasjonene integrerer seg mot formidlingstjenesten til Altinn finnes i [implementasjonsguiden til Brønnøysundregistrene \(krever tilgang\)](#).

TILGANGER

For at organisasjonene skal kunne sende inn dokumenter til formidlingstjenesten, må de sette opp brukere med virksomhetssertifikater, brukernavn og passord i Altinn. For utfyllende informasjon om dette, se [implementasjonsguiden \(krever tilgang\)](#).

I tillegg er organisasjonene nødt til å sørge for at de som representerer medlemsregisteret, er gitt tilgang i "Tjenesteeierstyrt rettighetsregister" i både test- og produksjonsmiljø. Dette må håndteres direkte av Brønnøysundregistrene.

TEKNISK BESKRIVELSE

Formidlingstjenesten har krav som gjør den noe mer komplisert å kommunisere med. Der hvor Folkeregisteret og Brønnøysundregistrene ikke krever annet enn tilgangskontroll (brukernavn og passord i riktig format), krever formidlingstjenesten både en spesiell type sertifikat og en lite brukt sikkerhetsprotokoll ved navn WS-SecureConversation (WS-SC), som også krever SOAP 1.2.

Disse kravene gjør at utvalget av bibliotek og teknologier som kan brukes til å implementere denne delen av prosessen blir vesentlig mindre. Dersom utviklere ønsker å implementere WS-SC på egenhånd, bør de vite at dette krever en dyp forståelse av XMLDsig med signering av flere elementer i samme dokument (noe mange bibliotek for behandling av XML i seg selv ikke nødvendigvis støtter), manipulering av sesjoner, HTTP-headere og request/response-strukturen i SOAP-biblioteket de bruker.

Dersom dere ønsker mer informasjon om spesifikasjonen og dens aspekter, se:

[WS-SecureConversation 1.4 Specification \(OASIS\)](#)

[XMLDsig \(W3C\)](#)

Hvorfor ikke Python?

For alle deler av dette dokumentet har vi beskrevet og gitt eksempler på hvordan dette kan gjøres i Python. Dette er ikke tilfellet for formidlingstjenesten. Årsaken til dette er at SOAP og lignende teknologier har en noe begrenset støtte i Python. Språket er mye brukt til webapplikasjoner og scripting, men mange av sikkerhetsprotokollene under SOAP er ikke støttet av noen implementasjon i Python per i dag. Vi har derfor valgt å gå for en løsning i Java, som er beskrevet under.

Bibliotek

Navn	Språk	Lenke	Kommentar
CXF	Java	http://cxf.apache.org/	Open source, utviklet av Apache
WCF	C#	https://docs.microsoft.com/en-us/dotnet/framework/wcf/	Microsofts eget rammeverk for C#-applikasjoner

Altinn Broker Service Toolkit (BST)

Denne komponenten er den eneste beskrevet i dette dokumentet som leveres som en ferdig applikasjon. Årsaken til dette er at den ikke inneholder noen form for proprietær informasjon og fint kan brukes og utvides fritt av andre. Det er også mulig å bruke det som et bibliotek i en annen Java-applikasjon dersom utvikler ønsker dette.

Struktur

Applikasjonen er strukturert som en microservice og kan være frittstående dersom utvikler ønsker det. Det kreves ingen database eller tredjepartsverktøy for å kjøre den. For å gjøre det enklest mulig, har vi unngått å legge inn for mange abstraksjoner.

Applikasjonen gir et REST-grensesnitt med JSON for handlingene utvikler utfører mot formidlingstjenesten. Rammeverket som brukes er [SpringBoot](#).

Installasjon og bruk

Applikasjonen bruker Maven som verktøy for konfigurasjon, avhengigheter og bygging. Maven kan enten lastes ned og installeres som en egen applikasjon, eller brukes som integrert verktøy gjennom en IDE. JetBrains IntelliJ IDEA har innebygd støtte for Maven, og er gratis. Dersom utvikler kun ønsker å sette opp og kjøre applikasjonen, kan de installere den på egen hånd.

<https://maven.apache.org/>

<https://www.jetbrains.com/idea/?fromMenu>

Installere

Med prosjekt-roten som nåværende mappe, kjør:



```
mvn clean install
```

Denne kommandoen kompilerer og tester applikasjonen, før den produserer en unix executable fil i target/ med navn altinn-broker-service-toolkit-<versjon>.war. Versjonsnummer settes i pom.xml, som også ligger i roten av prosjektet.

Konfigurasjon

Før utvikler kan compilere applikasjonen til bruk, må den konfigureres for miljøet den kjører i. For at den skal være konfigurert, kreves det en application.properties-fil, samt et virksomhetssertifikat (se [sertifikater](#)).

Under src/main/resources/config/ ligger det en eksempelfil for application.properties som kan brukes som utgangspunkt. Husk at man her har en egen konfigurasjonsfil, og eget sertifikat, for hvert miljø man kjører i. Følg guiden gitt i [offisiell dokumentasjon](#) for å se hvordan man setter opp brukere for formidlingstjenesten for sitt foretak.

API

Tabellen under beskriver endepunkter, metoder og parametere for APIet.

Alle endepunkter ligger under /api/.

Funksjon	Endepunkt	Metode	Parametere
Hent tilgjengelige filer	getAvailableFiles	GET	- status [String, valgfri]
Last opp ny fil	uploadFile	POST	- file_name [String] - sender_reference [String] - file_data [String] - mime_type [String]
Bekreft nedlasting	confirmDownload	POST	- reference [String]
Start ny opplasting	initiateUpload	POST	- file_name [String] - reference [String]

Alle endepunkter i BST returnerer JSON.

Driftsmiljø

Altinn har et robust driftsmiljø som er driftet av eksterne parter. Det anbefales å melde seg på driftvarsel-listen, som er tilgjengelig på <https://altinn.brreg.no/sites/dokument2/Lists/Varsling%20av%20driftshendelser/AllItems.aspx> (del av sluttbrukersystem-portalen).

For å kunne melde seg på denne listen må utvikler først ha en Altinn SharePoint-bruker. Dersom du ikke har, kan du kontakte Altinn-servicedesk (også kalt “Avdeling for digitalisering servicedesk”). Kontaktinformasjon finner du på <https://altinnett.brreg.no/no/Veiledning/Datakonsumenter/Support/>.

Merk at ettersom varsling skjer ved at du abonnerer på en liste i SharePoint og får beskjed om endringer, vil disse meldingene gå inn i eventuelle andre varsler om endringer. Det medfører også at det ikke er mulig å få varsel til ikke-personlig adresse, slik som driftskonto eller liknende. Medlemmene i driftsteamet til leverandør må derfor individuelt få brukere i Altinn sin SharePoint-løsning.

SERTIFIKATER OG AUTENTISERING

Visse deler av infrastrukturen for denne prosessen har strengere krav til sikkerhet enn andre, noe som er eksemplifisert av formidlingstjenesten. Enkelte av tjenestene (Folkeregisteret og Brønnøysundregistrene) krever kun brukernavn og passord, mens andre (ID-porten og Altinn) krever bruk av en fil for sikker kommunikasjon ("sertifikat").

Sertifikater utstedes av enkelte pålitelige leverandører og garanterer sikker kommunikasjon mellom medlemsregisteret og de enkelte tjenestene. De har begrenset gyldighet og må typisk fornyes hvert tredje eller femte år. Merk at utstedelse av sertifikater også innebærer en teknisk verifiseringsprosess (avhengig av leverandøren som utsteder sertifikatet) og en kostnad ved utstedelse og fornying.

For å kunne brukes til kommunikasjon med Altinn, må sertifikatet være lastet opp i Altinn og tildelt rettigheter på vegne av foretaket som leverer medlemsregisteret. Dette gjøres ved at en person med tilstrekkelige tilganger for foretaket i Altinn (eks. daglig leder), logger inn og laster opp sertifikatet.

For å kunne brukes til kommunikasjon med ID-porten, må det lages et teknisk dokument (kalt "SAML Service Provider Metadata") som inneholder relevant informasjon om medlemsregisteret. Denne sendes til Difi per e-post.

TEKNISK BESKRIVELSE

For noen av tjenestene holder det med et brukernavn og passord, enten i meldingen eller i header over HTTPS. Dette er tilfellet for både Folkeregisteret og Brønnøysundregistrene sine SOAP-baserte tjenester.

Virksomhetssertifikater

For datautveksling med ID-porten (Difi) og Formidlingstjenesten (Altinn) kreves det virksomhetssertifikater. Merk at det kreves **to** virksomhetssertifikater, og begge må være gyldige og kjøpt fra en gyldig utsteder. For produksjonsmiljø benyttes sertifikat utstedt i navnet til foretaket som drifter medlemssystemet.

For testmiljø benyttes test-sertifikat utstedt til test-selskapet til foretaket som drifter medlemssystemet. Hvert enkelt medlemsregister som skal benytte seg av integrasjonsløsningen, får utstedt en test-organisasjon med gyldig organisasjonsnummer mot testmiljøene til Brønnøysundregistrene og Difi. Selv om dette er testmiljø, kreves det fortsatt kjøpte, gyldige sertifikater.

Vær oppmerksom på at sertifikatene har begrenset gyldighet. Typisk er gyldigheten på tre eller fem år, men det kan variere mellom leverandører. Alle som implementerer løsningen bør ha rutiner på å kjøpe nye sertifikater i god tid før utløp av de gamle.

Vi har selv valgt å bruke Buypass, men det finnes andre leverandører. Godkjente leverandører til dialog med Altinn per dags dato er listet opp under.

Tilbyder	Lenke
BuyPass	https://www.buypass.no
Commfides	https://www.commfides.com

Disse sertifikatene kommer gjerne i [PKCS#12](#)-formatet, og må for visse bruksområder bearbeides for å kunne brukes. Det finnes flere måter å gjøre dette på, men vi legger ved noen beskrivelser under, som alle viser hvordan de forskjellige formatene kan eksporteres ved hjelp av **openssl**-verktøyet i Unix.

Ved konfigurering av BST kan sertifikatet i sitt originalformat brukes, da Java-plattformen kan åpne og lese disse sertifikatene som det er.

Eksportere X.509-sertifikat i PEM-format

Når virksomhetsbrukere skal registreres i Altinn, må det også lastes opp et sertifikat som tilhører brukeren. Bruk autentiseringssertifikatet og gjør følgende:



```
openssl pkcs12 -in <sertifikatfil>.p12 -out <ønsket navn>.crt.pem -clcerts
-nokeys
```

Dette gir sertifikatet i PEM-format, med metadata som deretter må tas vekk for at sertifikatet skal bli akseptert av Altinns nettside. Fjern toppdelen av sertifikatet og la følgende være igjen:



```
-----BEGIN CERTIFICATE-----
MIIFG...ARVlaW
-----END CERTIFICATE-----
```

Deretter kan sertifikatet lastes opp.

Eksportere X.509-nøkkel i PEM-format

Dersom utvikler ønsker å bruke python-saml-pakken vi har lenket til andre steder i dokumentet, må de også ha nøkkel og sertifikat fra virksomhetssertifikatet. Sertifikatet skal være i samme format som sertifikatet som lastes opp i Altinn, og nøkkelen hentes ut på følgende måte:



```
openssl pkcs12 -in <sertifikat> -out <ønsket filnavn>.key.pem -nocerts
-nodes
```

“OG SÅ HAR ME JUKSA LITT...”: EKSEMPLER

Her har vi inkludert noen tekniske eksempler.

OPPSLAG MOT FOLKEREGETERET

Under følger eksempel på bruk av vårt bibliotek, **pydsf**. Dette er tenkt som en enkel måte å oppnå kontakt med tjenesten og foreta søk på. Merk at dette biblioteket ikke nødvendigvis støtter alle funksjoner i Folkeregisteret, da det er laget med dette formålet for øye, men det kan bli utvidet over tid med flere og mer avansert funksjoner. Funksjonene som vises under har flere alternativer og muligheter enn de som er gitt. Disse er dokumentert i koden.

Oppsett av klient

```
from pydsf.client import DSFClient
client = DSFClient(url, username, password, system_name)
```

pydsf.client.DSFClient		
Parameter	Type	Beskrivelse
url	string	Adressen til tjenesten, avhengig av miljø.
username	string	Brukernavn, gitt av EVERY
password	string	Passord, gitt av EVERY
system_name	string	Navnet på systemet som skal gjøre oppslag.

Foreta spørring

```
response = client.find_person(end_user='someuser', first_name='tomas',
last_name='topstad', ssn='01010750160')
```

pydsf.client.DSFClient.find_person		
Parameter	Type	Beskrivelse
end_user	string	Identifiserende informasjon om brukeren som gjør oppslaget
first_name	string	Fornavn på personen det søkes på
last_name	string	Etternavn på personen det søkes på
ssn	string	Fødselsnummer (11 siffer) på personen det søkes på

SDO (FRIVINTVERIFISERTMELDING)

I dette eksemplet referer integrasjonERFV til endringsmeldingen som XML, og saml:Assertion for hver signatur, også i XML.

```
from pybrreg.models.sdo import BrregSDO
```



```
data = {
    'change_report': '[integrasjonERFV]',
    'authentications': [
        '[saml:Assertion]', '[saml:Assertion]'
    ],
    'secret': '[pre-shared secret]'
    'ns_prefix': '[custom namespace prefix]'
}
```

```
document = BrregSDO(**data)
```

RECIPIENTS.XML

```
from pybrreg.models.recipients import BrregRecipientList
```



```
data = {
    'parties': [
        '[recipient 1]',
        '[recipient 2]'
    ]
}
```

```
document = BrregRecipientList(**data)
```

MANIFEST.XML

```
from pybrreg.models.manifest import BrregManifest
```

```
data = {  
    'senders_reference': '[UUID reference]',  
    'reportee': '[reportee ID]',  
    'file_list': [  
        ('[file name]', '[checksum]'),  
        ('[file name 2]', '[checksum 2]'),  
    ],  
    'property_list': [  
        ('[property]', '[value]'),  
        ('[property 2]', '[value 2]')  
    ]  
}
```

```
document = BrregManifest(**data)
```

ERFARINGER FRA PROSJEKTET

TIPS OG RÅD FOR BYGGING AV MELDING

Vær oppmerksom på at det er knyttet en rekke valideringskrav til de ulike feltene som kan meldes inn. Dette må valideres av systemet selv før meldingen sendes til prekontroll.

Gjør en vurdering av hvordan de ulike feltene skal behandles. Ligger data både hos Brønnøysundregistrene og systemet, må det gjøres en sammenligning første gang, slik at systemene blir synkronisert med samme opplysninger. Ligger data kun hos Brønnøysundregistrene, må data allikevel vises, og feltene må gjøres tilgjengelig for endring for brukeren.

FOKUS PÅ BRUKER

En suksessfaktor er at man i utviklingen av løsningen hele tiden har fokus på bruker. Det er viktig med brukerinvolvering gjennom hele prosessen, inkludert jevnlig brukertesting. Det er viktig at det finnes ressurser med den rette UX-kompetansen slik at brukergrensesnittene blir både brukervennlige og har en brukerreise som sikrer at bruker faktisk melder de korrekte opplysningene.

NYTTIG OM TESTING MOT TESTSYSTEMET

Vær tidlig ute med å hente inn relevant testdata fra Brønnøysundregistrene. Det ligger enkelte eksempler på korrekt genererte XML-filer i Confluence, men det kan være dere ser behov for flere eksempler for å teste spesifikke caser.

Vær oppmerksom på at testbasen til Brønnøysundregistrene oppdateres med jevne mellomrom, slik at organisasjonsnummer overskrives av nye foreninger. Mer info fås fra Brønnøysundregistrene.

For ende til ende-tester må dere koordinere med Brønnøysundregistrene på forhånd, slik at dere vet at de har ressurser til å behandle meldingene dere sender inn. Sett av flere dager eller mer til slik testing, da saksbehandling, mottak av vedtak og korrespondanse med kontaktperson for testingen hos Brønnøysundregistrene av naturlige årsaker kan ta tid.

Responstid per oktober 2019

Applikasjonskvitteringer genereres automatisk og er tilgjengelig i løpet av få minutter.

Vedtak kan ta en times tid fra saksbehandler hos Brønnøysundregistrene sender det fra seg og til det er tilgjengelig for henting fra meldingsboksen i Altinn.

AVSLUTTENDE ORD

Dokumentet dere nå har lest er ment å gi dere som organisasjon et startpunkt for å vurdere å ta i bruk integrasjonsløsningen i deres systemer. Det er også en samling av informasjon og erfaringer gjort i utviklings- og testfasen. Når dere som organisasjon først har konkludert med å starte på dette prosjektet og har signert avtaler, vil dere få tilgang til mer teknisk info.

En avklaring som kom frem i prosjektperioden, som det kan være greit å merke seg, er at integrasjoner som Folkeregisteret, signatur via ID-porten og Brønnøysundregister-oppslag av data, bare kan brukes i sammenheng med meldinger til Brønnøysundregistrene. Med andre ord må leverandører beholde for eksempel sine eksisterende signaturløsninger, og de kan ikke benytte integrasjoner de får tilgang til via dette prosjektet for andre formål enn innsending av meldinger til Brønnøysundregistrene.

Det er ulike erfaringer med bruk av løsningen i flere ulike nasjonale organisasjoner. Noen organisasjoner setter stor pris på integrasjonen og har god erfaring, mens andre har store utfordringer med å få lokallag til å prioritere tid til å sjekke informasjon i Brønnøysundregistrene. Vi vil tro at den samme erfaringen vil være gjeldende for senere leverandører. Mye er altså avhengig av sentralleddets innstilling og motivasjon for å ta i bruk løsningen.

Ved å starte en integrasjon tar leverandøren på seg et stort ansvar. Meldinger til Brønnøysundregistrene er juridisk bindende dokumenter, og organisasjonene må oppfylle mange lovkrav. De skal gjennom en lang og komplisert prosess, og det er viktig å være klar over at endringer vil komme også i fremtiden. Leverandører som iverksetter dette prosjektet, får mange fordeler, men også en del forpliktelser som må oppfylles.

Erfaringer fra HyperSys, fra før integrasjonsmodulen er tatt i bruk, viser at så lite som under 20 prosent av lokallagene som har registrert organisasjonsnummer, faktisk har synkronisert all informasjon med Brønnøysundregistrene. Det er med andre ord stort rom for forbedring, og en lengre testperiode viser klart at det er store fordeler med en direkte integrasjon mot Brønnøysundregistrene for de organisasjonene som velger å aktivt ta det i bruk.

Informasjon som er utdatert er et stort problem med lokallag i Norge, da mange personer som tidligere satt i styret til en organisasjon fortsatt står oppført som juridisk og økonomisk ansvarlige - ofte uten at det sittende styret er klar over at rett informasjon ikke er sendt inn til staten. Samtidig forventes det stadig mer profesjonell og effektiv oppfølging fra frivillige organisasjoner, og et medlemsregister som holder informasjonen oppdatert er uten tvil en forenklende brikke i dette puslespillet.

Det er med andre ord et stort potensiale for mer oppdatert data i offentlige registre, og det nye er at sentralleddet kan ha en viktig funksjon her ved å holde oversikt over lag som må rapportere endringer, og (til en viss grad) å bistå med dette.

**Prosjektgruppa takker for samarbeidet,
og ønsker lykke til med arbeidet.**

Frivillighet Norge
Brønnøysundregistrene
Unicornis

Brønnøysundregistrene | brreg.no | 75 00 75 00
Unicornis | unicornis.no | 22 99 37 40
Frivillighet Norge | frivillighetnorge.no | 21 56 76 50