Tjeneste for brukerhåndtering for Alma

# 1. Innledning

Brukerdata (studenter) ligger i «Felles studentsystem» (FS). FS tilbyr en REST-tjeneste som gjør brukerdata tilgjengelig.

* Framdrift: [http://www.fellesstudentsystem.no/dokumentasjon/brukerdok/fswebservice/fs-api/framdrift-i-utviklingen.html](http://www.fellesstudentsystem.no/dokumentasjon/brukerdok/fswebservice/fs-api/framdrift-i-utviklingen.htmlH). Message-kø er ikke dokumentert.
* Swagger til REST-API: <https://docker-demo.fsat.no/fsapi/swagger.html#!/personrolle/get_personroller_id> (bibsys, Bi69kKaaY56). (Epost fra Richard Edvin Borge: "Akkurat nå får dere tilgang til utviklingsbasen vår, og den er litt ustabil, siden den er i utvikling. Hvis dere vil ha tilgang til en demobase hos en institusjon for å ha litt mer stabilt miljø og reelle data, så må det innhentes tillatelse fra institusjon.")
* Dokumentasjon av tjeneste: <https://www.fellesstudentsystem.no/dokumentasjon/teknisk/fsws-dok/rest/bas.html> (Pålogging virker på med med NTNU-Feide.)

Alma har en REST-api hvor brukerdata kan håndteres.

* Dokumentasjon og REST-service: <https://developers.exlibrisgroup.com/alma/apis/users> (brukernavn: [47BIBSYS\_UBO-DEV](http://bum.bibsys.no/projects/Alma/Service%20API/Developers%20Network%20users%20and%20API-keys%20for%20Alma-production.xlsx), passord: Fw90wiqq)

Det trengs en «data-pumpe» som henter data fra FS, konverterer FS-format til Alma-format og poster disse inn i Alma. Noen data må mappes. Her er det spesielt viktig å mappe «institusjon» i FS til Alma-instans-kode sammen med «institutt» til «campus».

I FS kalles brukerdata person. I Alma er disse User. BIBSYS bruker ofte bruker eller låntaker. BIBSYS-konverteringsmotor vil omtale dette data-object som «*person*».

Det er planlagt å lage en Message-kø av avd. FS/CERES som avd. BIBSYS kan abonnere på, som informerer oss om endringer i FS-basen. Dette skal trigge hentingen av *person* fra FS sin REST-tjeneste og overføre disse til Alma.

BIBSYS-Konverteringsmotor

Alma

ALMA-REST

FS-REST

Message-kø

FS

# 2. Antagelser

* 1 to 1 forhold: Det finnes et REST-endepunkt respektive en Message-kø til hvert institusjon i FS. Dermed er det på forhånd klart for oss hvilken Alma-Instans vi må kommunisere med.
* Meldinger fra message-køen gir oss hendelsen (update, create, delete) per post-id, per *person* og per institusjon. Vi antar at det finnes en message-kø per REST-endepunkt (og per institusjon).
* Vi har mappingen fra FS-institusjonsregister til Alma-instans resp. Alma-Campus. Vi kan slå det opp en plass (BIBSYS institusjons-register?).
* Den eneste rollen som settes aktiv, vil være «PATRON». Andre roller som finnes fra før i Alma, vil ikke bli overskrevet.
* FS-personlopenr og ALMA-primary\_id identifiserer samme *person*.
* «merge *person*» blir ikke håndtert.
* Feide-id'er blir ikke automatisk genererte.
* Det er ikke noen kontakt med LDAP eller Feide-instanser.
* Text-encoding er UTF-8.

### 2.1 FS-person

GET https://docker-demo.fsat.no/fsapi/personer/1234

Response:

{

"meta": {

"href": "https://docker-demo.fsat.no/fsapi/personer/1234"

},

"personlopenr": 1234,

"fnr": "16537990108",

"fodselsdato": 165379,

"personnr": 90108,

"brukernavn": "16537990108",

"fornavn": "Terje",

"etternavn": "Larssen",

"kjonn": "M",

"adresser": [

{

"type": "HJEMSTED",

"gate": "Fyrstikkbakken 13",

"sted": "OSLO",

"postnr": 667

}

],

"harBilde": false,

"kvalifikasjonsGrunnlagKode": "ATA",

"kvalifikasjonsGrunnlagNavn": "Allmenne fag, tradisjonell struktur",

"student": {

"nr": 318,

"studierett": true

}

}

# 3. BIBSYS-konverteringsmotor

BIBSYS trenger en *person* som kan synkroniseres med Alma-User-Object. Denne User er spesifisert her (<https://developers.exlibrisgroup.com/alma/apis/xsd/rest_user.xsd>). For å kommunisere med ALMA må man har tak i en api-key som bestemmer ALMA-instansen. Denne api-key får en tak i ved hjelp av InstitutionRegistry (<http://haris.bibsys.no:8084/>).

For å definere data-strukturen til *person* som skal brukes i BIBSYS-konverteringsmotor, så holder vi oss nær datamodellen fra Alma-User.

## 3.1. Data-strukturen til *person*

|  |  |  |
| --- | --- | --- |
| Felt | Type | Beskrivelse |
| record\_type |  | STAFF eller PUBLIC |
| primary\_id | String | unique identifikator til *person* (NB! case insensitive) |
| first\_name | String | fornavn |
| last\_name | String | etternavn |
| full\_name | String | fullt navn |
| gender |  | FEMALE, MALE, NONE |
| user\_group |  | Tall-kode etter [User\_group kodetabell](#_3.1.2._User_group_kodetabell) |
| library\_number |  | biblioteksnummer er avhengig av institusjon |
| preferred\_language |  | To-bokstaver sparkkode etter ISO639-1 |
| birth\_date | Date | yyyy-MM-dd |
| expiry\_date | Date | yyyy-MM-dd, forvented gyldighetsdato til *person* som låntaker på institusjonen |
| status |  | ACTIVE, INACTIVE, DELETED |
| Contact\_info | Contact\_info | Inneholder lister over [eposter](#_3.1.1.1_Emails_data-struktur), [telefoner](#_3.1.1.1_Phones_data-struktur) og [adresser](#_3.1.1.1_Addresses_data-struktur) |
| User\_identifiers | User\_identifiers | Liste om *person*s andre [identifikatorer](#_3.1.2_Identifiers_data-struktur) |

### 3.1.1 Contact\_info data-struktur

Inneholder lister over eposter, telefoner og adresser til en *person*.

#### 3.1.1.1 Emails data-struktur

Liste over alle eposter fra en *person*.

|  |  |  |
| --- | --- | --- |
| Felt | Type | Beskrivelse |
| email-address | string | epostadresse (obligatorisk) |
| email\_type |  | personal, work, school, alternative |
| preferred | boolean | true, false (obligatorisk) |

#### 3.1.1.2 Phones data-struktur

|  |  |  |
| --- | --- | --- |
| Felt | Type | Beskrivelse |
| phone | string | telefonnummer (obligatorisk) |
| phone\_type |  | home, mobile, office, fax |
| preferred | boolean | true, false (obligatorisk) |

#### 3.1.1.3 Addresses data-struktur

|  |  |  |
| --- | --- | --- |
| Felt | Type | Beskrivelse |
| line1 | string | gate (obligatorisk) |
| postal\_code | String | postnummer |
| city | String | poststed |
| country |  | landkode etter ISO3166-1 alpha-3 |
| address\_type |  | home, work, office, school, alternative |
| preferred | boolean | true, false (obligatorisk) |

### 3.1.2 Identifiers data-struktur

Identifiers er en liste over andre identifikatorer til en *person*. (NB! id\_type = UNIV\_ID blir brukt for å lagre Feide-Id).

|  |  |  |
| --- | --- | --- |
| Felt | Type | Beskrivelse |
| id\_type | string | BARCODE, UNIV\_ID, FACEBOOK, GOOGLE, TWITTER, OTHER\_ID\_1 (obligatorisk) |
| value | String | identifikatorverdi |

### 3.1.3. User\_group kodetabell

|  |  |
| --- | --- |
| kode | beskrivelse |
| 1 | Egne studenter |
| 2 | Egne studenter – høyere grad |
| 3 | Egne fjernstudenter |
| 4 | Egne ansatte |
| 5 | Eksterne stipendiater ved egen institusjon og gjesteforelesere |
| 6 | Studenter, stipendiater og ansatte ved andre høyere utd. inst. |
| 7 | Enkeltpersoner |
| 8 | Bedrifter og institusjoner uten bibliotek |
| 10 | Andre bibliotekenheter innen samme institusjon |
| 11 | Universitetsbibliotek i Norge |
| 12 | Høgskolebibliotek i Norge |
| 13 | Bedriftsbibliotek i Norge |
| 14 | Andre fag- og forskningsbibliotek |
| 15 | Nasjonalbiblioteket |
| 16 | Folkebiblioteksektoren |
| 17 | Bibliotek i grunn- og videregående skole |
| 20 | Bibliotek og lignende institusjoner i Danmark |
| 21 | Bibliotek og lignende institusjoner i Sverige |
| 22 | Bibliotek og lignende institusjoner i Finland |
| 23 | Bibliotek og lignende institusjoner på Island |
| 24 | Øvrige europeiske bibliotek |
| 25 | Verden ellers |
| 75 | Sommerskolestudenter |
| NOTFOUND | Ukjent kategori |

### 3.2. Faste verdier

Alma User må har flere felter fyllt ut enn det vi har bruk for. Noen Felter settes av BIBSYS-konverteringsmotor automatisk.

«job\_category» er et felt som inneholder informasjon om stillingen til *person*. Med fokus på FS *person* setter vi dette feltet vanligvis tomt. Det kan brukes i sammenheng med ansatte ved institusjon som ikke har en ALMA relatert rolle, f.eks. forelesere.

«account\_type» er alltid EXTERNAL med «external\_id» SIS.

«user\_roles» er et felt som styrer rolleoppsett i Alma. Med fokus på FS *person* setter vi bare «Patron».

### 3.3. Interne mappinger av felt/objekter

«campus\_code» og «rs\_libraries» kan bli generert ut fra «library\_number» (biblioteksnummere finnes i BaseBibliotek: <https://www.nb.no/BaseBibliotekSearch/rest/bibnr/1160103>). Vi kan slå opp ALMA campus\_code hos oss med hjelp av InstitutionRegistry (<http://haris.bibsys.no:8084/>). rs\_libraries er en liste av rs\_library. Normalt er det bare ett som genereres ut fra library\_number.

## 4. Kommunikasjon mellom FS og BIBSYS-konverteringsmotor

FS tilbyr en REST-tjeneste for å få tilgang til blant annet *person*-data:

* GET personer: <https://docker-demo.fsat.no/fsapi/swagger.html#!/person/get_personer>
* GET personer/id: <https://docker-demo.fsat.no/fsapi/swagger.html#!/person/get_personer_id>
* GET personroller ?
* GET personroller/id ?

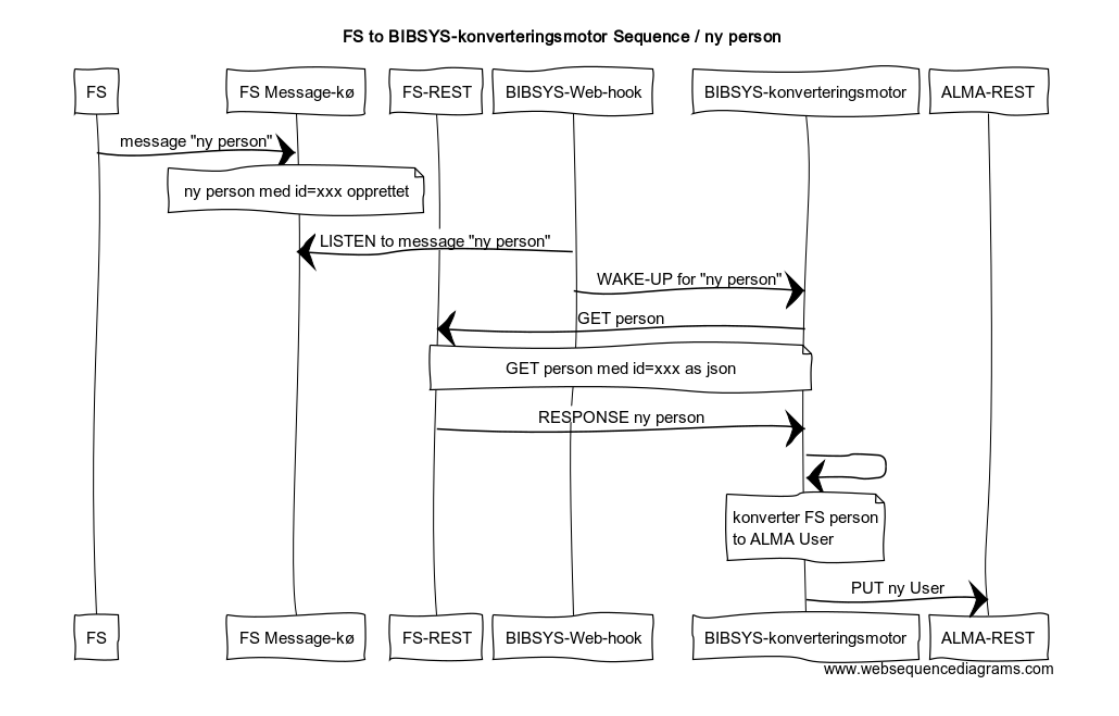
FS bruker sitt eget institusjonsregister. Det har inntil fire nivåer: institusjon, fakultet, institutt og gruppe. BIBSYS InstitutionRegistry har institution og library. Vi må lage en mapping mellom disse eller utvide vår InstitutionRegistry med FS sine koder.

### 4.1. Message-kø fra FS og Web-hook i BIBSYS-konverteringsmotor

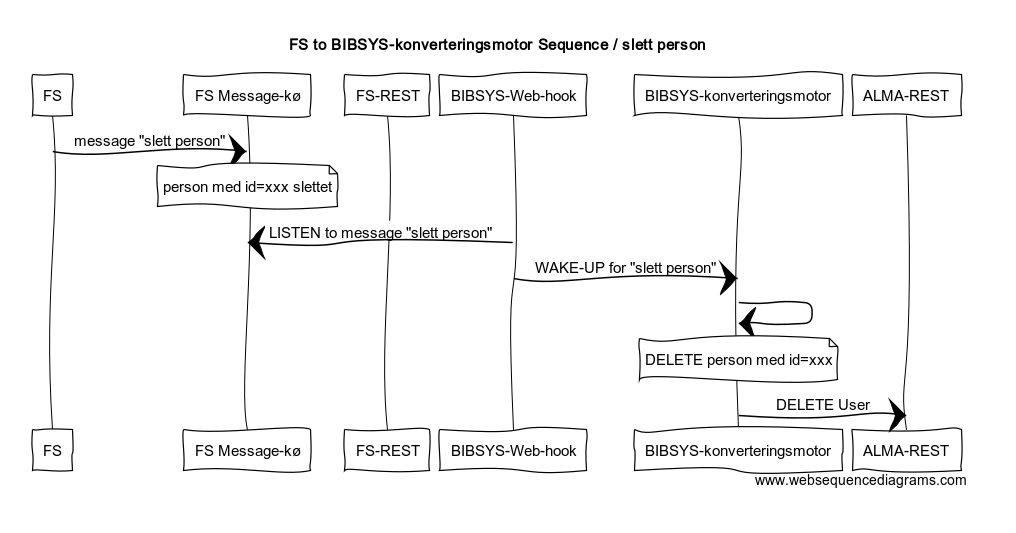
FS vil sette opp en Message-kø som vi kan abonnere på. Ved hjelp av denne message-køen blir det annonsert om en post har endret seg eller blitt slettet, eller om en ny post er blitt opprettet. BIBSYS vil lytte på denne Message-køen med en Web-hook. Etterpå kan BIBSYS hente data fra FS REST-tjeneste.

Da Message-køen inneholder nok informasjon om hvilke felter er endret, vil BIBSYS slippe å lese data på REST. Det vil forenkle BIBSYS-konverteringsmotoren og dermed gjør den raskere.

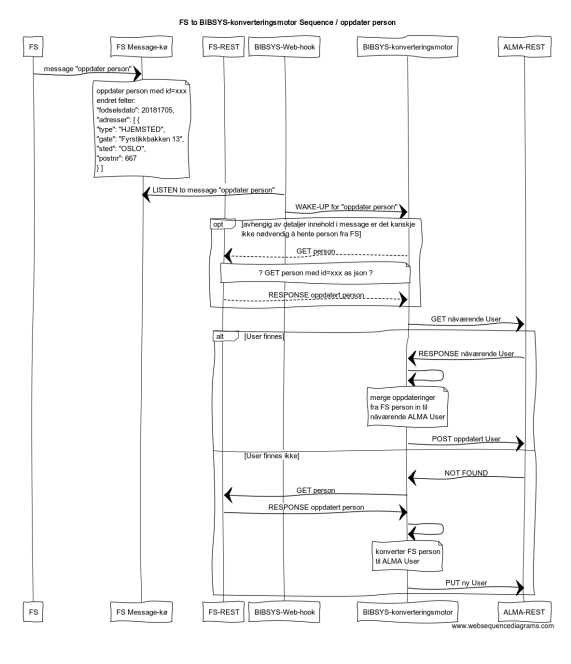
#### 4.1.1 Ny person



#### 4.1.2 Slett person



#### 4.1.3 Update person



## 5. REST-grensesnitt fra BIBSYS-konverteringsmotor til bruk utenom FS integrasjon

Det er rett fram å sette opp et REST-endepunkt til BIBSYS-konverteringsmotor. REST-apien ligner den fra ALMA i en slanket versjon.

* Lesing av en person: GET bups.kdto.no/person/id
* Søk eller uthenting av flere personer: GET bups.kdto.no/person/?q=query&limit=100&offset=0&order\_by
* Oppdatere en person: PUT bups.kdto.no/person/id
* Lage en ny person: POST bups.kdto.no/person/
* Slette en person: DELETE bups.kdto.no/person/id

Denne REST-tjenesten må være beskyttet med en autoriseringsmekanisme som f.eks. api-key. Det er i dag problematisk siden BIBSYS ikke enda har noe arkitektur for autorisasjon/autentifisering. BIBSYS har ikke noe eget opplegg for hvordan vi distribuerer og administrere passord/brukernavn og heller ikke api-key’er. Dette bør løses på bedriftsnivå og betraktes derfor ikke som en del av tjenesten for brukerhåndtering.

## 6. Åpne spørsmål

* Når er Message-kø fra FS på plass? Er den definert allerede?
* Hva skal BIBSYS-konverteringsmotor gjøre når en «slett person» feiler f.eks. pga aktive lån. (Sette ALMA User INAKTIV, setter user\_blocks???)
* Har hvert FS-REST endepunkt sin egen pålogging/autentisering?
* Hvordan ønsker vi å håndtere passord/brukernavn styrt til FS-REST endepunkt?