

SOFD Core

Datamodel

Version: 1.0.0

Dato: 08.11.2019

Forfatter: BSG

Indhold

| | | |
|---|----------------------------------|---|
| 1 | Indledning | 3 |
| 2 | Objekter | 3 |
| 3 | Feltbeskrivelser | 3 |
| 4 | Sammenhæng mellem objekter | 7 |

1 Indledning

Dette dokument giver en overordnet beskrivelse af datamodellen i SOFD Core, i form af en feltbeskrivelse af de enkelte objekter, og forholdet mellem objekterne.

2 Objekter

SOFD Core vedligeholder 2 domæne-objekter, der er udstillet via API'et.

- Person
- OrgUnit

Disse to domæne-objekter er indgangen til alle andre objekter i SOFD Core, og såvel udlæsning som vedligehold af objekter sker altid gennem det ejende domæne-objekt.

Ikke alle felter på et objekt kan vedligeholdes via API'et. Nogle felter er interne statusfelter, og vedligeholdes af SOFD Core; disse er

- created
- lastChanged
- deleted
- manager (på OrgUnit)

3 Feltbeskrivelser

SOFD Cores objekter er struktureret som beskrevet nedenfor. Her skal man være opmærksom på at datamodellen udvides løbende efter behov, og specielt enums vil udvides løbende, og kan afvige fra kommune til kommune.

Alle objekter har et master og masterId, der identificerer hhv hvilken eksterne kilde der er master for objektet, samt hvad den unikke nøgle er det i kildesystem. Et samlet domæne-objekt udlæst af API'et, kan derfor være stykket sammen af data fra flere kilder.

```
class Person {
    String uuid;
    String master;
    Date created;
    Date lastChanged;
    boolean deleted;
    String cpr;
    String firstname;
    String surname;
    String chosenName;
    Date firstEmploymentDate;
    Date anniversaryDate;
    Post registeredPostAddress;
    Post residencePostAddress;
    List<Phone> phones;
    List<User> users;
    List<Affiliation> affiliations;
```

```
        String localExtensions;
    }

    class Post {
        String master;
        String masterId;
        String street;
        String localname;
        String postalCode;
        String city;
        String country;
        boolean addressProtected;
        boolean prime;
    }

    class Phone {
        String master;
        String masterId;
        String phoneNumber;
        PhoneType phoneType;
        FunctionType functionType;
        boolean prime;
    }

    class User {
        String uuid;
        String master;
        String masterId;
        String userId;
        String employeeId;
        UserType userType;
        String localExtensions;
        Email email;
        boolean prime;
    }

    class Email {
        String master;
        String masterId;
        String email;
        boolean prime;
    }

    class Affiliation {
        String uuid;
        String master;
        String masterId;
```

```
Date startDate;
Date stopDate;
boolean deleted;
boolean prime;
OrgUnit orgUnit;
Person person;
String employeeId;
String employmentTerms;
String employmentTermsText;
String payGrade;
Double workingHoursDenominator;
Double workingHoursNumerator;
AffiliationType affiliationType;
String positionId;
String positionName;
String positionTypeId;
String positionTypeName;
List<String> klePrimary;
List<String> kleSecondary;
String localExtensions;
List<AffiliationFunction> functions;
Set<OrgUnit> managerFor;
}
```

```
class OrgUnit {
    String uuid;
    String master;
    String masterId;
    boolean deleted;
    Date created;
    Date lastChanged;
    OrgUnit parent;
    List<OrgUnit> children;
    String shortname;
    boolean orderItUser;
    String name;
    Long cvr;
    Long ean;
    Long senr;
    Long pnr;
    String costBearer;
    String orgType;
    Long orgTypeId;
    String keyWords;
    String openingHours;
    List<Post> postAddresses;
    List<Phone> phones;
```

```
List<Email> emails;
OrgUnitManager manager;
List<Affiliation> affiliations;
List<String> klePrimary;
List<String> kleSecondary;
String localExtensions;
}

class OrgUnitManager {
    String name;
    Person manager;
    boolean inherited;
    OrgUnit orgUnit;
}

enum PhoneType {
    MOBILE,
    BROADBAND,
    IP,
    DATA_SIM,
    LANDLINE,
    FUNCTION
}

enum FunctionType {
    NONE,
    BUSINESS,
    ALARM,
    EMERGENCY,
    TECHNICAL_MONITORING
}

enum UserType {
    ACTIVE_DIRECTORY,
    OPUS,
    UNILOGIN
}
```

4 Sammenhæng mellem objekter

Ovenstående datamodel beskriver kardinaliteterne, så nedenstående illustration, viser alene ejerskabsrelationen (pilens retning).

