

Full SOR importer

Generelt

Der er lavet en separat pakke til denne importer i følgende namespace
`com.trifork.stamdata.importer.jobs.sor.sor2`
Navnet sor2 bør nok ændres til noget bedre.

Common ændringer

RecordSpecifikation er ændret så numeric nu er en Long da den svarer til en BIGINT i databasen.

Der er tilføjet en FOREIGN_KEY RecordFieldType som også svarer til en BIGINT, jeg tror dog ikke denne håndteres der hvor Atom feeded dannes.

Parser

Parsing sker med en standard SAX parser.

Der skelnes mellem XML noder der kan verificeres mod databasen og dem der ikke kan, fx kan man ikke bestemme om <PostalAddress> har ændret sig i forhold til databasen, da der ikke er nogen nøgle men kun tekst felter.

Derimod kan man godt se om <OrganizationalUnit> har ændret sig i forhold til databasen da den indeholder feltet <SorIdentifier> som man kan slå op (der snydes også ved at kigge i SorStatus).

Parseren virker simplificeret på følgende måde:

Parseren møder et åbnings tag:

Der oprettes et objekt tilsvarende navnet på noden, fx ved en <SorStatus> oprettes et SorStatus object.

Parseren møder et slut tag:

- Hvis nuværende objekt er sat til at have unik nøgle i xml bliver objektet sammenlignet med det der ligger i databasen er der forskel bliver det persisteret. Der efter bliver objektet fjernet fra hierarkiet
- Hvis dette objekt ikke er unik tilføjes det som barn til fader noden.

SorNode

SorNode
<ul style="list-style-type: none">- children: List<SorNode>- parentNode: SorNode- parentTag: String- PID: Long- dirty: boolean
<ul style="list-style-type: none">- SorNode(Attributes, parent, parentTag)- persist(RecordPersister)- <i>compareAgainstDatabaseAndUpdateDirty</i>(Fetcher)- addChild(SorNode)- removeChild(SorNode)- parseEndTag(String, String)- <i>compareAgainstDatabaseAndUpdateDirty</i>(RecordFetcher)

Der mappes fra XML til objekt hierarki vha. SorNode, for alle xml noder som skal i databasen gælder det at der skal oprettes en klasse som er nedarvet fra SorNode, den nedarvede klasse skal indeholde følgende:

- parseEndTag: Kode så den kan parse sine egne properties fra XML
- persist den skal kunne gemme sig selv
- compareAgainstDatabaseAndUpdateDirty sammenlign med databasen og sæt dirty flag hvis der er sket ændringer, dette er kun relevant på unikke noder.

Vi har brugt RecordSpecs for skrive i databasen

SorFullRecordSpecs klassen indeholder de recordspecs som vi har nået ind til videre.

Database tabeller

Følgene tabeller har vi nået der mangler fortsat nogle overordnede tabeller fx HealthInstitution og InstitutionOwner derudover mangler der også nogle små tabeller der er TODO's de steder vi har mødt nogen der mangler.

Skemaet nedenfor kan åbnes i MySQL Workbench som er crossplatform og gratis, det ligger i db/ folderen.

SORInstitutionOwner <ul style="list-style-type: none">PID BIGINT✦ sorIdentifier BIGINT✦ entityName VARCHAR(60)✦ ownerType BIGINT✦ SOREanLocationCodeId BIGINT✦ postalAddressInformationId BIGINT✦ virtualAddressInformationId BIGINT✦ sorStatusId BIGINT✦ fromDate VARCHAR(10)✦ toDate VARCHAR(10)✦ updatedAt VARCHAR(10)✦ validFrom DATETIME✦ validTo DATETIME✦ modifiedDate DATETIME <div>Indexes</div>	SOREanLocationCode <ul style="list-style-type: none">PID BIGINT✦ eanLocationCode BIGINT✦ onlyInternalIndicator CHAR✦ nonActiveIndicator CHAR✦ systemSupplier BIGINT✦ systemType BIGINT✦ communicationSupplier BIGINT✦ regionCode BIGINT✦ ediAdministrator BIGINT✦ sorNote VARCHAR(254)✦ fkSorStatus BIGINT✦ validFrom DATETIME✦ validTo DATETIME✦ modifiedDate DATETIME <div>Indexes</div>	SORVirtualAddressInformat... <ul style="list-style-type: none">PID BIGINT✦ emailAddressIdentifier VARCHAR(254)✦ website VARCHAR(254)✦ telephoneNumberIdentifier VARCHAR(20)✦ faxNumberIdentifier VARCHAR(20)✦ validFrom DATETIME✦ validTo DATETIME✦ modifiedDate DATETIME <div>Indexes</div> <div>PRIMARY</div>
SOROrganizationalUnit <ul style="list-style-type: none">PID BIGINT✦ sorIdentifier BIGINT✦ entityName VARCHAR(60)✦ fkOrganizationalChildUnit BIGINT✦ unitType BIGINT✦ localCode VARCHAR(20)✦ pharmacyIdentifier VARCHAR(20)✦ shakIdentifier VARCHAR(7)✦ providerIdentifier VARCHAR(9)✦ eanLocationCodeInheritedIndicator CHAR✦ fkEanLocationCode BIGINT✦ geographicalParentRelation BIGINT✦ geographicalParentSorIdentifier BIGINT✦ fkPostalAddressInformation BIGINT✦ fkVisitingAddressInformation BIGINT✦ fkActivityAddressInformation BIGINT✦ fkVirtualAddressInformation BIGINT✦ fkClinicalSpecialtyCollection BIGINT✦ fkSorStatus BIGINT✦ fkReplacesSorCollection BIGINT✦ fkReplacedByCollection BIGINT✦ ambulantActivityIndicator VARCHAR(1)✦ patientsAdmittedIndicator VARCHAR(1)✦ reportingLevelIndicator VARCHAR(1)✦ localAttribute1 VARCHAR(20)✦ localAttribute2 VARCHAR(20)✦ localAttribute3 VARCHAR(20)✦ localAttribute4 VARCHAR(20)✦ localAttribute5 VARCHAR(20)✦ validFrom DATETIME✦ validTo DATETIME✦ modifiedDate DATETIME <div>Indexes</div>	SORSorStatus <ul style="list-style-type: none">PID BIGINT✦ fromDate VARCHAR(10)✦ toDate VARCHAR(10)✦ updatedAt VARCHAR(10)✦ firstFromDate VARCHAR(10)✦ validFrom DATETIME✦ validTo DATETIME✦ modifiedDate DATETIME <div>Indexes</div>	SORPostalAddressInformation <ul style="list-style-type: none">PID BIGINT✦ mailDeliverySublocationIdentifier VARCHAR(34)✦ streetName VARCHAR(40)✦ streetNameForAddressingName VARCHAR(20)✦ streetBuildingIdentifier VARCHAR(10)✦ floorIdentifier VARCHAR(10)✦ suiteIdentifier VARCHAR(4)✦ districtSubdivisionIdentifier VARCHAR(34)✦ postOfficeBoxIdentifier INT✦ postCodeIdentifier INT✦ districtName VARCHAR(20)✦ countryIdentificationCodeScheme SMALLINT✦ countryIdentificationCode VARCHAR(10)✦ stairway VARCHAR(40)✦ validFrom DATETIME✦ validTo DATETIME✦ modifiedDate DATETIME <div>Indexes</div> <div>PRIMARY</div>