

Patients

Contents

- [Overview](#)
- [Patient](#)
- [PatientIdentifierType](#)
- [PatientIdentifier](#)

Overview

OpenMRS provides `Patient`, `PatientIdentifier`, `PatientIdentifierType` and objects through the API. A `Patient` is a [Person](#) and inherits all attributes/objects from that object. A `PatientIdentifier` is a medical record number. A patient can have any number of medical record numbers. Each number has a certain type: `PatientIdentifierType`. An example of a type would be a "hospital X id" or "Country X National ID"

Patient

A `Patient` object represents the reason we write OpenMRS. It represents a human that medical data ([Obs](#)) can be collected on. A `Patient` has very little metadata itself because it inherits from [Person](#).

Properties on a patient:

- `patientId` is a local database generated integer that is used internally (should not be used in external applications as it may change)
- 1 to n `PatientIdentifiers` (see below)

Inherited from [Person](#):

- 1 to n `PersonNames`
- 0 to n `PersonAddresses`
- 0 to n `PersonAttributes`
- ...see others as described on [Person API page](#)

PatientService

```
// Gets patients with the given criteria. only one of name/identifier is required. if identifier is non-null,
// identifier types are matched against and matchIdentifierExactly is considered
public List<Patient> getPatients(String name, String identifier, List<PatientIdentifierType> identifierTypes,
boolean matchIdentifierExactly)

// Saves the given Patient object into the database. Saves are cascaded to all objects on the Patient object
// (Identifiers, PersonName, PersonAddress, etc)
public void savePatient(Patient p);

// Marks the given patient as invalid. All obs and encounters associated to this patient are left as-is.
public void voidPatient(Patient p);

// Gets all patient objects in the system. If a database has more than 10k patients, this will probably cause
// an [OOM|docs:Troubleshooting Memory Errors]
public List<Patient> getAllPatients();
```

Example:

```
Patient pat = Context.getPatientService().getPatientByUuid("1234-56789-123");
pat.setGender("M");
pat.addName(new PersonName("Jim", "Bob", "Smith"));
Context.getPatientService().savePatient(pat);
```

A `Patient` can be linked to a user via the `Person`. If `User.getPerson()` is the `Person` parent class, they are the same person. Note: A `Patient` can have multiple user accounts linked to it with this modeling.

The `Patient` class source code can be seen [here](#).

PatientIdentifierType

Administrators define what types of identifiers they will collect. These range from National ID numbers, to driver's license numbers, to per-hospital medical record numbers.

Properties on PatientIdentifierType:

- format: a regular expression defining what the identifier text should contain.
- formatDescription: the text an admin can enter describing the regex format they just added
- required: a true/false whether every patient MUST have this type
- checkDigit: a true/false whether this identifier has a checkdigit at the end. See [Check Digit Algorithm](#) (this should ideally be done by the validators in 1.6+)
- validator: full class name of the IdentifierValidators
- locationBehavior: REQUIRED=there must be patient_identifier.location_id defined by the user for this identifier or NOT_USED = patient_identifier.location_id is null

See LuhnIdentifierValidator class for the typical validator

PatientIdentifier

A patient has any number of identifiers. The patient can be found using any of their identifiers. A number has to be unique across all PatientIdentifierTypes in the database.

Properties on PatientIdentifier:

- patient: the Patient object using this identifier
- identifier: the string for actual number. Must adhere to identifierType.format and validate according to identifierType.validator
- identifierType
- location: The location this was given to the patient. may or may not be required. see identifierType.locationBehavior
- preferred: if true, this is the identifier that should be displayed for the patient by default