Controlling User Access (Roles and Privileges)

Roles and Privileges are controlled through the administration screen, under the *Manage Users* section.

OpenMRS uses privileges and roles to control access to data within the system. Privileges define what can or cannot be done in the system (e.g., *Edit People* or *Add User*) while Roles are used to group privileges into more manageable grouping. To make the system easier to manage, roles can contain other roles as well as privileges. Roles inherit all privileges that exist within the child roles.

We'll use this example: you are working with several privileges related to patient data — e.g., *View Patient*, *Edit Patient*, and *Add Patient*. The *View Patient* privilege lets users look at patients in the system, the *Edit Patient* privilege lets users edit information about existing patients, and the *Add Patient* privilege allows users to create a completely new patient record within the system. Now imagine that you need to assign the proper rules to three people: Mary the Medical Student, Bob the Data Assistant, and Erica the Data Manager. You want medical students to be able to view patients, but not edit or add them. Data assistants should be able to not only view, but also edit patient data. And you want your data managers to be able to create new patients within your system. In the simplest method of assigning privileges you could individually assign the privileges:

User	Privilege(s)
Mary the Medical Student	View Patient
Bob the Data Assistant	View Patient Edit Patient
Erica the Data Manager	View Patient Edit Patient Add Patient

However, with this scheme, each time you add a new user to the system, you will have to make sure that you assign all of the proper privileges to the user. As your system grows, you may have dozens of privileges to manage for many of your users and the management will become very difficult.

Now, let's introduce roles. We will define a role for each type of user we need. In this example, we have a medical student, a data assistant, and a data manager. Each one of these could be a role.

Role	Privilege(s)
Medical Student	View Patient
Data Assistant	View Patient Edit Patient
Data Manager	View Patient Edit Patient Add Patient

This looks very similar to the first case, except we are now assigning privileges to roles instead of specific users. Why bother? The benefit comes as you add more users to the system. Now when you need to add a new user, say John the Medical Student, you need only assign a single role to that user.

User	Role(s)
Mary	Medical Student
Bob	Data Assistant
Erica	Data Manager
John	Medical Student

Now, by defining the main roles for users of your system and assigning users to those roles, you have a much easier system to manage and users will automatically inherit all privileges given to their role(s). Of course, some users will have multiple roles. You can also assign specific privileges to users in special cases. Now, let's take this process one step further. While it may not seem necessary in this simple example, as your system grows, you will likely end up with a large number of different roles. Very often, certain roles can be defined as a combination of other roles. In our example, a Data Manager oversees the Data Assistants and, therefore, should have all of their privileges *plus* some additional privileges. So, let's redesign our roles slightly to show how this might work.

Role	Inherit Privileges from Role(s)	Privilege(s)
Medical Student		View Patient
Data Assistant		View Patient Edit Patient
Data Manager	Data Assistant	Add Patient

You can see that the Data Manager role is more clearly defined as a *Data Assistant* with the extra ability to add patients to the system. In addition, if you should change or enhance the privileges of the *Data Assistant* role at any time in the future, the *Data Manager* will *automatically* adapt to those changes — for example, if you decided a month later to allow any *Data Assistant* to *Edit Encounters* (by adding the *Edit Encounters* privilege to the *Data Assistant* role), the *Data Manager* role would automatically gain the ability to edit encounters as well.

Common scenarios would be to define roles like *Provider* that is inherited by *Physician*, *Nurse*, *Clinical Officer*, etc. You can then control most of the privileges within the *Provider* role and those changes will effect all types of providers in the system. If you find that you have to go through multiple roles and edit them to make a change, then you could likely benefit from defining a role that applies to all of the roles and/or users you are editing and make define a new role to manage those privileges. For example, if you found that you were constantly editing roles like *Provider', _Data Assistant*, and *Caregiver* whenever you adjusted how patient data are allowed to be viewed in your system (i.e., affecting all users/roles that are allowed to view patient data), you might benefit from creating a new _Patient Data Viewer" role, assigning it to each of those other roles, and then managing the privileges in one place (under that new role).

Some privileges are built into the system and cannot be deleted. Other privileges may be added by modules. It is unlikely that you will be adding new privileges yourself, since privileges are only useful when they are understood and used by the system. On the other hand, you will definitely be creating new roles to fit your needs and will be managing privileges within those roles.

There are some special roles that are predefined within OpenMRS and cannot be deleted: *Anonymous*, *Authenticated*, *Provider*, and *System Develope_r. Any privileges granted to the _Anonymous* role will be available to people without logging into the system. Generally, *Anonymous* privileges are kept very restricted, since patient information might otherwise be compromised. Privileges granted to the *Authenticated* role are granted to anyone that logs into your system, no matter what other role(s) they might be assigned. Granting privileges to the *Authenticated* role is an easy way to grant privileges to all users of the system. *Provider* represents the most basic care provider and can serve as the basic role from which to build specialized providers (physicians, nurses, medical students, etc.). The *System Developer* role is automatically granted full access to the system and should only be granted to system administrators.

Super users (system administrators) are automatically granted all privileges in the system; therefore, you must be very careful to protect your system administrator password.

See the Privilege Helper module which lets you figure out what privileges are required to perform a particular task, and then helps you assign those privileges to roles.

What privileges are required for various actions?

- View Patients
  *__Manage Relationships__ `info | Manage Relationships privilege is *not` required after `build 5050`.
  - Patient Dashboard - View Demographics Section
  - Patient Dashboard - View Encounters Section
  - Patient Dashboard - View Forms Section
  - Patient Dashboard - View Graphs Section
  - Patient Dashboard - View Overview Section
    - Patient Overview - View Patient Actions (As from OpenMRS version 1.10)
    - Patient Overview - View Programs (As from OpenMRS version 1.10)
    - Patient Overview - View Relationships (As from OpenMRS version 1.10)
    - Patient Overview - View Allergies (As from OpenMRS version 1.10)
    - Patient Overview - View Problem List (As from OpenMRS version 1.10)
  - Patient Dashboard - View Patient Summary
  - Print Clinical Summary
  - View Clinical Summary
  - View Concept Classes
  - View Concept Datatypes
  - View Concepts
  - View Encounter Types
  - View Encounters
  - View Forms
  - View Locations
  - View Observations
  - View Patient Programs
  - View Patients
  - View People
  - View Person Attribute Types
  - View Programs
  - View Relationship Types
  - View Relationships

- View Encounters of a given Encounter Type
  - Go to the respective edit encounter type page and set the 'View Privilege' field to the desired privilege.
  - Assign the privilege you have selected above to the roles you wish to view the encounters of the edited encounter type above.

- Edit Encounters of a given Encounter Type
  - Go to the respective edit encounter type edit page and set the 'Edit Privilege' field to the desired privilege.
  - Assign the privilege you have selected above to the roles you wish to edit the encounters of the edited encounter type above.

- Do Data Exports
  - View Data Exports
  - Add Data Exports

    - Edit Data Exports
    - Delete Data Exports

- Work with Programs
    - Manage Programs - Required to add a new program, work flow, etc.
    - Edit Patient Programs - Required to allow a user to change the program that a patient is in.

- Tribes
    - View Tribes
    - Manage Tribes
      *__Edit Person Tribe - Assign a patient to a tribe. Only required if global property restrict_patient_attribute.tribe is set to true__ `info | The global property restrict_patient_attribute.tribe is *not` used in OpenMRS 1.5 and later.

What privileges are required at the API level?

This is 1.10 specific. For 1.9 and below, substitute "View" for "Get" (same as web level privileges). See TRUNK-3372

At the API level, the following privileges are required for each of the above view privileges:

- Get Concepts
- Get Concept Proposals
- Get Users
- Get Encounters
- Get Encounter Types
- Get Locations
- Get Observations
- Get Patients
- Get Patient Identifiers
- Get Patient Cohorts
- Get Orders
- Get Forms
- Get Identifier Types
- Get Concept Classes
- Get Concept Datatypes
- Get Privileges
- Get Roles
- Get Field Types
- Get Order Types
- Get Relationship Types
- Get Concept Sources
- Get Concept Map Types
- Get Concept Reference Terms
- Get Programs
- Get Patient Programs
- Get Global Properties
- Get Person Attribute Types
- Get People
- Get Relationships
- Get Database Changes
- Get Problems
- Get Allergies
- Get HL7 Source
- Get HL7 Inbound Queue
- Get HL7 Inbound Archive
- Get HL7 Inbound Exception
- Get Visit Types
- Get Visits
- Get Visit Attribute Types
- Get Location Attribute Types
- Get Providers
- Get Encounter Roles

To view All the available Priviledges, Go to **System Administration - Advanced Administration - Manage Priviledges** under "**Users**"

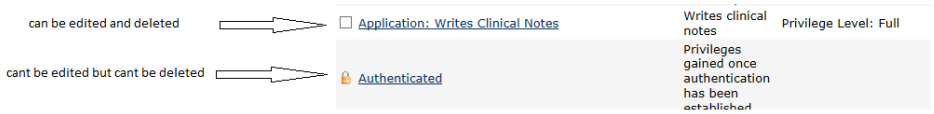
How to use priviledges and roles and avoid pitfalls

- when creating new roles .always take advantage of inheriting common priviledges from already existing roles.
  this makes the use of roles/priviledges very clear , easily understandable and flexible whenever a change is to be made regarding a certain role as illustrated below

Role	Inherit Privileges from Role(s)	Privilege(s)
Medical Student	View Patient	
Data Assistant	View Patient Edit Patient	
Data Manager	Data Assistant	Add Patient

- However much the **"add priviledge"** option exits under priviledge management on the admin page, the implementer is unlikely to be able to add a working new priviledge

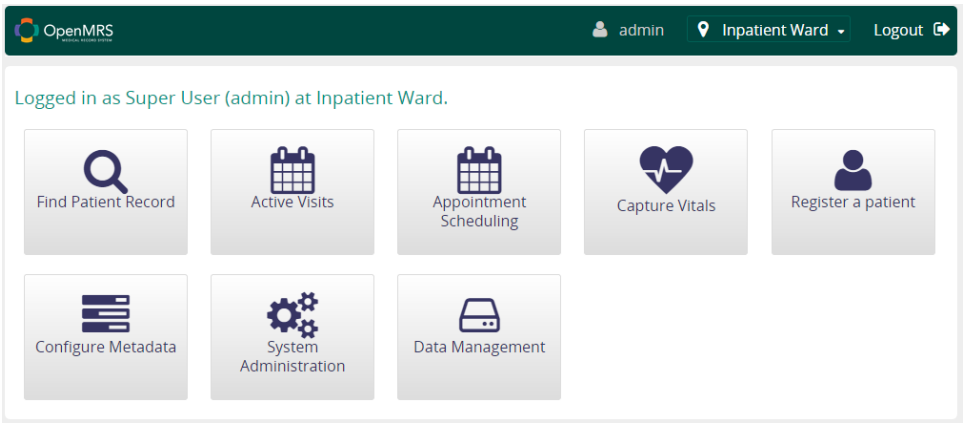
  .Since the priviledge must be understood and used by the system, its only a system devoloper who can define a new priviledge and add it to the list
- Limit the priviledges assigned to the "**Anonymus"** Role , as this will give acces to anyone who can acces the system without authentication and inreases the ability for a hacker to acces the system data ,etc
- Limit the priviledges assigned to the **"Authenticated",** as this will grant priviledges to any one who has just loggen in , and it will also compromise patient data confidentiality.
- Limit the Number of users granted the "**System Developer**" as it grants all the priviledges and roles to the user to acces any part of the system.
- Some in built Roles/Priviledges can not be deleted, but can be edited
  example of the roles that cant be deleted are **Anonymus, Authenticated, System developer** and **Provider**
  note All Role /Privildges that have a Locked Checkbox cant be deleted



Tips to note when using the openmrs Reference Application

- in the Reference Application , API level privileges are assigned to all roles automatically , and access is limited by assigning UI level privileges

  hence on the home page of the Reference Application, the apps displayed depends on the priviledges assigned to a given user.



- To be able to to register a patient, the logged in user needs to have an associated provider account, meaning no patient registration will be possible if there is no user that has a provider account and yes this includes super user. Go to **System Administration - Manage Accounts - Add New Account** to add a new user and provider. Remember to create both a user account (with appropriate privileges) and a provider account. or in case u created a new user account with out an associated provider account, Go to **System Administration - Advanced Administration - Manage Providers-Add provider ,** select the user without a provider account , and create a provider account for them

Trouble shooting.

- whenever u get the error "**java.lang.IllegalStateException: Can't handle users with multiple provider accounts**"..
  the cause of that, is creating duplicated provider account with the same user. Go to **System Administration - Advanced Administration - Manage Providers** and delete a duplicated provider account

Resources

Privilege Helper Module