

# Metadata Sharing Server Project

Assigned to: [Unknown User \(sriharsha\)](#)  
Mentor: [Unknown User \(raff\)](#)

 Metadata Sharing Server has been renamed to Metadata Repository.

## Background

The [Metadata Sharing Module](#) lets an administrator export a set of metadata (concepts, forms, locations, etc.) and publish it to a URL so that other OpenMRS implementations can import it and subscribe to future updates. In this model each OpenMRS server acts like a subscription server for clients. Clients download and import metadata from the given URL on the subscription server.

## Purpose

The goal of this project is to create a central repository of metadata where OpenMRS installations can publish their metadata. The Metadata Sharing Server will store metadata from different OpenMRS servers and expose them to clients. Clients will be able to search through metadata packages and choose some to import and subscribe. In this model the Metadata Sharing Server will act as a central subscription server for clients.

## Objectives

1. Implement the Metadata Sharing Server.
  - a. Allow for uploading metadata packages from a file and the Metadata Sharing Module.
  - b. List and search for metadata packages (based on indexed package headers).
  - c. Gather usage statistics and suggest the most popular packages first.
  - d. Authenticate OpenMRS installations while uploading packages and verify package publishing server information.
2. Modify the Metadata Sharing Module.
  - a. Allow for uploading metadata packages to the Metadata Sharing Server.
  - b. Add information about a publishing server (where it comes from) to a package.
  - c. Allow to import a selected version of a metadata package.
  - d. Search for metadata packages on the Metadata Sharing Server.

## Contacts

- [Unknown User \(raff\)](#)
- [Unknown User \(darius\)](#)

Do not hesitate to contact us, if you are interested in the project.

## Requirements

1. Technologies  
It should be possible to easily deploy the MDS server application on both Linux or Windows. The JVM is a preferred deployment environment, but it is not strictly required. Technologies used must be free and preferably open-source. The goal is to be able to deploy the MDS Server under mds.openmrs.org or any other server.
2. Authentication  
Each OpenMRS server has a unique [implementation ID](#) that should be used together with a password for the authentication process. We will only authenticate servers that want to upload metadata to the central repository. Importing should be possible for everyone without additional authentication. The authentication is thought as a way to confirm that the package comes from the given server so that someone who imports it can be sure that it is from a trusted source. We might allow for uploading a package without authentication, but it should be marked as not trusted. The authentication of subscribers is not planned in the first phase.
3. Versioning  
It should be possible to upload new versions of a package. A new version can be published only by someone who originally uploaded a package. It should be possible to import any version of the given package. Publishers should be able to unpublish any version of a package that was once published.

## Scenarios

### Using the web interface

#### Starting assumption

User with a web browser and an internet connection.

1. **Exporting**  
The user has a metadata package zip file. The user navigates the browser to a valid metadata sharing server address. From the main page the user selects the option to upload a metadata package from file. The user is prompted for and enters their implementation ID and exporting password. If the impID password combination is accepted proceed to the next step otherwise re-prompt up to three times before proceeding to an error page. The user is then given a file upload dialog to select the package they want to upload. The package header is parsed for the package information and it is displayed to the user for confirmation. If the package header is corrupted the user is directed to an error page. If there is a namespace collision the user is notified that the upload failed because a package with the same name was already on the server. If the upload is successful the user is shown a confirmation message. From all cases the user is then returned to the main page.

## 2. Importing

The user navigates the browser to a valid metadata sharing server address. From the main page the user selects the option to browse metadata packages available for download. The user is shown a tabbed list of available metadata packages. There are all, most popular, recent, and search tabs. These tabs contain tables showing the names of packages as well as their version number, labels, date uploaded, and number of times downloaded. Clicking on the package name will bring up a display including all the information parsed from the package header such as the package description. There will be a download link next to the package names which will download the corresponding package. When a package download is complete the user will be notified and returned to the main page.

## Using the metadata sharing module

### Starting assumption

User with a properly functioning OpenMRS implementation with an internet connection and the metadata sharing module installed. The metadata sharing module has been configured with the address of a valid metadata sharing server.

### 1. Exporting

The user accesses the metadata sharing module through the OpenMRS admin panel. The user chooses the option to export an existing metadata package to Metadata Sharing Server. An error is generated if the module has not been configured with the address of a valid metadata sharing server. The user is prompted to enter the exporting password connected to their implementation ID. If the implID/password combination is accepted proceed to the next step otherwise re-prompt up to three times before proceeding to an error page. The user selects a package to export from a list of those packages available locally. When this information is submitted if a name collision is detected the user shown an error and prompted to select a different package. When the user has successfully completed this step the package is uploaded. When the package has been uploaded and parsed the user is shown a confirmation message including the information parsed from the package header. Then the user is returned to the main metadata sharing module menu.

### 2. Importing

The user accesses the metadata sharing module through the OpenMRS admin panel. The user chooses to import a new metadata package and is given an option of sources: from file, from URL, or from Metadata Sharing Server. The user selects the Metadata Sharing Server source. The user is shown a tabbed list of available metadata packages. There are all, most popular, recent, and search tabs. These tabs contain tables showing the names of packages as well as their version number, labels, date uploaded, and number of times downloaded. Clicking on the package name will bring up a display including all the information parsed from the package header such as the package description. There will be a download link next to the package names. When the user has found a package to download, that package's corresponding download link is clicked. Once the download is complete a confirmation message is shown and the user proceeds to confirm the name, version, and description of the package, choose the trust level, and select options for the metadata as in normal operation of the metadata sharing module. After these steps are complete the user is returned to the metadata sharing module main menu.

## Project Timeline

### 1st week:

- 1.Create a basic Spring application.
- 2.Providing a form which takes the details like description, version of the package.
- 3.Creating a service layer with basic CRUD operations on packages.

### 2nd week:

- 4.Implementing an upload package method in the service layer.
- 5.Creating a page with Upload Package feature.Here when a user uploads a file, we will show a form pre-filled with details read from the file.(the form from the 2nd point )
- 6.Creating a page to list the packages.(preferably a paginated list).

### 3rd week:

- 7.Providing a feature to download the packages.
- 8.Providing a feature to search the packages.

### 4th week:

- 9.Providing the authentication and the administrator page to manage users(view,delete,reset password etc..)
- 10.Registration form for the users and providing an option for the users to manage their packages (view and delete).
- 11.Adding validations to the forms wherever its needed.
- 12.Testing the application and adding any features to complete the basic working application.

### 5th week and 6th week:

- 11.Adding the most downloaded packages feature.
- 12.Modifying the MDS module for updating the packages directly to the centralized server .
- 13.Adding information about the published server to a package and allow to import a selected version of metadata package.

## Q & A

Joe: How metadata sharing module gets configured with the metadata sharing server address?

Rafal: There is a settings page in the module.

Joe: How the admin gets/sets an exporting password tied to their implementation ID?

Rafal: One needs to generate it by accessing the MDS server with a browser (later we can add that to the module).

Joe: Allow private packages and select a password for those?

Rafal: Not in the first pass. We'd like our users to be open(-source).

## Resources

1. [Metadata Repository](#)
2. [Metadata Sharing Module](#)
3. [Implementation ID](#)
4. [Form Bank](#)
5. [OpenMRS Concept Collaborative \(OCC\)](#) - a central repository of concepts
6. [OCC Module](#)