

# Database Message Properties Module (Design Page)



This project was completed as part of GSoC 2011. You can see the final documentation of the module on our wiki: [Database Messages Module](#)

Primary mentor	<a href="#">Unknown User (sunbiz)</a>
Backup mentor	<a href="#">Unknown User (bwolfe)</a>
Assigned to	<a href="#">Unknown User (sreyakmit)</a>

## Background

Spring handles our internationalization of the web application. The files to do so are stored in .properties files. This is great for all the messages that we want to translate and that modules want to translate. However, what about instant translations that an openmrs installation might want to do? What about some temporary overrides of certain keys?

If the messages had the ability to be stored in the database, then the installation could easily change them. If the properties are left in the war file as they are now, then the installation will not want to edit them.

The MessageSourceService is already a registered service and calls are made through it. This module simply needs to hook into that and provide additional keys.

## Abstract

Currently all of our message properties translation files are stored in the messages.properties files in the webapp. Spring reads these and outputs for the user. The MessageSourceService has the ability to be overwritten and hence read the properties from the database.

Create a module that will register itself as a message provider and provide a set of management screens that will allow the user to add/edit/translate messages.

## Project Champions

[Unknown User \(akanter\)](#)

## Objectives

A module that will allow the user to add/modify message properties from within the database.

Admin screen to add/edit properties

## Extra Credit

Ability to compare message keys that are in the messages.properties file in openmrs war and not in a new locale in the db.

## Milestones

- Probably we'll finish the project by mid-term
- Mid-term goal: Atleast API ready by mid-term, so that you can change values in database and view them on the interface
- Final Time: Make the UI

## Resources

MessageSourceService, MessageSourceServiceImpl

### April 25 – May 23

1. Set up development environment
2. Go through the existing implementation of MessageSourceService, MessageSourceServiceImpl

### May 24 – June 13

1. Create a db for the module to hold the keys and the values in all possible values the application supports
2. Create a DAO/Hibernate classes to insert the values in the db
3. Create a service which would add/edit/delete and show all the values from the DB
4. Design the UI to add/edit the values from the DB

### June 13 -- July 1

- Hook the service with the MessageServiceSource

**July 1 – July 18**

- Add test cases for the module

**July 18 – July 25**

- Documentation of the project

**July 25 – August 1**

- Scrub the code

**August 1 – August 22**

**Make further improvements if required**