

# **CSCD01 Deliverable 4**

Team Number: 11

Team Name: Phantom Developers

Date: April 7th, 2020

# Table of Contents

<b>User Guide</b>	<b>3</b>
OpenMRS JIRA Issue: Support formulary status for drugs	3
Description	3
Requirements	3
User Guide Installation	4
Database Installation	4
Database Startup	4
OpenMRS Installation	4
<b>Code Documentation</b>	<b>7</b>
Design	7
Differences	8
Code Changes	8
OpenMRS-core	8
Implementation	8
Testing	8
OpenMRS RESTful Web Services	8
Implementation	8
<b>Acceptance Test Suite</b>	<b>9</b>
Save Formulary	9
Clients should be able to get a list of available formularies	10
Privileged clients should be able to manage the list of formularies and assignment of drugs to formularies	11
Clients should be able to check if a drug is on a particular formulary	11
Clients should be able to get the list of formularies for a particular drug	12
Clients should be able to search for a drug within a specified list of formularies	12
Clients should be able to ask the API for a list of all drugs within a given formulary	13
Clients should be able to delete given formulary	14
Clients should get an error when putting an invalid drugId or formularyId	14
<b>Unit Test Suite</b>	<b>15</b>
HibernateFormularyDAOTest	15
FormularyServiceTest	15

# User Guide

OpenMRS JIRA Issue: [Support formulary status for drugs](#)

## Description

This feature adds a “formulary” status to drugs, which are used to put drugs into a list. Each drug can be assigned to more than one formulary. Queries and management of formulary should also be supported. Formularies are lists of approved medication for prescription. They are used by hospitals and insurance companies to determine the eligibility of approving a claim, being prescribed, etc.

## Requirements

- Clients should be able to get a list of available formularies.
- Clients should be able to check if a drug is on a particular formulary.
- Clients should be able to get the list of formularies for a particular drug.
- Clients should be able to search for a drug within a specified list of formularies.
- Clients should be able to ask the API for a (paginated) list of all drugs within a given formulary.
- Clients should be able to manage the list of formularies and assignment of drugs to formularies.

# User Guide Installation

If you have any trouble during the installation, do not hesitate to email us.

## Database Installation

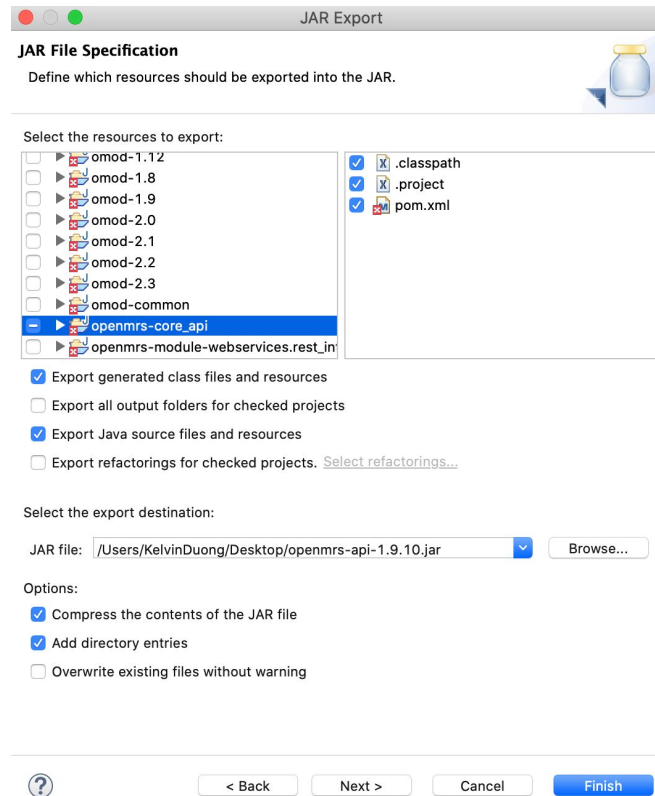
1. Download and install [MySQL 5.6](#) (MySQL version should below 5.6.7)
2. Use Setup type: Developer Default
3. Remember to set root password

## Database Startup

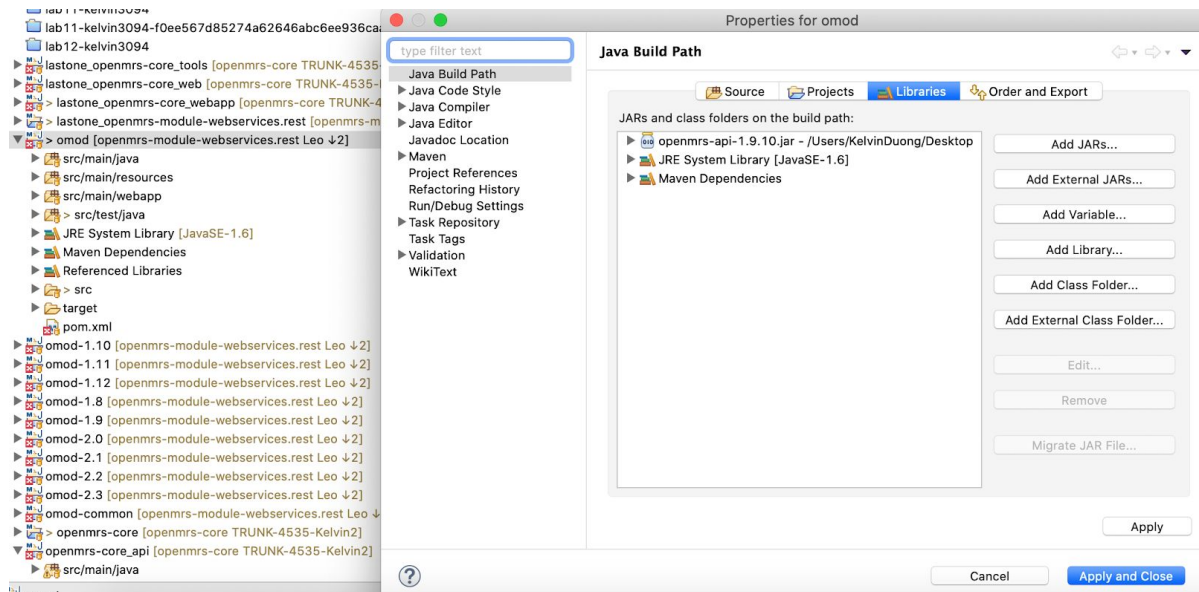
1. Open terminal
2. Go to %ProgramFiles%/MySQL/MySQL Server 5.6/bin
3. Type mysqld to start the database server

## OpenMRS Installation

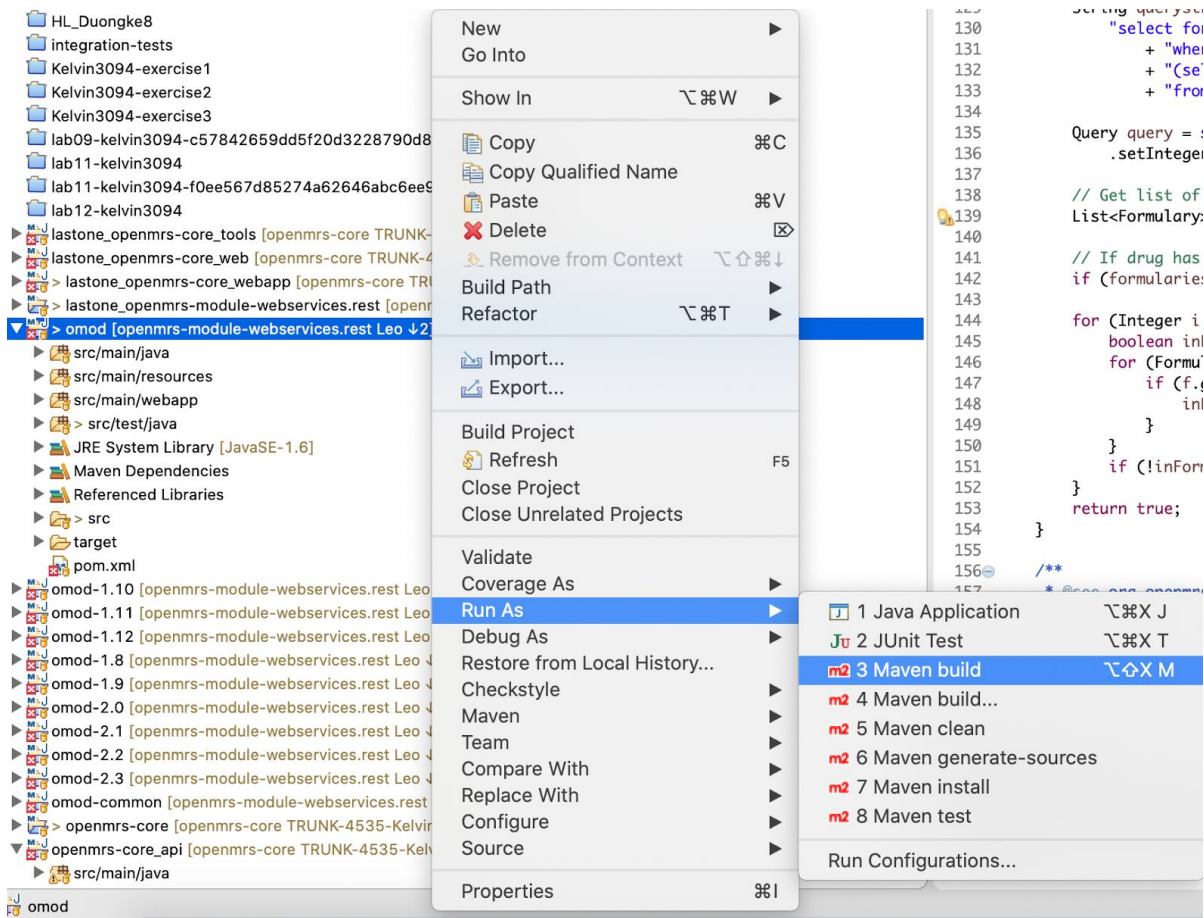
1. Install Java SE Development Kit 8.
2. Clone [OpenMRS Legacy UI](#) and install the OpenMRS Legacy UI first by following this [guide](#)
3. Clone [OpenMRS-Core](#)
4. In the terminal, cd into the openmrs-core folder and run “mvn clean install”. At the moment, the changes from the original OpenMRS repository may have a few tests that fail, so run “mvn clean install -DskipTests” instead if it fails
5. Then cd into the webapp folder and run “mvn jetty:run”
6. On your browser, go to “<http://localhost:8080/openmrs/>”
7. Login as admin with password Admin123
8. If this is your first time installing, you may get a prompt to initialize the database, so click the green arrow button.
9. Clone [Rest API](#) and open in Eclipse
10. In Eclipse, export the openmrs-core-api package from OpenMRS-Core as a JAR file, with ‘Export generated class files and resources’ and ‘Export Java source files and resource’ checked. Name it “openmrs-api-1.9.10.jar”



11. Add the “openmrs-api-1.9.10.jar” JAR file to the build path of the omod package



12. Build only the omod package with your IDE

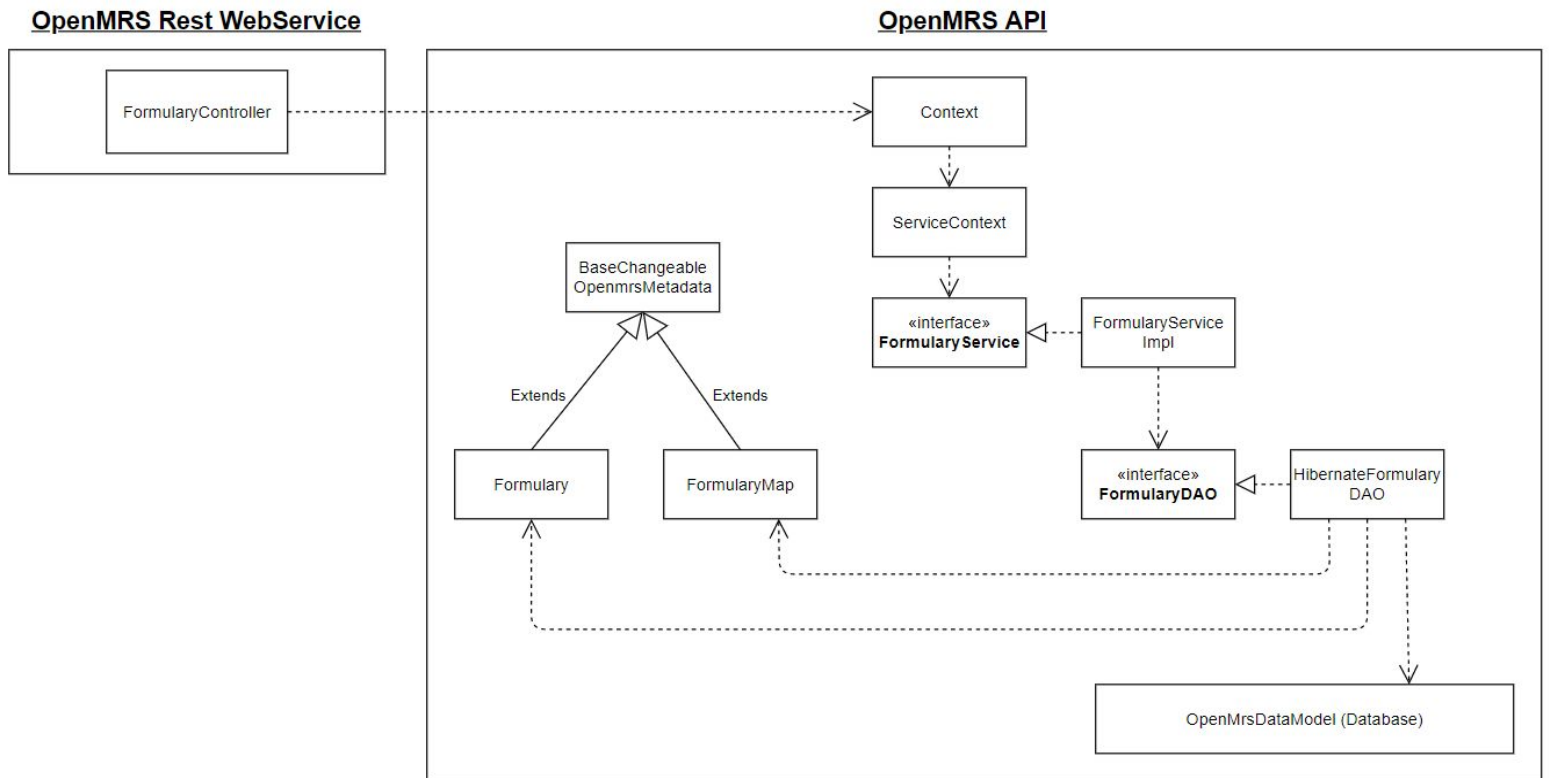


13. Stop openmrs using CTRL-C
14. Build the REST Api module using “mvn clean install” in the openmrs-module-webservices.rest folder. After you have built the project, the omod file should be located in openmrs-module-webservices.rest/omod/target/webservices.rest-2.29.0-SNAPSHOT.563510.omod (the last 6 digits in the file name may be different).
15. Copy webservices.rest-2.29.0-SNAPSHOT.563510.omod into the same module folder as the Legacy UI omod.
16. Restart OpenMRS with “mvn jetty:run” (see step 5)
17. Click Administration again. Under REST Web Services, click test. This page can be used for the acceptance tests below and should look like this.



# Code Documentation

## Design



### UML Diagram

To support the formularies, we first updated the database in liquibase-update-to-latest.xml to add the new tables `drug_formulary` and `drug_formulary_map`.

The `HibernateFormularyDAO` class allows us to execute queries on the new tables in the database.

We also added the `Formulary` and `FormularyMap` classes and the hibernate mapping files for the hibernate mapping which is used in the DAO. This allows for the hibernate database to properly convert rows in the MySQL database with the Java Objects in the source code.

The `FormularyService`, `FormularyDAO` is created as a SpringBean Object, that allows Spring to manage them and expose the endpoints to be accessible from the static `Context` class. Since these were newly created, we updated the `Context` class and the `applicationContext-service.xml` to support them.

In the REST module web service repository, we then added the `FormularyController` to handle the requests and give responses to the user by calling the OpenMRS-core API. This was created as we needed a way for the user to interact with the feature we implemented.

## Differences

We made some changes to our original design from deliverable 3. First, we added a hibernate mapping file that maps the `Formulary` class to the `drug_formulary` table in the database making it easier to insert, update, and get the relations from the database because it allows you to save the object and Hibernate will automatically do the database updates without having to write SQL. We also added a `FormularyMap` class, because that allows us to use Hibernate to map the `FormularyMap` class to the `drug_formulary_map` table, allowing the benefit of not querying the SQL database directly.

We also did not modify `liquibase-schema-only.xml` and instead modified `liquibase-update-to-latest.xml` to follow the OpenMRS guidelines on making database changes. We modified the rest module for OpenMRS and added a rest api to give front facing access for executing the customer acceptance tests.

## Code Changes

### [OpenMRS-core](#)

#### Implementation

Added [Formulary](#) and [FormularyMap](#) object classes

Added [FormularyService](#) interface

Added [FormularyServiceImpl](#) class for extra business logic to create and delete objects

Added [FormularyDAO](#) interface

Added [HibernateFormularyDAO](#) class for main business logic for interacting with the database

Modified [liquibase-update-to-latest.xml](#) to update the database

Added [Formulary.hbm.xml](#) and [FormularyMap.hbm.xml](#) to add mapping to the new tables

Updated [applicationContext-service.xml](#) to add connecting to new classes and service

Updated [Context](#) class to get the formulary service

#### Testing

Added [FormularyServiceTest](#) junit test class

Added [HibernateFormularyDAOTest](#) junit test class

Updated [standardTestDataset.xml](#) to add data to the mock database

### [OpenMRS RESTful Web Services](#)

#### Implementation

Added [FormularyController](#) class to support user requests



# Acceptance Test Suite

In a Terminal, enter the following commands to make a request. The uuid's and id's may not be exactly the same.

## Save Formulary

Description	Clients should be able to create new formulary
Request	<pre>curl -i -u Admin:Admin123 -X POST http://localhost:8080/openmrs/ws/rest/v1/formulary/saveFormulary -d '{ "name" : "formulary name", "description" : "formulary desc", "doses" : 10 }'</pre>
Expected Result	<pre>{   "uuid" : "11f2f381-c93e-4867-9a1d-6eed19bc661c",   "name" : "formulary name",   "description" : "formulary desc",   "retired" : false,   "formularyId" : 1,   "doses" : 10,   "id" : 1 }</pre>

Description	Clients should be able to create new formulary and id should be auto incrementing
Request	<pre>curl -i -u Admin:Admin123 -X POST http://localhost:8080/openmrs/ws/rest/v1/formulary/saveFormulary -d '{ "name" : "another formulary name", "description" : "new formulary desc", "clinicalInformation" : "important information", "doses" : 200 }'</pre>
Expected Result	<pre>{   "uuid" : "4a90f649-495c-4f2c-b269-3f65caea4017",   "name" : "another formulary name",   "description" : "new formulary desc",   "retired" : false,   "formularyId" : 2,   "clinicalInformation" : "important information",   "doses" : 200,   "id" : 2 }</pre>

Clients should be able to get a list of available formularies

Description	Clients should be able to get a list of formularies
Request	<code>curl -i -u Admin:Admin123 http://localhost:8080/openmrs/ws/rest/v1/formulary/getAllFormularies</code>
Expected Result	<pre>[ {   "uuid" : "11f2f381-c93e-4867-9a1d-6eed19bc661c",   "name" : "formulary name",   "description" : "formulary desc",   "retired" : false,   "formularyId" : 1,   "doses" : 10,   "id" : 1 }, {   "uuid" : "4a90f649-495c-4f2c-b269-3f65caea4017",   "name" : "another formulary name",   "description" : "new formulary desc",   "retired" : false,   "formularyId" : 2,   "clinicalInformation" : "important information",   "doses" : 200,   "id" : 2 }]</pre>

Description	Clients should be able to get a list of formularies with specific page number and limit
Request	<code>curl -i -u Admin:Admin123 http://localhost:8080/openmrs/ws/rest/v1/formulary/getAllFormularies?limit=1&amp;page=0</code>
Expected Result	<pre>[ {   "uuid" : "11f2f381-c93e-4867-9a1d-6eed19bc661c",   "name" : "formulary name",   "description" : "formulary desc",   "retired" : false,   "formularyId" : 1,   "doses" : 10,   "id" : 1 }]</pre>

Privileged clients should be able to manage the list of formularies and assignment of drugs to formularies

Description	Clients should be able to insert a drug to a particular formulary
Request	<code>curl -i -u Admin:Admin123 -X POST http://localhost:8080/openmrs/ws/rest/v1/formulary/insertDrugToFormulary?drugId=2&amp;formularyId=1</code>
Expected Result	Success

Description	Clients should be able to insert a drug to a particular formulary
Request	<code>curl -i -u Admin:Admin123 -X POST http://localhost:8080/openmrs/ws/rest/v1/formulary/insertDrugToFormulary?drugId=3&amp;formularyId=1</code>
Expected Result	Success

Clients should be able to check if a drug is on a particular formulary

Description	Clients should be able to check if a drug is in a particular formulary
Request	<code>curl -i -u Admin:Admin123 http://localhost:8080/openmrs/ws/rest/v1/formulary/isDrugInFormulary?drugId=2&amp;formularyId=1</code>
Expected Result	true

Description	Clients should be able to check if a drug is in a particular formulary
Request	<code>curl -i -u Admin:Admin123 http://localhost:8080/openmrs/ws/rest/v1/formulary/isDrugInFormulary?drugId=2&amp;formularyId=2</code>
Expected Result	false

Clients should be able to get the list of formularies for a particular drug

Description	Clients should be able to get a list of formularies for a particular drug
Request	<code>curl -i -u Admin:Admin123 http://localhost:8080/openmrs/ws/rest/v1/formulary/getFormulariesFromDrug?drugId=2</code>
Expected Result	<pre>[ {   "uuid" : "11f2f381-c93e-4867-9a1d-6eed19bc661c",   "name" : "formulary name",   "description" : "formulary desc",   "retired" : false,   "formularyId" : 1,   "doses" : 10,   "id" : 1 }]</pre>

Description	Clients should be getting an empty list if no formularies for a specific drug
Request	<code>curl -i -u Admin:Admin123 http://localhost:8080/openmrs/ws/rest/v1/formulary/getFormulariesFromDrug?drugId=4</code>
Expected Result	<pre>[]</pre>

Clients should be able to search for a drug within a specified list of formularies

Description	Clients should be able to search for a drug within a specified list of formularies
Request	<code>curl -i -u Admin:Admin123 http://localhost:8080/openmrs/ws/rest/v1/formulary/isDrugInFormularies?formularyId=1&amp; formularyId=2&amp;drugId=2</code>
Expected Result	true

Description	Should be able to return false if a drug is not within a specified list of formularies
Request	<code>curl -i -u Admin:Admin123 http://localhost:8080/openmrs/ws/rest/v1/formulary/isDrugInFormularies?formularyId=1&amp; formularyId=2&amp;formularyId=3&amp;drugId=5</code>
Expected Result	false

Clients should be able to ask the API for a list of all drugs within a given formulary

Description	Clients should be able to get a list of drugs for a specific formulary
Request	Request: curl -i -u Admin:Admin123 http://localhost:8080/openmrs/ws/rest/v1/formulary/getDrugsFromFormulary?formularyId=1
Expected Result	[ { "uuid" : "a5d200e4-744b-11ea-9de3-00ff3772e317", "name": "Triomune-30", "combination" : "1", "drugId" : "2", "conceptId" : "792" "id" : "2" }, { "uuid" : "a5d201f5-744b-11ea-9de3-00ff3772e317", "name": "Triomune-40", "combination" : "1", "drugId" : "3", "conceptId" : "792" "id" : "3" }]

Description	Clients should get an empty list if no drugs for a specific formulary
Request	curl -i -u Admin:Admin123 http://localhost:8080/openmrs/ws/rest/v1/formulary/getDrugsFromFormulary?formularyId=3
Expected Result	[ ]

Clients should be able to delete given formulary

Description	Clients should delete a specific formulary and all the formulary drug mapping of the formulary
Request	<code>curl -i -u Admin:Admin123 -X DELETE http://localhost:8080/openmrs/ws/rest/v1/formulary/deleteFormulary?formularyId=1</code>
Expected Result	Success

Description	Clients should be able to remove a specific drug for the given formulary
Request	<code>curl -i -u Admin:Admin123 -X DELETE http://localhost:8080/openmrs/ws/rest/v1/formulary/removeDrugFromFormulary?formularyId=2&amp;drugId=2</code>
Expected Result	Success

Clients should get an error when putting an invalid drugId or formularyId

Description	Clients get an error when the drugId is not in the database
Request	<code>curl -i -u Admin:Admin123 http://localhost:8080/openmrs/ws/rest/v1/formulary/isDrugInFormularies?formularyId=1&amp;formularyId=2&amp;drugId=99</code>
Expected Result	Status Code: 400. Invalid drug id

Description	Clients get an error when the formularyId is not in the database
Request	<code>curl -i -u Admin:Admin123 http://localhost:8080/openmrs/ws/rest/v1/formulary/isDrugInFormularies?formularyId=1&amp;formularyId=2&amp;formularyId=99&amp;drugId=5</code>
Expected Result	Status Code: 400. Invalid formularyId

# Unit Test Suite

*The name of the tests should be self explanatory, and they can be run as a JUnit test on Eclipse*

## HibernateFormularyDAOTest

Tests to verify methods in the DAO

saveFormulary\_shouldSaveFormulary  
getFormulary\_shouldGetFormulary  
getFormulary\_shouldReturnNullIfFormularyDoesNotExist  
getAllFormularies\_shouldGetAllFormularies  
getAllFormularies\_shouldGetFormularyGivenPageAndLimit  
getDrugsFromFormulary\_shouldGetDrugs  
getDrugsFromFormulary\_shouldThrowExceptionWhenInvalidFormularyId  
isDrugInFormulary\_shouldDrugBeInFormulary  
isDrugInFormulary\_shouldDrugNotBeInFormulary  
isDrugInFormulary\_shouldDrugInFormularyFormularyNotExist  
isDrugInFormularies\_shouldBeTrueIfDrugsInFormularies  
isDrugInFormularies\_shouldBeFalseIfDrugsInNoFormularies  
isDrugInFormularies\_shouldBeTrueIfDrugsInSomeFormularies  
isDrugInFormularies\_shouldThrowExceptionForInvalidDrugId  
isDrugInFormularies\_shouldThrowExceptionForInvalidFormularyId  
getFormulariesFromDrug\_shouldGetFormularies  
insertDrugToFormulary\_shouldInsertDrug  
deleteFormulary\_shouldDeleteFormularyFromDatabase  
deleteFormulary\_shouldCascadeDelete  
deleteFormulary\_shouldGiveErrorIfInvalidFormularyId  
removeDrugFromFormulary\_shouldRemoveDrugFromFormulary  
removeDrugFromFormulary\_shouldGiveErrorIfInvalidDrugId

## FormularyServiceTest

Tests to verify methods in the DAO

shouldSaveGetDeleteFormulary  
getAllFormularies\_shouldGetAllFormularies  
shouldSaveGetDeleteFormularyMap