

How to Set Up A Development Environment for the On the Open Community Information Sharing System (OpenCISS/OpenHMIS)

updated: 7/25/2010 (if you have access rights, please feel free to make updates/edits; the public can comment on this document as well)

[Introduction](#)

[Setting Up Eclipse Workbench](#)

[Setting Up Google Eclipse Plugins](#)

[Configuring the Eclipse Project](#)

[Getting the Code](#)

[Configuring App Engine](#)

[Configuring Google Web Toolkit](#)

[Java Project Facets](#)

[JAXB](#)

[Configuring the JAXB facet](#)

[JPA](#)

[Configuring the JPA Facet](#)

[JAX-RS \(Jersey\)](#)

[Setting up a local MySQL Development Database](#)

[Configuring Access to the Google Cloud SQL App Engine Instance](#)

[Using the Eclipse Data Management View](#)

[Deploying the Code](#)

[Accessing the Google Cloud SQL Database Console](#)

[Accessing the Google App Engine Project Dashboard](#)

[Code Overview](#)

[Directory Orientation](#)

Introduction

Development on the OpenCISS requires a series of SDKs and Eclipse plugins. Equivalent tools may be used as well, but these are recommended.

Setting Up Eclipse Workbench

- Install the Java EE Developer's workbench for Eclipse from: <http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/junor>

Setting Up Google Eclipse Plugins

Once Eclipse is installed, install the latest Google Plugins For Eclipse (GPE): <https://developers.google.com/eclipse/>

Use the GPE installation wizard to also install the latest:

- Google App Engine SDK
- Google Web Toolkit

Configuring the Eclipse Project

Create a new Google “Web Application” project named “openciss”.

Create a new “source folder” in the new Google Web Application project, and place the code checked out from the /src folder from [Getting the Code](#) into this new source folder

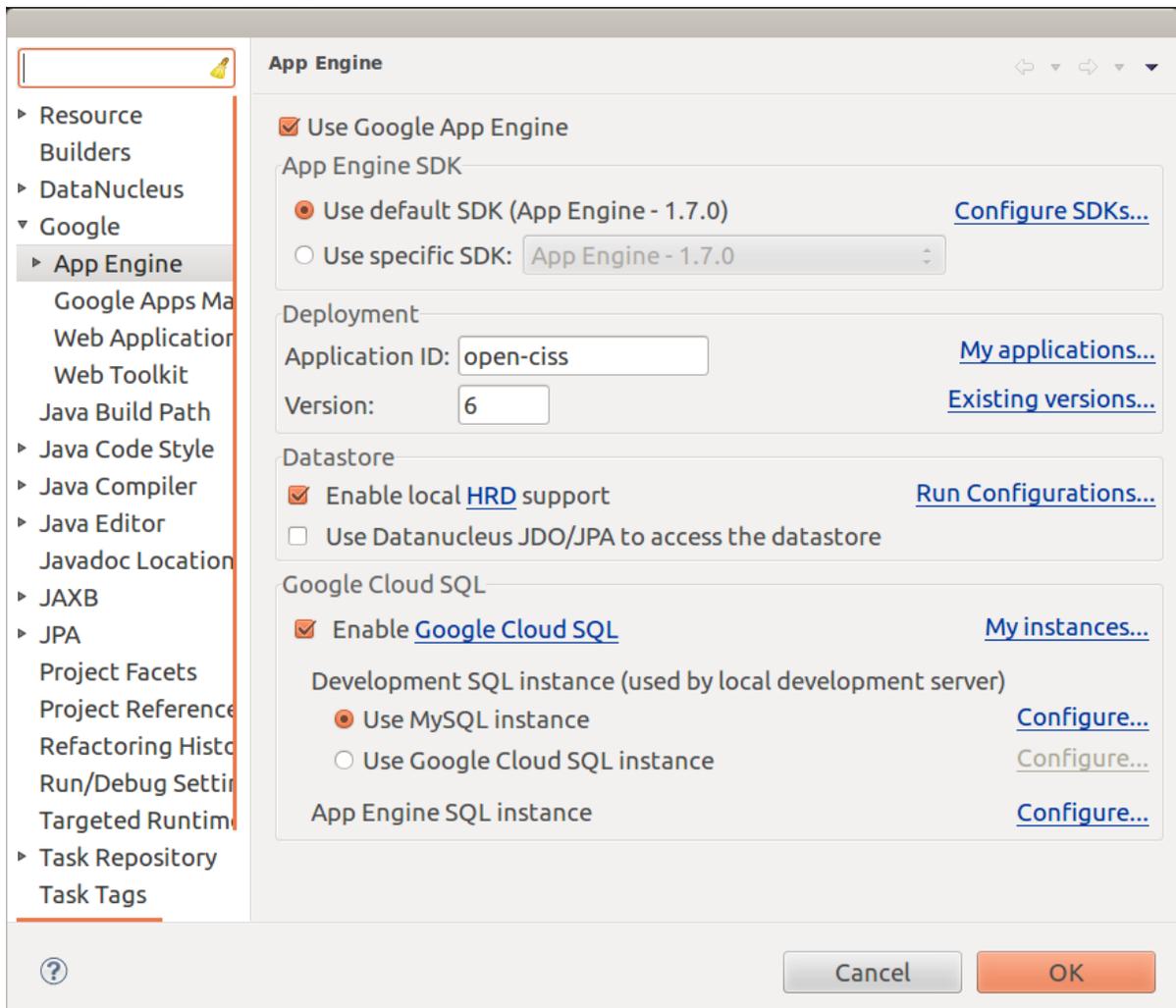
Getting the Code

Check out the code hosted on the Mercurial repository at:

<https://code.google.com/p/open-community-information-sharing-system/source/checkout>

The project uses a standard Java web container directory structure.

Configuring App Engine



Configuring Google Web Toolkit

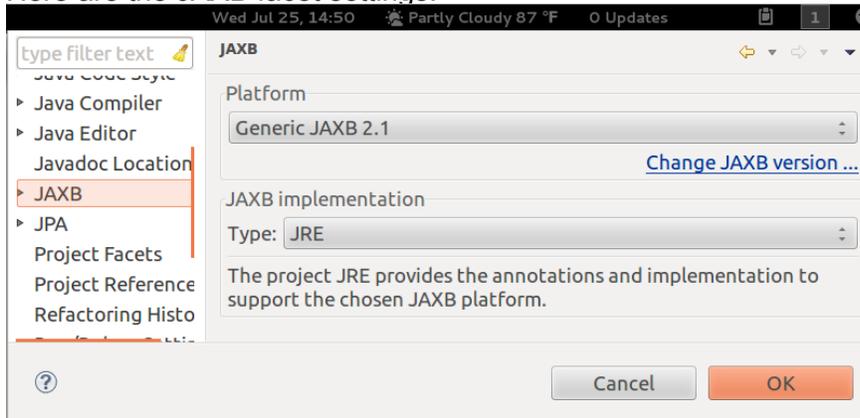
Java Project Facets

JAXB

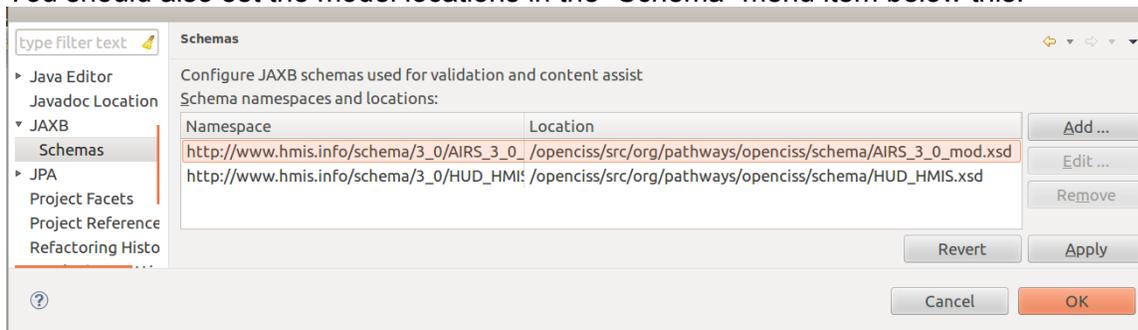
JAXB lets us make and read XML and JSON.

Configuring the JAXB facet

Here are the JAXB facet settings:



You should also set the model locations in the “Schema” menu item below this:

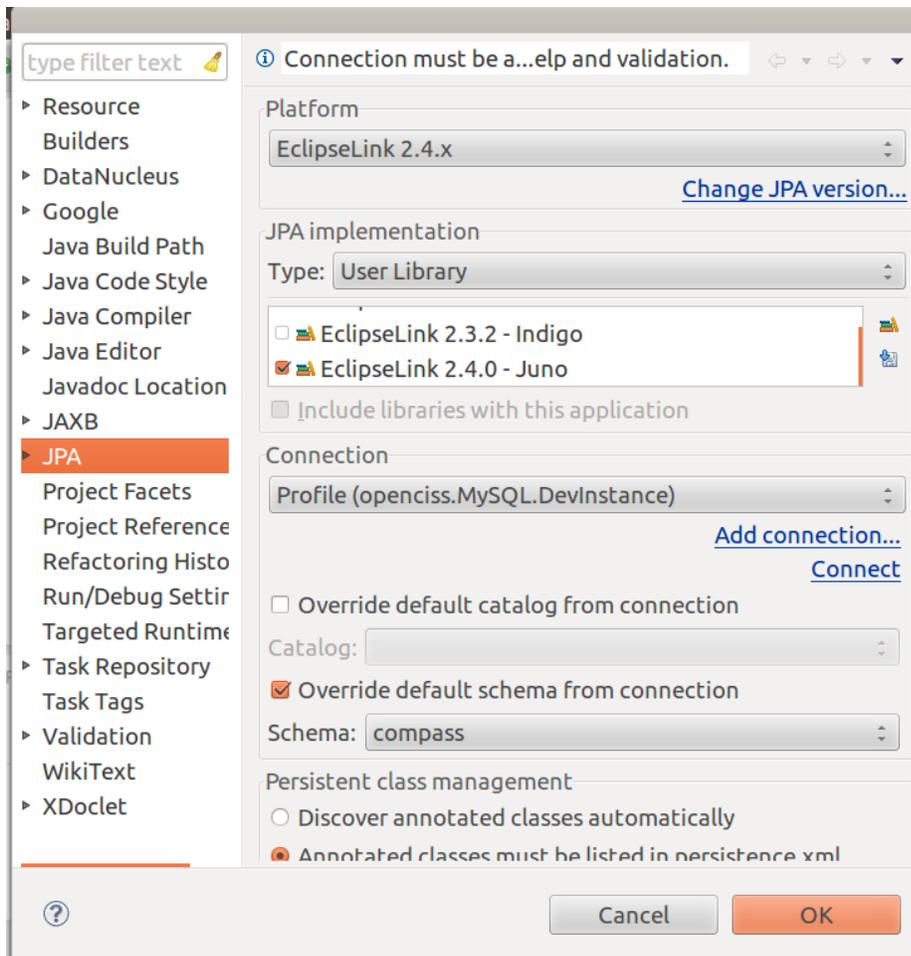


JPA

JPA lets Java talk to the database without having to use cumbersome SQL.

Configuring the JPA Facet

Screenshot from Project->Properties:



JAX-RS (Jersey)

Jersey lets us create RESTful urls like /client/add and /shelter/search.

The JAX-RS project facet is not used at this time, since we can't get the facet to work. Perhaps this is because JAX-RS libraries were already in use in the code before attempting to configure the facet. Feedback from other developers requested on this, but JAX-RS is working fine without the facet.

Setting up a local MySQL Development Database

add settings for the connection dialog

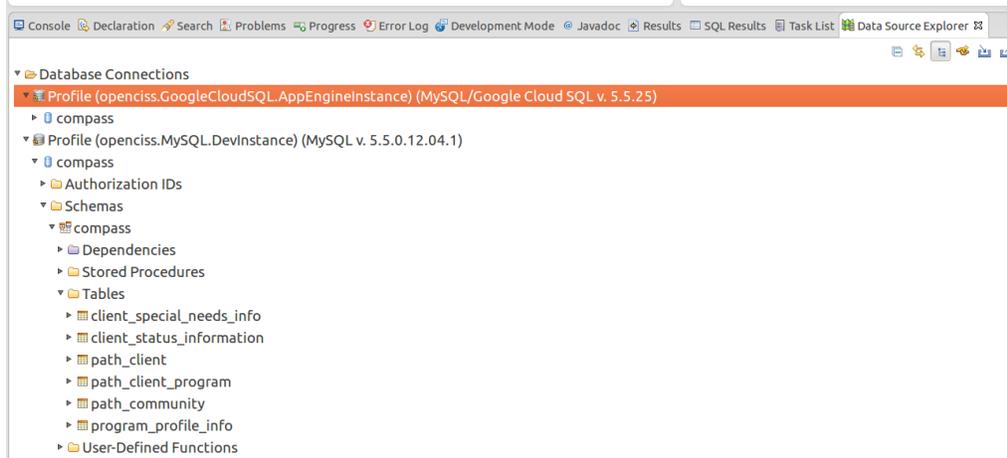
Configuring Access to the Google Cloud SQL App Engine Instance

add settings for the connection dialog

Using the Eclipse Data Management View

Allows you to see/edit table data, and run SQL scripts.

Data Source Explorer view:



Deploying the Code

“Project(right click on name)->Google->Deploy to App Engine”. Make sure you increment the app version number, or it will overwrite the last one you uploaded (which you may in fact want to do).

Accessing the Google Cloud SQL Database Console

You'll need permissions to access this at pcni.org. You can perform most database functions you'll need with the Eclipse Database Development view.

Accessing the Google App Engine Project Dashboard

You'll need permissions to access this at pcni.org. The dashboard allows you to run a deployed version. Even if you deploy the version, it will not run until it is made the default version within the dashboard.

Code Overview

Directory Orientation