

Developer Training 001

Overview

Resources for training.

Topics Covered

- Obtaining and building the code
- Prerequisites installed (do not show this)
- Clone Github project
- Build project at top level using mvn command line

Video

- [Developer Training 001 Video](#)

Script

Following is a script for the attached video.

- Preamble
 - Welcome to the IHTSDO map tool video tutorials. This developer training video covers obtaining and building the code.

Prerequisites

- Git client
- Java 7 installed
- Mysql 5.5 or 5.6 installed
- Maven 3.2.+ installed
- Tomcat 7 installed and configured
- Eclipse
 - Use version 4.4 (Luna) - J2EE edition
 - m2e (the maven integration for eclipse - download from the market place)
 - m2e-egit connector
 - eGit - <http://eclipse.github.com/>
 - Jautodoc - import configuration from [Code and XML Formatting in Eclipse](#)
 - ObjectAid UML Diagram tool for class hierarchy visualization. Installed via Help->Install New Software. Instructions at <http://www.objectaid.net/installation>
 - FindBugs (static code analysis)
 - AngularJS - see <https://github.com/angelozerr/angularjs-eclipse/wiki/Getting-Started>

As this project is organized around Maven and may make use of resources on IHTSDO Nexus, properly configure your [Settings.xml Page](#).

Details

Step 1 - Create directories

Setup for Windows

- Create a directory to hold your data files (e.g. c:/mapping/data)
- Create a directory to hold your config files (e.g. c:/mapping/config)
- Create a directory to hold the code (e.g. c:/workspace/OTF-Mapping-Service)
- Make sure the "mvn" executable for your local maven installation is in the path
 - On Windows this means adding to the PATH variable so that it runs in a "cmd" shell without fully qualified path.

Step 2 - Clone repositories

Clone the [Github repository](#) to the directory created to hold the code.

Clone the mapping-service-data project (e.g. `ssh://<yourUsername>@csfe.aceworkspace.net:29418/mapping`) to the directory created to hold your data files..

- In this context "yourUsername" is the CollabNet user you use to access the "The IHTSDO Open Tooling Framework Project"

Step 3 - Build project

Build all project modules with "mvn clean install" at the top level - either through Eclipse or via the command line.

- NOTE: this uses the standard "dev-windows" configuration. To use a different configuration artifact pass the following three parameters:
 - -Dconfig.groupId=...
 - -Dconfig.artifactId=...
 - -Dconfig.version=...
- NOTE: the specified configuration artifact must either have been built locally or be in a repository accessible based on the settings.xml file
- For most dev deployments, the default build is fine because the only setting really used for the rest/webapp packages is the "base.url" which by default is set to <http://localhost:8080/mapping-rest>.
- For a UAT or production deployment, you would want to use a different setting. For example, see [Deploy Instructions](#), the prod deployment instructions.

Step 4 - Setup Configuration

Choose a "dev-windows", "uat", or "prod" config project target and unzip it into the directory created to hold your config files.

- config/dev-windows/target/mapping-config-dev-windows.*.zip
- config/prod/target/mapping-config-prod.*.zip
- config/uat/target/mapping-config-uat.*.zip

Step 5 - Edit configuration

Edit the "config.properties" file in your config files directory to set correctly for your environment. In particular, edit these:

- javax.persistence.jdbc.url
- javax.persistence.jdbc.user
- javax.persistence.jdbc.password
- hibernate.search.default.indexBase (recommend choosing something in your data dir, e.g. c:/mapping/data/indexes or ~/indexes)
- ihtsdo.security.activated (set to false for dev environment)
- mail.smtp.to (list for automated system emails)
- send.notification.recipients (list for automated tools, like reports generation and database qa)

References/Links

- n/a