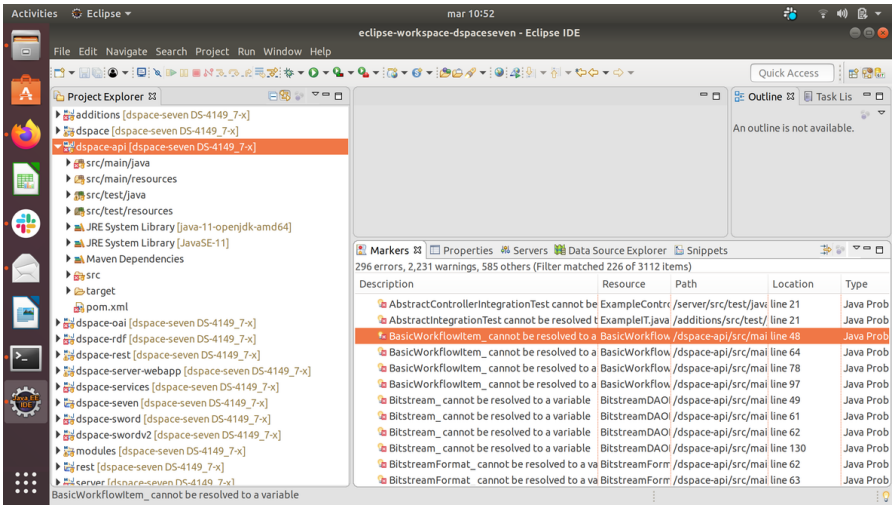
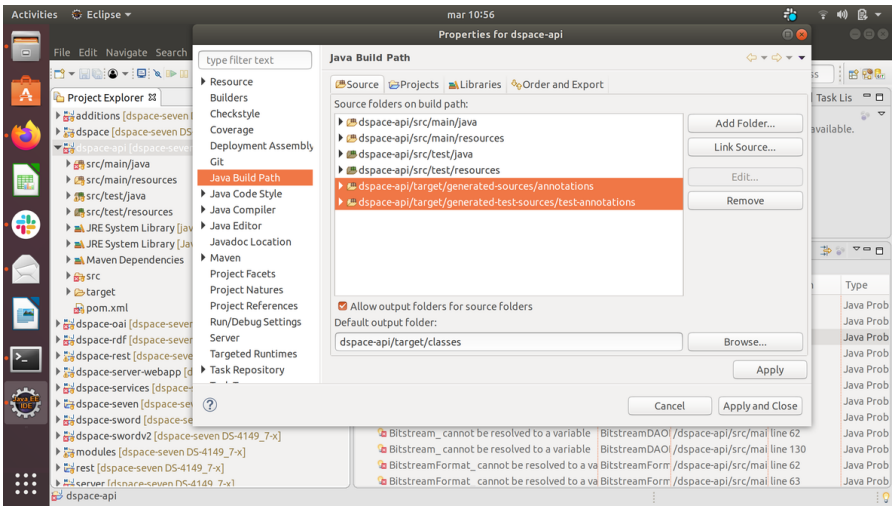
DSpace 7 development with Eclipse

DSpace(-CRIS) 7 requires :

- JAVA 11
- Tomcat 9
- SOLR 8.11 installed as service
    - NB: with SELinux the startup is blocked by policy. Create a custom policy and disable SELinux with `setenforce 0`.
- Eclipse IDE for Enterprise Java Developers. Version: 2018-12 (4.10.0) available here blocked URL newest versions don't work
    - Install the CheckStyle Eclipse plugin: Eclipse Checkstyle Plugin
        - the version used should be the same one used by the maven plugin in dspace (currently 8.30), but since 8.30 is no longer available 8.41.1 should be installed using the following steps:
            1. go to help > install new software > add
                - sonarlint (checkstyle dependency)
                    - https://eclipse-uc.sonarlint.org/
                - checkstyle
                    - https://checkstyle.org/eclipse-cs-update-site/
            2. after adding both sources select in the "Work with" dropdown checkstyle's url
            3. uncheck "show only the latest versions of available software"
            4. in the result box open the "checkstyle" dropdown and scroll down until you find "Eclipse Checkstyle Plug-in" version 8.41.1.202105102159
            5. select the entry just found and continue with the installation by clicking next
        - Configure checkstyle using the following guide: Code Style Guide - DSpace - LYRASIS Wiki
            - creating the configuration for each project can be skipped by going to window > preferences > checkstyle and adding the configuration there (and optionally setting it as default), making sure to put `/[nome cartella parent progetto(nel mio caso dspace-cris-7)]/checkstyle.xml` in the location field
    - Import the parent pom directly from eclipse and select all projects.
    - Close the rest and dspace-rest projects.
    - Once the maven projects are imported into the workspace there will be a series of Java Problems in the Markers tab on the "dspace-api" project due to access to private attributes via autogenerated classes. Since these are fake errors they only create "noise". To fix this, add to the build path "target/generated-sources/annotations" and "target/generated-test-sources/test-annotations". The second folder should be available only after externally launching the command "mvn test".





NB: the result is to solve the pre-existing problems of access to variables, will remain some compilation errors on the project "server" and the project "additions" that we decided not to solve because they lead to make an advanced configuration of the buildpath between the project "dspace-server-webapp" and project "server" and project "additions".

Code checkout and set-up (applies on both Eclipse and IntelliJ Idea set up)

Checkout dspace(-cris) 7 (branch might vary)

```
cd /home/user/git 2git clone 3git checkout dspace-cris-7
```

configure local environment for dspace(-cris) installation (solr, postgres, dspace.dir)

```
cp dspace/config/local.cfg.EXAMPLE dspace/config/local.cfg 2vi dspace/config/local.cfg
```

this implies

```
dspace.dir=/d/install/dspace7 2db.url = jdbc:postgresql://localhost:5432/dspace7 3solr.server = http://localhost:8983/solr
```

if more dspace7 installation handling is needed, single solr cores endpoints should be overwritten

```
discovery.search.server = ${solr.server}/search 2oai.solr.url=${solr.server}/oai 3solr-statistics.server = ${solr.server}/statistics 4solr.authority.server=${solr.server}/authority
```

for dspace-cris a multicore prefix property is available

```
solr.multicorePrefix
```

copy content of {dspace-install-dir}/solr to /var/solr/data and restart Solr

Install DSpace https://wiki.lyrasis.org/display/DSDOC7x/Installing+DSpace#InstallingDSpace-BackendInstallation (in short)

```
mvn package

cd dspace/target/dspace-installer

ant update
```

(Optional) install file of geolite (https://wiki.lyrasis.org/display/DSDOC6x/SOLR+Statistics#SOLRStatistics-ManagingtheGeoLiteDatabaseFile - https://jira.lyrasis.org/browse/DS-4409)

1. Create an account MaxMind
2. Set a password and create a license key
3. Download il GeoIP Update program o creando un direct download script
4. locate downloaded files (probably under /usr/share/GeoIP)
5. Confiugre property usage-statistics.dbfile in file {dspace.install-dir}/config/local.cfg with path of file GeoLite2-City.mmdb (such file migh also be moved to {dspace-install-dir}/config directory)

Run in Eclipse

In eclipse run web server project (not dspace-server-webapp) which will include via maven overlay all dspace needed modules (REST, OAI, SWORD, RDF,…) but legacy api REST, to be built and run on request (not mandatory)

Add to tomcat server “eclipse” a context.xml

```
    <?xml version="1.0" encoding="UTF-8"?>
<Context>
  <WatchedResource>WEB-INF/web.xml</WatchedResource>
  <WatchedResource>WEB-INF/tomcat-web.xml</WatchedResource>
  <WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>
  <Parameter name="dspace.dir"  value="/d/install/dspace7"  override="false" />
  <Parameter name="dspace-config" value="/d/install/dspace7/config/dspace.cfg"  override="false" />
</Context>
```

Java Scripts

Java scripts can be run from eclipse, specifying

- main class
    - org.dspace.app.launcher.ScriptLauncher
- program argument
    - script alias as defined in launcher.xml file, or any class with main method, using command “dsrun”
        - oai import -c
        - dsrun org.dspace.xoai.app.XOAI import -c
- vm argument

  ```
  1-ea -Ddspace.dir=/install/dspace7
  ```

Unit tests can be run specifying test as class to be executed in a JUnit launcher, and as vm arguments (target directory generated by following maven command is used, in case of eclipse clean it needs to be recreated, or, alternatively, read following paragraph to save test.zip environment in a separate folder)

```
-ea -Ddspace.dir=/home/openmind/git/DSpace7/dspace-server-webapp/target/testing/dspace -Dsolr.install.dir=/home
/openmind/git/DSpace7/dspace-server-webapp/target/testing/dspace/solr
```

before running tests, they should have been run at least once from command line with

```
1mvn test -Dtest=org.dspace.app.rest.AuthorityRestRepositoryIT -Dmaven.test.skip=false -DskipUnitTests=false -
DskipIntegrationTests=false -DfailIfNoTests=false
```

test execution will create a .zip file that can be saved and unzipped in a dedicated folder, such folder can be specified as vm argument while running tests from Eclipse. Ensure that testing folder is always up to date with latest changes on source code and at configurations (dspace.cfg, spring, etc. both general and specifics of test environment “dspace-api/src/test/data/dspaceFolder”)

VM Arguments needs to be enriched with additional settings, if needed, for example -Dsubmission.lookup.ads.apikey, that cannot be stored in test local.cfg because of security or privacy related problems , for example API-KEY. Those information, typically, are recovered by CI tools during deployment process (Travis/Bitbucket pipelines) from environment variables and injected into local.cfg during deployment.