Zero-Config Installation Notes

for Materials Data Curation Project

These notes are for requirement 1.A7 of the Product Backlog (<http://goo.gl/ce9SNz>)

**Setup:**

**For python:**

1. Download python 2.7 for windows 32bits (<https://www.python.org/download/>)
2. Add to PATH:

C:\Python27\

C:\Python27\Scripts

**For pip:**

We need pip to do the installation of the required dependencies. pip requires setuptools and it has to be installed first, before pip can run: <http://www.pip-installer.org/en/latest/installing.html>

1. Download get-pip.py
2. In a command prompt :

python get-pip.py

**For the virtual environment:**

1. In a command prompt:

pip install virtualenvwrapper-win

1. Add environment variable (Win 7 :Start, right click on Computer, Properties, click on Advance system settings in the left panel, go to advanced tab, Environment Variables…, New)

WORKON\_HOME=%USERPROFILE%\Develop\Envs

1. In a command prompt and then :

mkdir %WORKON\_HOME%

cd %WORKON\_HOME%

mkvirtualenv mgi

1. To use the environment:

workon mgi (the prompt will change and become mgi. You should always see the mgi prompt when installing new packages)

**For ZeroMQ:**

1. In a command prompt:

pip install pyzmq

**For Jena:**

1. Be sure that the folder <PROJECT PATH>\mdcs\data exists and contains a folder named ts. Otherwise, in a command prompt:

cd <PROJECT PATH>\mdcs

mkdir data

cd data

mkdir ts

1. Be sure that a jdk7 is installed on your computer and that it is in your path. In a command prompt, the following command should display the version of javac (1.7.\*):

javac –version

Otherwise:

* Download a jdk7 from <http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>
* Install the jdk
* Add C:\Program Files\Java\jdk1.7.0\_60\bin to your path.

1. In a command prompt:

cd <PROJECT PATH>\mdcs\rdf

javac -cp “lib/\*” RdfServer.java SparqlServer.java JenaServers.java

java -cp “lib/\*;.” JenaServers -rdfserver\_endpoint “tcp://127.0.0.1:5555” -sparqlserver\_endpoint “tcp://127.0.0.1:5556” -tdb\_directory <TS PATH> -project\_uri <PROJECT URI>

Command Example: C:\Users\<username>\workspace\MDCS\_PROJECT\mdcs\rdf>java -cp "lib/\*;." JenaServers -rdfserver\_endpoint "tcp://127.0.0.1:5555" -sparqlserver\_endpoint "tcp://127.0.0.1:5556" -tdb\_directory "C:\Users\<username>\workspace\MDCS\_Project\mdcs\data\ts" -project\_uri “http://www.example.com/”

**For MongoDB:**

1. Download Mongo db (<http://www.mongodb.org/downloads>) and install it in C:.
2. In a command prompt :

c:

mongod –dbpath PROJECT\_PATH\data\db --bind\_ip 127.0.0.1

To restrict access to mongodb database and ensure a better level of security, you may also:

* Add authentication/access control,
* Add traffic encryption,
* Have firewall rules.

You can find more about this topic: <http://docs.mongodb.org/manual/administration/security-checklist/>

**For Mongoengine:**

1. In a command prompt :

pip install mongoengine (<http://docs.mongoengine.org/guide/installing.html>)

**For Django/Dajax/Dajaxice:**

1. In a command prompt :

pip install Django (<https://docs.djangoproject.com/en/1.6/topics/install/>)

pip install django-dajax (<http://django-dajax.readthedocs.org/en/latest/installation.html>)

pip install django-dajaxice (<http://django-dajaxice.readthedocs.org/en/latest/installation.html>)

Go to C:\Users\<username>\Develop\mgi\Lib\site-packages and modify the following files to make dajaxice work:

Any Python version :

* In dajaxice\core\\_\_init\_\_.py :
  + Replace :
    - from Dajaxice
  + By :
    - from .Dajaxice
* In dajaxice\urls. py
  + Replace :
    - from django.conf.urls.defaults import \*
  + By :
    - from django.conf.urls import patterns, url, include

Python 3.\* :

* In Views.py :
  + Change safe\_dict(d) to :

def safe\_dict(d):

                                """

                                Recursively clone json structure with UTF-8 dictionary keys

<http://www.gossamer-threads.com/lists/python/bugs/684379>

                                """

                                if isinstance(d, dict):

                                                return dict([(k, safe\_dict(v)) for k, v in d.items()])

                                elif isinstance(d, list):

                                                return [safe\_dict(x) for x in d]

                                else:

                                                return d

**For ldap:**

1. Install the django-auth-ldap package:
   1. Download django-auth-ldap package from pypi (<https://pypi.python.org/pypi/django-auth-ldap/1.2.0>)
   2. In a command prompt:

cd PATH\_TO\_django-auth-ldap\_FOLDER

python setup.py install build

**For Django Rest Framework:**

1. In a command prompt :

pip install djangorestframework (<http://www.django-rest-framework.org/>)

pip install django-rest-framework-mongoengine

pip install django-rest-swagger

**For MS Excel to HDF5 translation/mapping:**

1. In a command prompt :

pip install xlrd

**For lxml:**

1. Download lxml from pypi (<https://pypi.python.org/pypi/lxml/2.3>)
   1. Use the Windows link for Python 2.7 : lxml-2.3.win32-py2.7.exe (win32)
2. Run the installer
   1. If you use a virtual environment.

You can’t change the destination folder set in the installer so you will have to install the package in the specified folder (C:\Python27\Lib\site-packages\) and then copy the lxml directories from C:\Python27\Lib\site-packages\ to %USERPROFILE%\Develop\Envs\mgi\Lib\site-packages.

**For xmltodict:**

1. In a command prompt :

pip install xmltodict

**For dateutil:**

1. In a command prompt:

pip install python-dateutil

**For Requests:**

1. In a command prompt:

pip install requests

**To Run software:**

1. Run mongodb. In a command prompt :

mongod –dbpath PROJECT\_PATH\data\db --bind\_ip 127.0.0.1

1. Run Jena. In a command prompt :

java -cp “lib/\*;.” JenaServers -rdfserver\_endpoint “tcp://127.0.0.1:5555” -sparqlserver\_endpoint “tcp://127.0.0.1:5556” -tdb\_directory <TS PATH> -project\_uri <PROJECT URI>

1. Run the MDCS. In a command prompt :

workon mgi

cd path\_to\_mdcs\_folder

python manage.py runserver

To be able to access the system **remotely**, instead of using python manage.py runserver, use: python manage.py runserver 0.0.0.0:<port>

**Access:**

For Materials Data Curation System, Go to: <http://127.0.0.1:8000/>

For Materials Data Curation Administration, Go to: <http://127.0.0.1:8000/admin/>