

How to install the Cytomine Java Client

Here we describe how to install Cytomine Java Client and how to test basic features of Cytomine with it (there will be no major differences for Python).

First of all, launch Cytomine in your browser, go in your Account page (click on the user icon and on "Account") and save your public and private keys for the Java client.

Then, get the URL of the image server (IMS). We will also need the idStorage which can be found at the link http://CORE_URL/api/storage.json (see *Installation*).

Below we propose a template of a simple Groovy script with methods useful to add a project, add an ontology, upload an image and list annotations within an image.

1) Download and install the Cytomine Java Client on your local computer:

```
sudo apt-get install maven
sudo apt-get install -y software-properties-common
sudo add-apt-repository -y ppa:webupd8team/java
sudo apt-get -y update
sudo apt-get install -y oracle-java8-installer
sudo apt-get install groovy

cd Cytomine_src/
git clone https://github.com/cytomine/Cytomine-java-client.git
cd Cytomine-java-client/
mvn -U clean install

mkdir test
cp target/cytomine-java-client-1.0-SNAPSHOT-jar-with-dependencies.jar ../../test/
cd ../../test/
```

Then copy the content below into script.groovy file and run it with:

```
groovy -cp 'Cytomine-client-java.jar' script.groovy
```

Groovy script

```
//run me with
//groovy -cp 'Cytomine-client-java.jar' script.groovy
import be.cytomine.client.*;
import be.cytomine.client.collections.*;
import be.cytomine.client.models.*;
String publickey = "";
String privatekey = "";
String cytomineCoreUrl = ""; // http://CORE_URL
String cytomineImsUrl = ""; // http://IMS_URL

println "*****"
println "Launch Test:"
Cytomine cytomine = new Cytomine(cytomineCoreUrl, publickey, privatekey, "./");

println "Try to add an ontology"
Ontology ont = cytomine.addOntology("testAddOntology")
println "Ontology created : "
println "    name : " + ont.get("name")
println "    id : " + ont.getId()
println "Try to add a project"
Project proj = cytomine.addProject("testAddProject", ont.id)
println "Project created : "
println "    name : " + proj.get("name")
println "    ontology : " + proj.get("ontology")
println "    id : " + proj.getId()
println "*****"
println "List your project:"
ProjectCollection projects = cytomine.getProjects();
for(int i=0;i<projects.size();i++) {
    Project project = projects.get(i)
    println project.getId() + " " + project.get("name") + " (" + project.getLong("numberOfImages") + "
images)"
}
String imagePATH = ""; // fill here the absolute path of an image.
Long idStorage = ; // fill here the storage id from http://CORE_URL/api/storage.json

Cytomine cytomineUpload = new Cytomine(cytomineImsUrl+"/", publickey, privatekey, "./");
println cytomineUpload.uploadImage(imagePATH, proj.getId(), idStorage, cytomineCoreUrl, null, true);

println "*****"
println "List the images of your project:"
ImageInstanceCollection images = cytomine.getImageInstances(proj.getId());
for(int i=0;i<images.size();i++) {
    ImageInstance image = images.get(i)
    println image.getId()
}

println "*****"
println "List the annotations of your project:"
AnnotationCollection annotations = cytomine.getAnnotationsByProject(proj.getId());
for(int i=0;i<annotations.size();i++) {
    Annotation annotation = annotations.get(i)
    List termNameList = annotation.getList("term").collect{ termId ->
        return terms.list.find{it.id==termId}.get("name")
    }
    println annotation.getId() + " " + annotation.get("location") + " " + annotation.get("image") + " " +
annotation.get("name") + " " + annotation.get("cropURL") + (termNameList.isEmpty()? "" : " has term
${termNameList.join(",")}")
}
}
```