Codenvy Documentation

1. **Create a new workspace**
    a. Go to Dashboard > Create workspace
        i. Select Source: new from blank
        ii. Select Workspace: create new workspace from stack
            - Select stack authoring
            - Type in: FROM codenvy/ubuntu_jdk8_x11
        iii. Configure workspace: type in desired workspace name
        iv. Template: select blank-project
            - NOTE: This project blank project will be later deleted
        v. Click Create
        vi. Delete blank project inside workspace
    b. Or go to Workspaces > Add Workspace
        i. Name workspace
        ii. Click on the Runtime tab
            - Select Stack Authoring
            - Type in: FROM codenvy/ubuntu_jdk8_x11
        iii. Click Save

2. **Create a new Maven Project**
    a. Click Workspace on the top menu > Create Project
        i. Choose Maven under JAVA
        ii. Name project
        iii. Click next
            - **Artifact ID:** *whatever you choose you must remember it because it will be used in the pom.xml file
            - **Group ID:** *better to keep it same as Artifact ID, it will also be used in the pom.xml file
            - **Version:** 1.0
            - **Packaging:** JAR
        iv. Click create
    b. Right click on project folder > create new folder > name it “lib”

3. **Upload files**
    a. Small files

i. Click on the folder you want the file to be uploaded into, i.e. lib
ii. Click on the Project tab at the top of the menu
iii. Choose Upload file
iv. Select the file you want to upload from your local computer and click Upload

b. Big files (around 40 MB or more)
i. Save you files in a repository on GitHub or Bitbucket (or whatever you choose)
ii. In Codenvy, go to terminal
iii. Navigate to your folder, i.e. cd ProjectName/folderName/
iv. Clone your remote repository inside the lib folder

4. Adding Source code

a. Right Click on src/main/java folder
i. Click New
ii. Click on Java Package
iii. Upload/Add source code inside java package

5. Jar files

a. Make sure jar files are inside the lib folder
b. For each jar file come up with a
i. **Group ID**: *this preferably should be the package name of the java file you're referencing from
- Example:
a. From the Dyehard.jar, if you are referencing GameWindow.java from the Engine package, then group id would be Engine
ii. **Artifact ID:** *anything you want
c. Rename your jars
i. Artifact_ID-1.0.jar
d. In the terminal, navigate to your project folder: cd ProjectName
i. For each jar run this command

mvn install:install-file -**Dfile**=/projects/ProjectName/lib/nameOfJar.jar -**DgroupId**=GroupIDForJar -**DartifactId**=ArtifactIDForJar -**Dversion**=1.0 -**Dpackaging**=jar

***NOTE:** Make sure the main class doesn't inherit from any class from your jar files, or else the executable jar created by Maven will not execute.

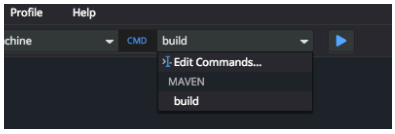
6. Modify pom.xml file

*Modify the green highlighted sections

*Modify the yellow highlighted sections if needed

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>Project_Group_ID</groupId>
  <artifactId>Project_Artifact_ID</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>
  <dependencies>
    <dependency>
      <groupId>Jar1_Group_ID</groupId>
      <artifactId>Jar1_Artifact_ID</artifactId>
      <version>1.0</version>
    </dependency>
    <dependency>
      <groupId>Jar2_Group_ID</groupId>
      <artifactId>Jar2_Artifact_ID</artifactId>
      <version>1.0</version>
    </dependency>
    <dependency>
      ...
    </dependency>
  </dependencies>
  <build>
<plugins>
  <plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-install-plugin</artifactId>
    <version>2.5.2</version>
    <executions>
      <execution>
        <id>install-external</id>
        <phase>clean</phase>
        <configuration>
          <file>${basedir}/lib/nameOfJar1.jar</file>
          <repositoryLayout>default</repositoryLayout>
```

```
                <groupId>Jar1_Group_ID</groupId>
                <artifactId>Jar1_Artifact_ID</artifactId>
                <version>1.0</version>
                <packaging>jar</packaging>
                <generatePom>true</generatePom>
                <file>${basedir}/lib/nameOfJar2.jar</file>
                <repositoryLayout>default</repositoryLayout>
                <groupId>Jar2_Group_ID</groupId>
                <artifactId>Jar2_Artifact_ID</artifactId>
                <version>1.0</version>
                <packaging>jar</packaging>
                <generatePom>true</generatePom>
                …
            </configuration>
            <goals>
              <goal>install-file</goal>
            </goals>
          </execution>
        </executions>
      </plugin>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-jar-plugin</artifactId>
        <version>2.3.1</version>
        <configuration>
          <archive>
            <manifest>
              <mainClass>packageOfMainClass.NameOfMainClass</mainClass>
              <addClasspath>true</addClasspath>
              <classpathPrefix>${basedir}/lib/</classpathPrefix>
            </manifest>
          </archive>
        </configuration>
      </plugin>
    </plugins>
    </build>
    </project>
```

7. **Create Maven build command**
   a. 

   b. Go to the commands tab and click on the drop down button

c. Click Edit Command
d. Add a Maven Command
    i. **Name:** build
    ii. **Preview URL:** http://${server.port.6080}

8. Run Java Swing Project

a. In the command tab, choose the Maven build command
b. Click the blue play button
c. After build is done, click on the preview URL
    i. noVNC window will open
d. In the noVNC window
    i. Right click and choose Terminal
    ii. Do the following commands
        - Navigate to target folder inside the project folder: cd ProjectName/target
        - Run executable jar file inside target: java –jar nameOfJar.jar
e. If you make a change in the code
    i. Close noVNC window
    ii. Build the project again
    iii. Click on the preview URL
    iv. Do those commands in noVNC all over again