

## Comment déployer la base de donnée :

### Matériel requis :

- Machine virtuelle distante tournant sur Linux
- Docker installé

### Commandes à exécuter dans la machine virtuelle:

-> `systemctl start docker.service`

-> `sudo docker run --publish 3306:3306 --detach --name TrombiBDD --env MARIADB_ROOT_PASSWORD=trombipw mariadb:latest`

-> `sudo docker ps` (facultatif permet de vérifier si le docker tourne)

-> `sudo docker exec -it TrombiBDD mariadb --user root -ptrombipw`

// le terminal devrait désormais être dans mariadb

-> `CREATE DATABASE datatest;`

-> `CREATE TABLE datatest.ELEVE (  
 prenom TINYTEXT,  
 nom TINYTEXT,  
 email TINYTEXT,  
 specialite TINYTEXT,  
 option TINYTEXT,  
 td TINYTEXT,  
 tp TINYTEXT,  
 tdMut TINYTEXT,  
 tpMut TINYTEXT,  
 ang TINYTEXT,  
 innov TINYTEXT,  
 mana TINYTEXT,  
 expr TINYTEXT,  
 annee TINYTEXT,  
 image LONGBLOB,  
 col1 TINYTEXT,  
 col2 TINYTEXT,  
 col3 TINYTEXT,  
 col4 TINYTEXT,  
 CONSTRAINT antiDoublon UNIQUE (email)  
);`

## Comment lancer le projet (usage de maven) :

Dans le terminal -> se placer dans le répertoire ./v1 et lancer la commande "mvn javafx:run"

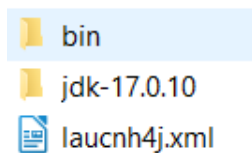
## Comment déployer le projet (maven + launch4j) :

Dans le terminal -> se placer dans le répertoire ./v1 et lancer la commande  
"mvn clean install assembly:single"

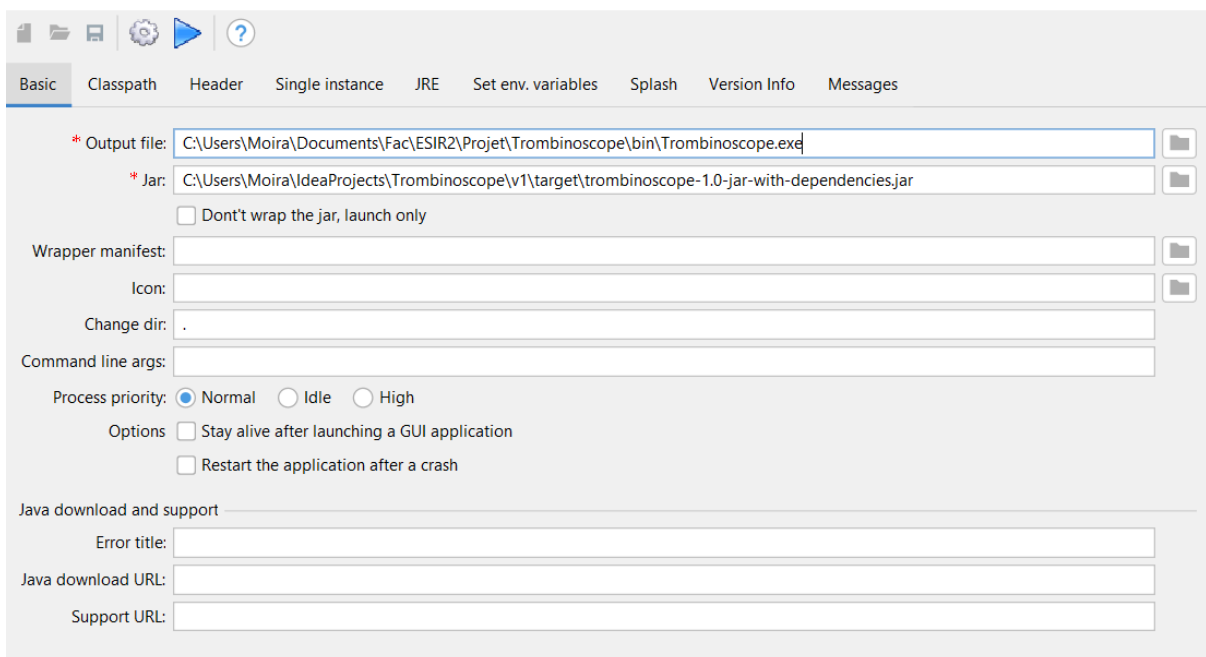
Récupérer le fichier trombinoscope-1.0-jar-with-dependencies.jar présent dans le dossier target du dossier v1.

Créer un répertoire pour l'application comportant un sous-répertoire bin (dans lequel on va mettre le fichier .exe) et le jdk-17.0.10 (disponible au téléchargement sur internet).

Le dossier de l'application devrait ressembler à ça :



Lancer launch4j (application permettant de créer des fichiers .exe à partir de fichiers .jar). Dans l'onglet Basic, mettre le chemin vers le futur fichier .exe (donc dans le sous-dossier bin du dossier de l'application) et dans Jar, mettre le chemin vers le fichier trombinoscope-1.0-jar-with-dependencies.jar.



Dans l'onglet JRE, dans la case JRE paths, mettre "..\jdk-17.0.10\bin". Appuyer sur le petit engrenage en haut à gauche, le fichier .exe doit être créé (enregistrer le fichier .xml dans le dossier du trombinoscope).

The screenshot shows the JRE configuration window with the following details:

- Tab:** JRE
- JRE paths:** ..\jdk-17.0.10\bin
- Search options:**
  - Min JRE version: [ ]
  - Max JRE version: [ ]
  - ☐ JDK required
  - ☐ 64-Bit required
- Options:**
  - Initial heap size: [ ] MB [ ] % of available memory
  - Max heap size: [ ] MB [ ] % of available memory
  - JVM options: [ ]
- Variables / registry:** EXEDIR
- \* Environment var:** [ ]
- Buttons:** Property, Option

## Modifier une colonne :

Pour changer le titre de la colonne "col\_A" en "col\_B"  
(dans le cas où l'on veut rajouter une colonne il faut les ajouter de gauche à droite donc ne pas commencer par la plus à droite).

## Dans la BDD :

Se connecter à la machine virtuelle dans un terminal puis utiliser la commande :

-> sudo docker exec -it TrombiBDD mariadb --user root -ptrombipw

Une fois dans la BDD :

-> ALTER TABLE datatest.ELEVE RENAME COLUMN col\_A TO col\_B;

## Dans le code :

Dans ./V1/src/main/java/trombi/BDD/listType.java

Si la colonne était déjà utilisée aller à l'objet qui correspond et dans la méthode toString changer la valeur de retours par "col\_B"

Si la colonne n'était pas encore utilisé il suffit d'ajouter un objet au singleton (voir ceux présent pour exemple) le nom associé n'as pas d'importance il faut cependant que le retour du toString() soit bien "col\_B"