**Hack.me**



Table des matières

[Prérequis 3](#_Toc520058778)

[Comment ça marche 3](#_Toc520058779)

[Developpement 3](#_Toc520058780)

[Api node 3](#_Toc520058781)

[Client lourd 3](#_Toc520058782)

[Les IHM 3](#_Toc520058783)

[Le jeu 3](#_Toc520058784)

[Les Annotations 3](#_Toc520058785)

[La librairie 4](#_Toc520058786)

[Les plugins 4](#_Toc520058787)

[Site web 5](#_Toc520058788)

[Utilisation 5](#_Toc520058789)

[Commandes pour le client lourd 5](#_Toc520058790)

[Commandes pour l'api node 5](#_Toc520058791)

[Commandes pour le site web 5](#_Toc520058792)

[Liens 6](#_Toc520058793)

# Prérequis

* Un serveur Web
* Une base de données MySQL
* Node
* Java 9

# Comment ça marche

Tout d’abord vous devez démarrer votre serveur web ainsi que votre base de données, une fois cela fait lancez l’API node puis le site web (hackmeweb)

Ensuite lancer le setup.exe pour installer l’application sur votre ordinateur, puis lancer l’application.

Ensuite une fois le programme lancer une fenêtre de connexion apparaitra, une fois connecter vous pourrez vous déplacer sur l'application et ainsi vous pourrez télécharger des plugins, des maps ou jouer.

# Developpement

## Api node

## Client lourd

### Les IHM

Les IHM ont été créée à partir du scène builder d’Oracle, puis nous devons les placer dans le dossier view de notre application.

Ensuite chaque vu (fichier fxml créé avec le scène builder) possède un controller, dans le controller pour accéder au éléments de notre vu il faut créer une variable du même nom que le fx:id de l’élément et du même type que l’élément (AnchorPane, Button, …) avec l’annotation @FXML au-dessus.

Vous pouvez désormais utiliser vos éléments dans votre code.

Vous pouvez aussi créer une méthode nommé "public void initialize()" cette méthode définie le traitement à exécuter avant l’affichage de la page.

### Le jeu

### Les Annotations

### La librairie

Il y a une librairie "hackmelibrairie" possédant les interfaces utiles au chargement et à la création des plugins. Pour créer cette librairie nous avons donc créé un projet maven dans lequel nous avons créé des interfaces avec des méthodes que l’utilisateur devra redéfinir.

Cette librairie sera accessible via la dépendance maven :

<dependency>  
 <groupId>com.esgi</groupId>  
 <artifactId>hackme-library</artifactId>  
 <version>1.0.0</version>  
</dependency>

### Les plugins

#### Charger un plugin

Pour cela une classe PluginLoader à été créée permettant de charger la classe principale d’un jar.

Elle possède un constructeur par défaut dans lequel un chemin vers le dossier de plugin par défaut est donné et un second constructeur dans lequel vous pouvez définir le chemin vers le dossier que vous souhaitez.

Ensuite elle possède une méthode loadPlugin qui prend en paramètre une String qui est le nom du jar que vous souhaitez charger et elle vous renvois un objet que vous devrez caster dans l’objet que vous désirez pour l’utiliser.

Cette classe ouvrira votre jar et ira lire le fichier Manifest.mf pour déterminer quelle est la class main de votre plugin, ensuite elle chargera cette Class grâce à un ClassLoader avant de vous la retourner sous la forme d’un objet.

#### Créer un Plugin

Pour créer un Plugin vous devez importer une librairie contenant les interfaces que vous pourrez utiliser dans vos plugins :

<dependency>  
 <groupId>com.esgi</groupId>  
 <artifactId>hackme-library</artifactId>  
 <version>1.0.0</version>  
</dependency>

Ainsi que la dépendance du jeu :

<dependency>  
 <groupId>com.esgi</groupId>  
 <artifactId>hackme</artifactId>  
 <version>1.0.0</version>  
</dependency>

Pour pouvoir utilisez les Class du Client lourd ainsi que leurs méthodes comme celle reliant l’application à l’API node.

Ensuite vous devrez créer une Class main implémentant l’interface que vous souhaitez utiliser dans la librairie et définir ce que vous souhaitez faire dans les méthodes qui vous seront donnée.

## Site web

# Utilisation

## Commandes pour le client lourd

```bash

git clone https://github.com/bibeul/Projet\_annuel\_3AL1\_2018.git

cd Projet\_annuel\_3AL1\_2018

cd client\_lourd

mvn clean install

mvn exec:java -Dexec.MainClass="hackme.Main"

ou

java -Djava.library.path=src/main/resources/lib/natives -jar target/hackme-jar-with-dependencies.jar

```

Ou

Lancer le setup.exe pour installer le client lourd

## Commandes pour l'api node

```bash

git clone https://github.com/bibeul/Projet\_annuel\_3AL1\_2018.git

cd Projet\_annuel\_3AL1\_2018

cd api\_PA

npm install

npm start

```

## Commandes pour le site web

```bash

git clone https://github.com/bibeul/Projet\_annuel\_3AL1\_2018.git

cd Projet\_annuel\_3AL1\_2018

cd hackmeweb

npm install

npm start

```

# Liens

<http://deptinfo.unice.fr/twiki/pub/Minfo/GenieLog1415/cours6-GL-minfo-1415.pdf>

<http://miageprojet2.unice.fr/@api/deki/files/1399/=02_Chargement_dynamique.pdf>