

# Documentation Technique

## Sommaire

<b>1. Présentation de l'application</b>	3
<b>2. Architecture de l'application</b>	4
2.1 Architecture générale	4
2.2 Ressources externes	4
2.2.1 API utilisées	4
2.2.2 Plugins utilisés	4
2.3 Structuration de l'application	5
2.4 Spécifications techniques	5
<b>3. Conception et mise en oeuvre fonctionnalités</b>	6
<b>4. Procédures d'installation</b>	7
4.1 Installation pour développement	7
4.1.1 Prérequis	7
4.1.2 Étapes d'installation	7
4.2 Installation pour utilisation	8
4.2.1 Prérequis	8
4.2.2 Étapes d'installation	8
4.2.3 Étapes de lancement	9

### **Étudiants**

Nolhan Biblocque  
Léo Guinvarc'h  
Victor Jockin  
Mathys Laguilliez  
Mucahit Lekesiz

### **Enseignant**

André Péninou

### **Formation**

BUT Informatique  
2ème Année  
Promotion 2024-2025

### **Établissement**

IUT de Blagnac  
Université Toulouse II – Jean Jaurès (31)

# 1. Présentation de l'application

## 2. Architecture de l'application

### 2.1 Architecture générale

### 2.2 Ressources externes

#### 2.2.1 API utilisées

##### *JavaFX*

- **Rôles :**
  - Conception de l'IHM avec le module `javafx-fxml` (création d'interfaces utilisateur via des fichiers FXML).
  - Prise en charge et gestion de l'interface graphique dans l'application.
- **Version utilisée :** 17
- **Site officiel de JavaFX :** [JavaFX - Home](#)
- **Documentation officielle :** [Oracle - JavaFX Documentation](#)

##### *OpenCSV*

- **Rôle :** Lecture des fichiers de données au format `CSV` générés par le programme python collecteur de données.
- **Version utilisée :** 5.5.2
- **Site officiel de JavaFX :** [OpenCSV - About / Opencsv Users Guide](#)
- **Documentation officielle :** [OpenCSV - About / Developer Documentation](#)

#### 2.2.2 Plugins utilisés

##### *JavaFX Maven Plugin*

- **Rôle :** Packaging et exécution de l'application JavaFX.
- **Version utilisée :** 0.0.8
- **Site officiel de Maven Repository :** [Maven Repository - JavaFX Maven Plugin Maven Mojo](#)
- **Lien vers le dépôt GitHub du plugin :** [GitHub - Maven plugin for JavaFX](#)

##### *Apache Maven Shade Plugin*

- **Rôle :** Création d'un exécutable au format `JAR` contenant toutes les dépendances nécessaires au fonctionnement de l'application.
- **Version utilisée :** 3.4.1
- **Site officiel d'Apache Maven :** [Apache Maven Project - Apache Maven Shade Plugin](#)

### *Apache Maven Javadoc Plugin*

- **Rôle** : Génération de la documentation du projet Java avec **Javadoc**.
- **Version utilisée** : 3.4.1
- **Site officiel d'Apache Maven** : [Apache Maven Project - Apache Maven Javadoc Plugin](#)

## 2.3 Structuration de l'application

## 2.4 Spécifications techniques

### **3. Conception et mise en oeuvre fonctionnalités**

## 4. Procédures d'installation

### 4.1 Installation pour développement

#### 4.1.1 Prérequis

##### 1. Installer l'environnement de développement Java

- Télécharger le **JDK 17** (ou version compatible) depuis le site officiel d'Oracle : [Oracle - Java Downloads](#).
- Installer le JDK en suivant les instructions indiquées par l'installateur.
- Si nécessaire, ajouter le chemin vers le JDK à la variable d'environnement **PATH**.
- Dans un terminal, vérifier l'installation avec la commande `java -version` ou `java --version`.

##### 2. Installer Apache Maven

- Télécharger **Maven** (archive ZIP) depuis le site officiel d'Apache Maven : [Apache Maven Project - Downloading Apache Maven](#).
  - Pour une installation sur Linux ou Mac OS, télécharger la **Binary tar.gz archive**.
  - Pour une installation sur Windows, télécharger la **Binary zip archive**.
- Ajouter le chemin vers Maven à la variable d'environnement **PATH**.
- Dans un terminal, vérifier l'installation avec la commande `mvn -version`, `mvn --version` ou `mvn -v`.

##### 3. Configurer un IDE

- Si nécessaire, installer des plugins de prise en charge de **Maven** et **JavaFX** dans l'IDE utilisé pour le développement.

#### 4.1.2 Étapes d'installation

##### 1. Cloner le dépôt du projet

- Accéder au dépôt GitHub du projet : [GitHub - SAE S3.01 DevApp](#)
- Cloner le dépôt du projet via la commande :

```
git clone https://github.com/IUT-Blagnac/sae-3-01-devapp-2024-2025-g2b12.git
```

- Accéder au répertoire du projet Java situé sous `solution iot/application_iot` via la commande :

```
cd solution\ iot/application_iot
```

##### 2. Construire le projet avec Maven

- Supprimer les fichiers et ressources précédemment compilés avec la commande `mvn clean`

puis compiler le projet Java via la commande `mvn install`. Il est également possible d'utiliser directement la commande `mvn clean install`.

### 3. Exécuter l'application depuis Maven

- Exécuter le projet JavaFX via la commande `mvn javafx:run`.

## 4.2 Installation pour utilisation

### 4.2.1 Prérequis

#### 1. Installer le Java Runtime Environment (JRE)

- Vérifier que Java est installé sur la machine en exécutant la commande `java -version` dans un terminal.
- Si Java n'est pas installé, télécharger et installer le **JRE 8** ou version ultérieure depuis le site officiel de Java : [Java - Télécharger Java](#).

#### 2. Installer Python 3

- Vérifier que Python en version 3 est installé sur la machine en exécutant la commande `python -version` ou `python3 -version` dans un terminal.
- Si Python n'est pas installé, télécharger et installer la dernière version disponible sur le site officiel de Python : [Python - Downloads](#).

### 4.2.2 Étapes d'installation

#### 1. Télécharger l'application

- Télécharger l'archive de l'application (fichier ZIP) située sous le répertoire `livrables/IoT` du dépôt GitHub du projet : [GitHub - Livrables IoT](#)
  - Pour une installation sur Mac OS, préférer l'archive `application_jar_mac_os.zip`.
  - Pour une installation sur Windows ou Linux, préférer l'archive `application_jar_windows.zip`.

#### 2. Décompresser l'archive de l'application

- Décompresser l'archive téléchargée dans un répertoire à l'aide d'un outil de décompression tel que **WinRAR** ou **7-Zip**.
- L'arborescence de l'application après décompression doit ressembler à ceci :

```
application/  
|-- ressources/  
|   |-- data/  
|   |-- configuration.ini  
|   |-- mqtt.py  
|-- application_iot-1.0-SNAPSHOT-shaded.jar
```



## 4.2.3 Étapes de lancement

### 1. Lancer l'application dans le gestionnaire de fichiers

- Lancer l'exécutable `application_iot-1.0-SNAPSHOT-shaded.jar` en double-cliquant sur celui-ci.
- *Le menu principal de l'application devrait alors apparaître à l'écran.*

### 2. Lancer l'application en ligne de commande

- Ouvrir un terminal et se placer dans le répertoire `application` à l'aide de la commande `cd`.
- Lancer ensuite l'exécutable de l'application via la commande :

```
java -jar application_iot-1.0-SNAPSHOT-shaded.jar
```

- *Le menu principal de l'application devrait alors apparaître à l'écran.*

### **Étudiants**

Nolhan Biblocque  
Léo Guinvarc'h  
Victor Jockin  
Mathys Laguilliez  
Mucahit Lekesiz

### **Enseignant**

André Péninou

### **Formation**

BUT Informatique  
2ème Année  
Promotion 2024-2025

### **Établissement**

IUT de Blagnac  
Université Toulouse II – Jean Jaurès (31)