**SAE 302 -- Développer des applications communicantes**



**Installation et Configuration de TraceCord**

**Auteurs : Bastien Labeste, Robin Kwiatkowski, Quentin Chambelland**

**📅 Dernière mise à jour : 16 Février 2025**

Table des matières

[1.Introduction 3](#_Toc190080358)

[2. Pré-requis 4](#_Toc190080359)

[2.1. Matériel requis 4](#_Toc190080360)

[2.2. Logiciels et outils nécessaires 4](#_Toc190080361)

[3. Configuration du Serveur 4](#_Toc190080362)

[3.1 Installation des dépendances serveur 4](#_Toc190080363)

[Installer PHP 4](#_Toc190080364)

[Installer MariaDB et PHP-MySQL 4](#_Toc190080365)

[Redémarrer le serveur Apache pour appliquer les changements 4](#_Toc190080366)

[Configurer l’utilisateur root de MariaDB 5](#_Toc190080367)

[Installer phpMyAdmin 5](#_Toc190080368)

[Accéder à phpMyAdmin 5](#_Toc190080369)

[3.2 Création de la base de données et des tables 6](#_Toc190080370)

[Créer la base de données et la table des messages 6](#_Toc190080371)

[Vérifier que la table a bien été créée 6](#_Toc190080372)

[3.3 Importation d’une base de données existante 6](#_Toc190080373)

[Via le terminal MariaDB 6](#_Toc190080374)

[Via phpMyAdmin 7](#_Toc190080375)

[3.4 Vérification de la connexion à MariaDB 7](#_Toc190080376)

[Se connecter à MariaDB 7](#_Toc190080377)

[Lister les bases de données 7](#_Toc190080378)

[Sélectionner la base de données Discord 7](#_Toc190080379)

[Vérifier que des données sont présentes 7](#_Toc190080380)

[Quitter MariaDB 8](#_Toc190080381)

[4. Déploiement et Configuration de l’API 8](#_Toc190080382)

[4.1 Installation et Configuration du Serveur Web 8](#_Toc190080383)

[Vérification d’Apache et PHP 8](#_Toc190080384)

[Vérifier son statut 8](#_Toc190080385)

[Tester PHP 9](#_Toc190080386)

[4.2 Déploiement de l’API PHP 9](#_Toc190080387)

[Placer les fichiers API sur le serveur 9](#_Toc190080388)

[Configuration de l’API 10](#_Toc190080389)

[Vérification du bon fonctionnement de l’API 10](#_Toc190080390)

[4.3 Sécurisation et Accès à Distance 11](#_Toc190080391)

[Autoriser les connexions externes à la base de données 11](#_Toc190080392)

[Ouvrir le port 3306 sur le pare-feu 12](#_Toc190080393)

[4.4 Vérification Finale 12](#_Toc190080394)

[5. Installation de l’Application Android 12](#_Toc190080395)

[5.1. Configuration d’Android Studio 12](#_Toc190080396)

[5.2. Exécution de l’application 13](#_Toc190080397)

[6. Dépannage et Résolution des Problèmes 14](#_Toc190080398)

[7. Conclusion 14](#_Toc190080399)

# 1.Introduction

L’objectif de ce document est de détailler toutes les étapes nécessaires pour installer et configurer l’application **TraceCord**, un projet Android permettant d’afficher la liste des utilisateurs et leurs messages provenant d’une base de données MySQL.

Ce guide d’installation couvre :

* La mise en place de la base de données MySQL
* La configuration du serveur PHP
* L’installation et l’exécution de l’application Android

Chaque section est détaillée afin de garantir une mise en place fonctionnelle de l’application.

# 2. Pré-requis

## 2.1. Matériel requis

Pour installer et exécuter **TraceCord**, assurez-vous de disposer des éléments suivants :

* Un serveur Linux (Ubuntu/Debian recommandé)
* Un ordinateur avec Android Studio installé pour exécuter l’application

## 2.2. Logiciels et outils nécessaires

* MySQL : Gestion de la base de données
* Apache + PHP : Serveur web pour exécuter les scripts PHP
* PhpMyAdmin : Interface web pour gérer la base de données
* Android Studio : IDE pour exécuter et tester l’application

# 3. Configuration du Serveur

## 3.1 Installation des dépendances serveur

### Installer PHP

* sudo apt install php libapache2-mod-php -y

### Installer MariaDB et PHP-MySQL

* sudo apt install mariadb-server php-mysql -y

### Redémarrer le serveur Apache pour appliquer les changements

* sudo service apache2 restart

### Configurer l’utilisateur root de MariaDB

* sudo mysql --user=root

Puis, dans le terminal MariaDB, exécutez les commandes suivantes :

* DROP USER 'root'@'localhost';
* CREATE USER 'root'@'localhost' IDENTIFIED BY 'root';
* GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'localhost';
* QUIT;

### Installer phpMyAdmin

* sudo apt install phpmyadmin -y

Lors de l’installation :

* Sélectionnez **apache2** comme serveur web.
* Pour la configuration de phpMyAdmin, sélectionnez **NO** (le package n’est pas indispensable).

### Accéder à phpMyAdmin

1. Ouvrez un navigateur et entrez l’adresse suivante :

* http://<IP\_DU\_SERVEUR>/phpmyadmin

(Remplacez <IP\_DU\_SERVEUR> par l’adresse IP du serveur).

1. Connectez-vous avec :
   * **Utilisateur** : root
   * **Mot de passe** : root

## 3.2 Création de la base de données et des tables

Une fois connecté à MariaDB, exécutez les commandes suivantes pour créer la base de données et la table des messages.

### Créer la base de données et la table des messages

* CREATE DATABASE Discord;
* USE Discord;
* CREATE TABLE messages (

id INT PRIMARY KEY AUTO\_INCREMENT,

nom\_utilisateur VARCHAR(255),

message TEXT,

date DATETIME,

receveur VARCHAR(255),

message\_length INT

);

### Vérifier que la table a bien été créée

* SHOW TABLES;
* SELECT \* FROM messages;

## 3.3 Importation d’une base de données existante

Si vous disposez d’un fichier SQL contenant des données préexistantes, vous pouvez l’importer de deux manières :

### Via le terminal MariaDB

* mysql -u root -p Discord < dump.sql

### Via phpMyAdmin

* Accéder à phpMyAdmin
* Sélectionner la base de données Discord
* Aller dans l’onglet "Import"
* Charger le fichier SQL (dump.sql)
* Cliquer sur "Exécuter" pour importer les données

## 3.4 Vérification de la connexion à MariaDB

Après avoir installé et configuré MariaDB, nous allons tester la connexion et vérifier que les données sont bien présentes dans la base.

### Se connecter à MariaDB

Ouvrez un terminal et exécutez :

* sudo mysql --user=root

### Lister les bases de données

* SHOW DATABASES;

Vous devriez voir apparaître la base de données Discord.

### Sélectionner la base de données Discord

* USE Discord;

### Vérifier que des données sont présentes

* SELECT \* FROM messages;

Si la base de données est bien remplie, cette commande affichera les messages stockés dans la table messages.

### Quitter MariaDB

* QUIT;

# 4. Déploiement et Configuration de l’API

## 4.1 Installation et Configuration du Serveur Web

Une fois la base de données et MariaDB configurés, nous devons mettre en place l'API PHP qui permettra à l’application Android de récupérer et d’envoyer des données.

### Vérification d’Apache et PHP

Apache et PHP ont déjà été installés à l’étape **3.1**. Vérifions que le serveur fonctionne correctement :

Lancer Apache s’il n’est pas actif

* sudo systemctl start apache2

### Vérifier son statut

* sudo systemctl status apache2

Si Apache fonctionne correctement, vous verrez un message indiquant qu’il est "active (running)".

### Tester PHP

* Créez un fichier de test :
  + sudo nano /var/www/html/info.php
  + Ajoutez ce contenu :
    - <?php

phpinfo();

?>

* + Enregistrez et fermez le fichier (CTRL + X, puis Y, puis Entrée).
  + Ouvrez un navigateur et entrez :
    - http://<IP\_DU\_SERVEUR>/info.php

Si la page s’affiche avec les informations PHP, alors PHP est bien configuré.

## 4.2 Déploiement de l’API PHP

### Placer les fichiers API sur le serveur

Les fichiers API doivent être placés dans le dossier /var/www/html/. Copiez les fichiers PHP nécessaires dans ce répertoire :

* sudo mv get\_users.php /var/www/html/
* sudo mv get\_messages.php /var/www/html/
* sudo mv config.php /var/www/html/

Assurez-vous que les fichiers ont les bons droits d’accès :

* sudo chmod 644 /var/www/html/\*.php

### Configuration de l’API

Dans config.php, entrez les informations de connexion à la base de données :

* <?php

$host = "localhost";

$dbname = "Discord";

$user = "root";

$pass = "root";

try {

$pdo = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8", $user, $pass);

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

} catch (PDOException $e) {

die("Erreur de connexion : " . $e->getMessage());

}

?>

### Vérification du bon fonctionnement de l’API

* Tester les endpoints via un navigateur
  + Liste des utilisateurs :
    - http://<IP\_DU\_SERVEUR>/get\_users.php
  + Messages d’un utilisateur :
    - http://<IP\_DU\_SERVEUR>/get\_messages.php?username=<NOM\_UTILISATEUR>
* Si tout fonctionne bien, les résultats seront affichés en JSON.
* Tester via curl
  + Depuis un terminal, exécutez :
    - curl -X GET http://<IP\_DU\_SERVEUR>/get\_users.php
  + Vous devriez voir la liste des utilisateurs en JSON.
* Activer les logs Apache pour le débogage (en cas de problème)
  + sudo tail -f /var/log/apache2/error.log

## 4.3 Sécurisation et Accès à Distance

### Autoriser les connexions externes à la base de données

* Modifier le fichier de configuration MySQL
  + sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
  + Recherchez la ligne :
    - bind-address = 127.0.0.1
  + Remplacez-la par :
    - bind-address = 0.0.0.0
  + Sauvegardez (CTRL + X, puis Y, puis Entrée).
* Redémarrer MariaDB
  + - sudo systemctl restart mariadb
* Autoriser l’accès distant pour l’utilisateur root
  + GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'%' IDENTIFIED BY 'root' WITH GRANT OPTION;
  + FLUSH PRIVILEGES;

### Ouvrir le port 3306 sur le pare-feu

Si vous utilisez ufw (Uncomplicated Firewall) :

* sudo ufw allow 3306/tcp
* sudo ufw reload

Si vous utilisez iptables :

* sudo iptables -A INPUT -p tcp --dport 3306 -j ACCEPT

## 4.4 Vérification Finale

Après avoir suivi toutes ces étapes, assurez-vous que :

* L’API renvoie bien des données JSON.
* La base de données est accessible depuis une connexion distante.
* Les logs Apache et MySQL ne contiennent pas d’erreurs bloquantes.

# 5. Installation de l’Application Android

## 5.1. Configuration d’Android Studio

1. Ouvrir **Android Studio** et assurez-vous d’avoir installé :
   * JDK 11+
   * Gradle (dernière version compatible)
   * SDK Android API 33+
2. Vérifiez que les dépendances suivantes sont présentes dans **app/build.gradle** :

* dependencies {

implementation 'com.squareup.retrofit2:retrofit:2.9.0'

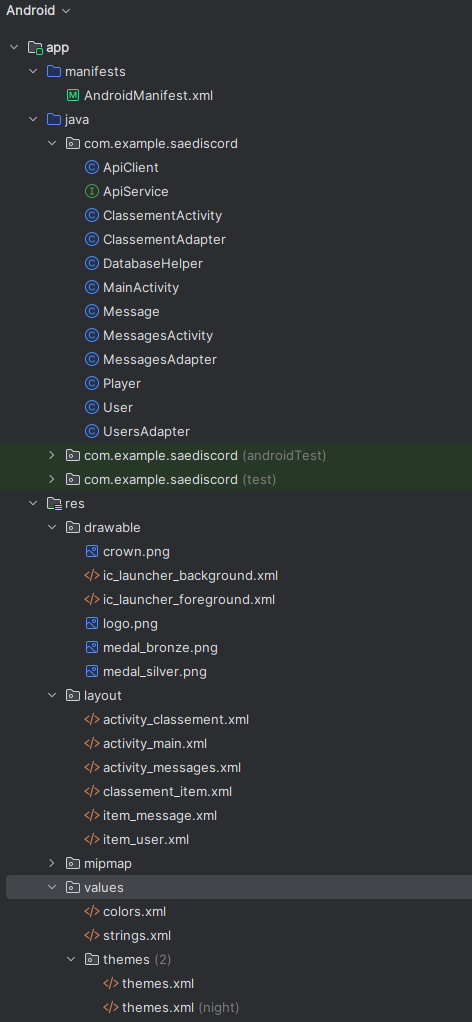
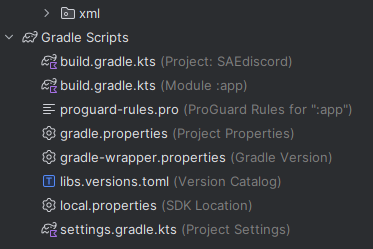
implementation 'com.squareup.retrofit2:converter-gson:2.9.0'

}

## 5.2. Exécution de l’application

1.Ouvrir le projet dans Android Studio

* Lancez **Android Studio**
* Créez le projet **SAEdiscord** en sélectionnant Empty Views Activity
* Récupérez les fichiers sur le Github et les importer aux bons endroits :



2. Configurer un appareil de test

* Option 1 : Tester sur un téléphone
  + Activez le **mode développeur** et le **débogage USB** sur votre téléphone
  + Branchez le téléphone à l’ordinateur via USB
  + Vérifiez qu’il est bien reconnu dans Android Studio (*onglet Select Deployment Target*)
* Option 2 : Utiliser un émulateur
  + Allez dans **Tools > Device Manager**
  + Créez un **Virtual Device** (API 30 ou plus recommandé)
  + Démarrez l’émulateur

3. Lancer le bot Discord et récupérer les messages

Avant d’exécuter l’application, il faut **récupérer les messages Discord** pour qu’ils soient affichés correctement.

* **Lancer le bot Discord**
  + Ouvrez un terminal dans le dossier du bot
  + Exécutez la commande :
    - python3 bot\_discord.py
  + Assurez-vous que le bot est bien en ligne sur Discord
* **Sélectionner les messages Discord**
  + Sur **Discord**, tapez la commande suivante dans un salon où le bot a accès :
    - !selection <nombre> #<serveur>

Remplacez **<nombre>** par le nombre de messages à récupérer et **<serveur>** par l’ID du serveur Discord.

* + **Alternative via le terminal**
    - Exécutez cette commande dans le terminal où tourne le bot :
      * python3 bot\_discord.py selection <nombre> <ID\_salon>

Remplacez **<nombre>** par le nombre de messages et **<ID\_salon>** par l’ID du salon à analyser.

* **Exécuter le traitement des messages**
  + Une fois les messages récupérés, tapez sur **Discord** :
    - !java
  + **Ou exécutez la commande directement dans un terminal** :
    - java -cp .:mysql-connector-java-9.2.0.jar DiscordDataProcessor
  + Cela enverra les messages analysés dans la **base de données**.

4. Exécuter l’application

* Dans Android Studio, cliquez sur **Run**
* Sélectionnez votre téléphone ou l’émulateur
* L’application se lance et affiche les utilisateurs

5. Fonctionnalités disponibles

Une fois l’application lancée, vous pouvez :

* **Consulter la liste des utilisateurs**
  + Rechercher un utilisateur via la barre de recherche
  + Cliquer sur un utilisateur pour voir ses messages
* **Analyser la toxicité des messages**
  + Chaque utilisateur a un **score de toxicité** basé sur les messages récupérés
  + Les messages sont triés et affichent leur score
* **Voir le classement de toxicité**
  + En appuyant sur **"Voir le classement"**, vous accédez au **top des utilisateurs les plus toxiques**
  + Les **trois premiers** reçoivent une médaille (or, argent, bronze)
* **Naviguer entre les écrans**
  + Depuis la liste des utilisateurs → Voir les messages
  + Depuis les messages → Voir le classement
  + Bouton **Retour** pour revenir en arrière

**6. Vérification finale**

* Si l’application ne récupère pas de messages :
  + Vérifiez la base de données avec :
    - SELECT \* FROM messages;
  + Vérifiez que le bot Discord a bien ajouté les messages
* Si le classement est vide :
  + Vérifiez que les utilisateurs ont bien des scores de toxicité

L’application est maintenant prête à être utilisée.

# 6. Dépannage et Résolution des Problèmes

| **Problème** | **Solution** |
| --- | --- |
| Aucun utilisateur trouvé | Vérifiez la base de données avec SELECT \* FROM messages; |
| Page blanche sur l’API | Activez error\_reporting(E\_ALL); et vérifiez les logs Apache sudo tail -f /var/log/apache2/error.log |
| L’application crash | Consultez **Logcat** dans Android Studio pour identifier l’erreur. |
| Impossible de se connecter à MySQL | Vérifiez bind-address=0.0.0.0 dans /etc/mysql/my.cnf et activez les connexions externes. |

# 7. Conclusion

L’installation et la configuration de **TraceCord** nécessitent une mise en place rigoureuse du serveur PHP, de la base de données MySQL et de l’environnement Android Studio. Si des erreurs persistent, assurez-vous de bien tester chaque partie indépendamment (API, serveur, base de données) avant de lancer l’application Android.

**Développé par :**

Bastien Labeste  
Robin Kwiatkowski  
Quentin Chambelland

**Dernière mise à jour :** 16 Février 2025