

# دليل بناء وتشغيل تطبيق HUS 🚀

## نظرة عامة 📋

تطبيق HUS هو تطبيق دردشة صوتية متطور يتكون من: - **الواجهة الخلفية (Backend):** Node.js + Fastify + MongoDB + Socket.io - **تطبيق الجوال:** Flutter مع دعم Android SDK 35 و JDK 21 - **قاعدة البيانات:** MongoDB Atlas (سحابية) - **المصادقة:** Firebase Authentication + JWT

## المتطلبات الأساسية 🛠️

### 1. متطلبات النظام

- **نظام التشغيل:** Linux , Windows 10/11, macOS
- **الذاكرة:** 8GB RAM كحد أدنى (16GB موصى به)
- **مساحة التخزين:** 10GB مساحة فارغة
- **الاتصال:** اتصال إنترنت مستقر

### 2. البرامج المطلوبة

Node.js و npm (أ)

```
# (الإصدار 18 أو أحدث) Node.js تحميل وتثبيت  
# من الموقع الرسمي: https://nodejs.org/
```

```
# التحقق من التثبيت
```

```
node --version # أو أحدث v18.x.x يجب أن يظهر  
npm --version # أو أحدث x.x.x يجب أن يظهر 9
```

## Flutter SDK (ب)

```
# من: https://flutter.dev/docs/get-started/install تحميل Flutter SDK  
  
# إضافة Flutter إلى PATH  
export PATH="$PATH:`pwd`/flutter/bin"  
  
# التحقق من التثبيت  
flutter --version  
flutter doctor # للتحقق من جميع المتطلبات
```

## Java JDK 21 (ج)

```
# من: https://adoptium.net/temurin/releases/ تحميل وتثبيت OpenJDK 21  
  
# التحقق من التثبيت  
java --version # يجب أن يظهر 21.x.x  
javac --version # يجب أن يظهر 21.x.x
```

## Android SDK (د)

```
# من: https://developer.android.com/studio تثبيت Android Studio  
# فقط command line tools أو تثبيت  
  
# تثبيت Android SDK 35 (API Level 35)  
sdkmanager "platforms;android-35"  
sdkmanager "build-tools;35.0.0"
```

## Git (هـ)

```
# من: https://git-scm.com/ تحميل وتثبيت Git  
  
# التحقق من التثبيت  
git --version
```

---

# تحميل المشروع 📦

## 1. استنساخ المستودع

```
# استنساخ المشروع من GitHub
git clone https://github.com/sukkar007/tre.git

# الانتقال إلى مجلد المشروع
cd tre
```

## 2. هيكل المشروع

```
tre/
├── backend/           # الواجهة الخلفية (Node.js)
│   ├── models/       # نماذج قاعدة البيانات
│   ├── routes/       # مسارات API
│   ├── services/     # خدمات WebSocket
│   ├── server.js      # الملف الرئيسي للخادم
│   └── package.json  # تبعيات Node.js
├── frontend/         # تطبيق Flutter
│   ├── lib/          # كود Dart
│   ├── android/      # إعدادات Android
│   └── pubspec.yaml  # تبعيات Flutter
├── README.md         # وثائق المشروع
└── BUILD_GUIDE.md    # هذا الدليل
```

## إعداد الواجهة الخلفية (Backend) 🛠️

### 1. تثبيت التبعيات

```
# الانتقال إلى مجلد الواجهة الخلفية
cd backend

# تثبيت جميع التبعيات
npm install
```

## 2. إعداد قاعدة البيانات MongoDB

### أ) إنشاء حساب MongoDB Atlas

1. اذهب إلى [MongoDB Atlas](#)
2. أنشئ حساباً جديداً أو سجل دخولك
3. أنشئ Cluster جديد (اختر الخطة المجانية M0)
4. أنشئ مستخدم قاعدة بيانات
5. أضف عنوان IP الخاص بك إلى القائمة البيضاء

### ب) الحصول على رابط الاتصال

1. اضغط على "Connect" في لوحة التحكم
2. اختر "Connect your application"
3. انسخ رابط الاتصال (Connection String)

## 3. إعداد متغيرات البيئة

```
# إنشاء ملف .env في مجلد backend
cp .env.example .env
```

```
# إضافة المعلومات التالية .env. تحرير ملف #
```

```
# .env ملف
NODE_ENV=development
PORT=3000

# MongoDB Atlas Connection
MONGODB_URI=mongodb+srv://username:password@cluster.mongodb.net/husapp?
retryWrites=true&w=majority

# JWT Secret (أنشئ مفتاح عشوائي قوي)
JWT_SECRET=your-super-secret-jwt-key-here-make-it-long-and-random

# Firebase Admin (اختياري للمرحلة الحالية)
FIREBASE_PROJECT_ID=your-firebase-project-id
FIREBASE_PRIVATE_KEY=your-firebase-private-key
FIREBASE_CLIENT_EMAIL=your-firebase-client-email

# CORS إعدادات
CORS_ORIGIN=http://localhost:3000,http://127.0.0.1:3000

# Socket.io إعدادات
SOCKET_CORS_ORIGIN=*
```

## 4. تشغيل الخادم

```
# تشغيل الخادم في وضع التطوير
npm run dev

# أو تشغيل عادي
node server.js
```

## 5. اختبار الخادم

```
# اختبار الاتصال الأساسي
curl http://localhost:3000/

# اختبار اتصال قاعدة البيانات
curl http://localhost:3000/test-db

# Splash محتوى API اختبار
curl http://localhost:3000/api/splash/content
```

# إعداد تطبيق Flutter

## 1. تثبيت التبعيات

```
# الانتقال إلى مجلد التطبيق
cd ../frontend

# تثبيت جميع التبعيات
flutter pub get
```

## 2. إعداد Firebase

### أ) إنشاء مشروع Firebase

1. اذهب إلى [Firebase Console](#)
2. أنشئ مشروعاً جديداً
3. فعّل Authentication وأضف طرق تسجيل الدخول:
4. Google Sign-In
5. Phone Authentication

### ب) إضافة التطبيق إلى Firebase

1. اضغط على "Add app" واختر Android
2. أدخل package name: `com.flamingolive.hus`
3. حمّل ملف `google-services.json`
4. ضع الملف في `/android/app`

## ج) تحديث إعدادات Android

```
// ملف android/app/build.gradle
android {
    compileSdkVersion 35

    defaultConfig {
        applicationId "com.flamingolive.hus"
        minSdkVersion 21
        targetSdkVersion 35
        versionCode 1
        versionName "1.0"
    }

    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_21
        targetCompatibility JavaVersion.VERSION_1_21
    }
}
```

## 3. تحديث إعدادات الاتصال

```
// ملف lib/src/utils/app_constants.dart
class AppConstants {
    // تحديث عنوان الخادم
    static const String baseUrl = 'http://10.0.2.2:3000'; // للمحاكي
    // أو
    static const String baseUrl = 'http://192.168.1.100:3000'; // للجهاز الحقيقي

    static const String socketUrl = 'http://10.0.2.2:3000'; // للمحاكي
    // أو
    static const String socketUrl = 'http://192.168.1.100:3000'; // للجهاز الحقيقي
}
```

## 4. بناء التطبيق

```
# بناء التطبيق للتطوير
flutter build apk --debug

# أو بناء للإنتاج
flutter build apk --release

# تشغيل على المحاكى أو الجهاز
flutter run
```

---

## اختبار النظام

---

### 1. اختبار الواجهة الخلفية

```
# تشغيل اختبارات API
cd backend
node test_api.js
```

### 2. اختبار تطبيق Flutter

```
# تشغيل اختبارات الوحدة
cd frontend
flutter test

# تشغيل اختبارات التكامل
flutter drive --target=test_driver/app.dart
```

### 3. اختبار التكامل الكامل

1. تأكد من تشغيل الخادم على المنفذ 3000
  2. شغل تطبيق Flutter
  3. جرب تسجيل الدخول
  4. أنشئ غرفة صوتية جديدة
  5. جرب الميزات المختلفة
-



## 1. نشر الواجهة الخلفية

أ) Heroku

```
# تثبيت Heroku CLI
# إنشاء تطبيق جديد
heroku create hus-backend

# إضافة MongoDB Atlas
heroku addons:create mongolab:sandbox

# رفع الكود
git push heroku main
```

ب) AWS أو DigitalOcean

```
# إنشاء خادم Ubuntu
# تثبيت Node.js و PM2
sudo apt update
sudo apt install nodejs npm
npm install -g pm2

# رفع الكود وتشغيله
pm2 start server.js --name "hus-backend"
```

## 2. نشر تطبيق Flutter

أ) Google Play Store

```
# للنشر AAB بناء
flutter build appbundle --release

# الملف سيكون في: build/app/outputs/bundle/release/
```

```
# macOS يتطلب للنشر IPA بناء  
flutter build ios --release
```

## استكشاف الأخطاء وحلها

### مشاكل شائعة في الواجهة الخلفية

#### 1. خطأ اتصال MongoDB

```
Error: MongoNetworkError: failed to connect to server
```

**الحل:** - تحقق من صحة رابط الاتصال في `env` - تأكد من إضافة IP الخاص بك إلى القائمة البيضاء -  
تحقق من اسم المستخدم وكلمة المرور

#### 2. خطأ منفذ مشغول

```
Error: listen EADDRINUSE: address already in use :::3000
```

**الحل:**

```
# إيجاد العملية التي تستخدم المنفذ  
lsof -i :3000  
  
# إنهاء العملية  
kill -9 <PID>
```

### مشاكل شائعة في Flutter

#### 1. خطأ Gradle

```
Could not resolve all artifacts for configuration ':classpath'
```

**الحل:**

```
cd android
./gradlew clean
cd ..
flutter clean
flutter pub get
```

## 2. خطأ Firebase

```
FirebaseException: No Firebase App '[DEFAULT]' has been created
```

**الحل:** - تأكد من وجود google-services.json في android/app - تحقق من إعدادات Firebase في main.dart

## 3. خطأ اتصال الشبكة

```
SocketException: Failed host lookup
```

**الحل:** - تحقق من عنوان الخادم في app\_constants.dart - تأكد من تشغيل الخادم - للمحاكي استخدم 10.0.2.2 بدلاً من localhost

# 📞 الدعم والمساعدة

## الموارد المفيدة

- [وثائق Flutter](#)
- [وثائق Node.js](#)
- [وثائق MongoDB](#)
- [وثائق Firebase](#)

## الحصول على المساعدة

- [GitHub Issues](#): [رابط المستودع](#)
- [المجتمع](#): Stack Overflow, Reddit r/FlutterDev
- [الوثائق](#): README.md في المستودع

## الخطوات التالية

بعد إعداد المشروع بنجاح، يمكنك:

1. دمج ZegoCloud: لإضافة الصوت الحقيقي
2. تطوير فلتر الكلمات: لتحسين الأمان
3. إضافة ميزات YouTube: للمشاهدة الجماعية
4. تحسين الأداء: وإضافة المزيد من الميزات
5. النشر: على متاجر التطبيقات

## ملاحظات مهمة

- الأمان: لا تشارك مفاتيح API أو كلمات المرور
- النسخ الاحتياطي: احتفظ بنسخة احتياطية من قاعدة البيانات
- التحديثات: تابع تحديثات Flutter و Node.js
- الاختبار: اختبر جميع الميزات قبل النشر

تم إنشاء هذا الدليل بواسطة فريق تطوير HUS - نوفمبر 2024