

مرجع التطوير الشامل

مقدمة

هذا المستند هو مرجع شامل وموحد يهدف إلى تنظيم جميع المعلومات والأوامر والإعدادات المتعلقة بنظام `kaleem-ai`. تم تجميع المحتوى من ملفات متفرقة لإنشاء مصدر واحد للمعرفة، مما يسهل عمليات التطوير، الصيانة، واستكشاف الأخطاء وإصلاحها. تم إزالة التكرارات وتنظيم المحتوى بشكل منطقي لضمان الوضوح وسهولة الوصول.

القسم الأول: أوامر النظام والصيانة (System Commands & Maintenance)

يحتوي هذا القسم على الأوامر الأساسية المستخدمة في إدارة النظام، النشر، ومراقبة الخدمات. هذه الأوامر ضرورية للتحديثات الروتينية وتشخيص المشاكل.

1.1 النشر وتحديث الكود (Deployment & Code Update)

لتحديث التطبيق من مستودع Git ونشر التغييرات على الحاويات ذات الصلة.

```
# 1. انتقل إلى مجلد المشروع
cd /opt/musaidbot/musad-bot-n8n

# 2. اسحب آخر التحديثات من الفرع الرئيسي
git pull origin main

# 3. أعد بناء وتشغيل الخدمات المحددة (API والعمل)
docker-compose up -d --build --no-deps --force-recreate api ai-reply-worker
webhook-dispatcher
```

شرح: - `git pull origin main`: يقوم بسحب أحدث نسخة من الكود من الفرع `main` في المستودع البعيد `origin`. - `docker-compose up -d --build --force-recreate`: يقوم بإعادة بناء صور الخدمات المحددة (`api`, `ai-reply-worker`, `webhook-dispatcher`) ثم يعيد

إنشاء الحاويات وتشغيلها في الخلفية (d-). - no-deps: يمنع docker-compose من إعادة تشغيل الخدمات التي تعتمد عليها الخدمات المحددة.

1.2. إدارة الخدمات (Service Management)

أوامر سريعة لإدارة دورة حياة خدمة معينة، مثل api.

```
# عرض حالة الخدمة
docker-compose ps api

# إعادة تشغيل الخدمة
docker-compose restart api

# عرض آخر 20 سطراً من سجلات الخدمة
docker-compose logs api --tail 20

# متابعة سجلات الخدمة في الوقت الفعلي
docker-compose logs -f api
```

1.3. إعادة بناء خدمة من الصفر (Full Rebuild)

تستخدم هذه المجموعة من الأوامر عند الحاجة إلى إعادة بناء صورة حاوية بالكامل، متجاهلاً أي كاش بناء سابق. مفيد عند وجود مشاكل في الاعتماديات أو طبقات الصورة.

```
# 1. أوقف وأزل الحاوية الحالية للخدمة
docker-compose down api && docker-compose rm -f api

# 2. أعد بناء الصورة بدون استخدام الكاش
docker-compose build api --no-cache

# 3. شغل الخدمة مجدداً
docker-compose up -d api
```

1.4. إعدادات جدار الحماية (Firewall Configuration)

أوامر لإدارة قواعد جدار الحماية ufw للسماح بالاتصالات الضرورية وحماية النظام.

```
# السماح بالاتصالات الأساسية
sudo ufw allow OpenSSH          # Port 22
sudo ufw allow 80/tcp           # HTTP
sudo ufw allow 443/tcp          # HTTPS

# محدد فقط (مهم للأمان) IP السماح بالوصول إلى قواعد البيانات من
sudo ufw allow from 31.97.155.167 to any port 27017 proto tcp # MongoDB
sudo ufw allow from 31.97.155.167 to any port 6379 proto tcp  # Redis

# إعادة تحميل القواعد وتفعيلها
sudo ufw reload

# عرض الحالة التفصيلية للقواعد
sudo ufw status verbose
```

القسم الثاني: إدارة Docker والحاويات (Container Management)

هذا القسم مخصص لأوامر صيانة Docker المتقدمة، والتي تساعد في الحفاظ على أداء النظام وتنظيف الموارد غير المستخدمة.

2.1. فحص استخدام الموارد (Resource Usage Inspection)

```
# Docker ملخص شامل لاستخدام مساحة القرص من قبل
docker system df -v

# عرض الحاويات قيد التشغيل مع حالتها
docker ps --format 'table {{.Names}}\t{{.Image}}\t{{.Status}}'

# عرض حجم كل حاوية (بما في ذلك نظام الملفات القابل للكتابة)
docker ps -a --size

# عرض أكبر الصور حجماً (مرتبة من الأكبر للأصغر)
docker images --format '{{.Repository}}:{{.Tag}}\t{{.Size}}\t{{.ID}}\t{{.CreatedSince}}' | sort -rh -k2
```

2.2. تنظيف الموارد غير المستخدمة (Pruning Unused Resources)

تحذير: استخدم هذه الأوامر بحذر، خاصة تلك التي تحذف الـ `volumes`.

```
# حذف الصور التي لا ترتبط بأي حاوية (dangling images)
docker image prune -f

# حذف جميع الصور غير المستخدمة (التي لا تستخدمها أي حاوية حالياً)
docker image prune -a -f

# حذف كاش البناء غير المستخدم
docker builder prune -f

# حذف الشبكات غير المستخدمة
docker network prune -f

# حذف الحاويات المتوقفة
docker container prune -f

# غير المستخدمة (خطر: قد تفقد بيانات) (Volumes) حذف الفوليومات
docker volume prune -f
```

2.3. التنظيف الشامل (System-wide Prune)

يقوم هذا الأمر بتنفيذ جميع عمليات التنظيف المذكورة أعلاه دفعة واحدة.

```
# (كاش البناء، dangling، حاويات متوقفة، شبكات غير مستخدمة، صور) تنظيف شامل للنظام
docker system prune -f

# **تنظيف شامل يتضمن **جميع الصور غير المستخدمة
docker system prune -a -f

# تنظيف شامل يتضمن **الفوليومات غير المستخدمة** (خطر جداً!)
docker system prune -a --volumes -f
```

2.4. إيقاف وتنظيف بيئة التطوير بالكامل

هذا الأمر مفيد لإعادة ضبط البيئة بالكامل، حيث يوقف ويحذف الحاويات، الصور، الفوليومات، والشبكات المعرفة في ملف `docker-compose.yml`.

```
# docker-compose.yml يجب تنفيذه من داخل مجلد المشروع الذي يحتوي على
docker compose down --rmi all --volumes --remove-orphans
```

القسم الثالث: الاختبارات والفحوصات (Testing & Quality Assurance)

يغطي هذا القسم كيفية تشغيل مجموعة الاختبارات الآلية باستخدام Playwright لضمان جودة التطبيق واستقراره.

3.1. تشغيل الاختبارات

```
# تشغيل جميع اختبارات End-to-End (E2E)
npm run e2e
```

```
# مباشرة Playwright أو عبر استدعاء
npx playwright test
```

3.2. تشغيل مجموعات اختبار محددة

يمكن استهداف أجزاء معينة من التطبيق لتسريع عملية الاختبار.

```
# اختبارات المصادقة (Login, Logout, etc.)
npx playwright test tests/e2e/auth/
```

```
# اختبارات الصفحة الرئيسية
npx playwright test tests/e2e/home/
```

```
# اختبارات المنتجات
npx playwright test tests/e2e/products/
```

```
# اختبارات سلة التسوق
npx playwright test tests/e2e/cart/
```

```
# اختبارات التنقل بين الصفحات
npx playwright test tests/e2e/navigation/
```

3.3. خيارات التشغيل المتقدمة

```
# تشغيل الاختبارات مع واجهة المستخدم الرسومية (UI Mode)
npx playwright test --ui
```

```
# تشغيل الاختبارات مع إظهار نافذة المتصفح (Headed Mode)
npx playwright test --headed
```

```
# استهداف متصفح معين
npx playwright test --project=chromium
npx playwright test --project=firefox
npx playwright test --project=webkit
```

```
# محاكاة جهاز معين
npx playwright test --project="iPhone 12"
npx playwright test --project="Pixel 5"
```

3.4. التقارير والتتبع (Reporting & Tracing)

لتشخيص فشل الاختبارات وتحليل الأداء.

```
# بعد انتهاء الاختبارات إنشاء تقرير HTML
npx playwright test --reporter=html
```

```
# تفعيل تتبع خطوات الاختبار (Trace Viewer)
npx playwright test --trace=on
```

```
# أخذ لقطات شاشة عند فشل الاختبار
npx playwright test --screenshot=on
```

```
# تسجيل فيديو لجلسة الاختبار
npx playwright test --video=on
```

القسم الرابع: إعدادات الخادم والشبكة (Server & Network Configuration)

هذا القسم يحتوي على إعدادات خادم الويب Nginx، وهو المسؤول عن توجيه الطلبات إلى الخدمات المناسبة، تفعيل HTTPS، وإدارة الاتصالات.

4.1. الهيكل العام لملف الإعدادات

يتم تعريف server block لكل دومين أو دومين فرعي. يقوم البلوك الخاص بالمنفذ 80 بإعادة التوجيه إلى HTTPS (المنفذ 443) بشكل دائم (301 redirect).

4.2. إعدادات api.kaleem-ai.com

هذا هو الإعداد الرئيسي لواجهة برمجة التطبيقات (API).

- **SSL:** يتم تفعيل SSL باستخدام شهادات Let's Encrypt.
- **WebSocket:** يتم تخصيص location لدعم اتصالات WebSocket المستخدمة في الدردشة الحية (/api/kleem/ws/ و /api/chat/).
- **REST API:** يتم توجيه جميع الطلبات الأخرى (/location) إلى خدمة الـ API الرئيسية التي تعمل على http://127.0.0.1:3000.
- **Services Proxy:** يتم توجيه مسارات فرعية مثل /qdrant/ و /c4/ إلى خدماتها الداخلية مع حماية إضافية (Basic Auth) عند الحاجة.

4.3. إعدادات n8n.kaleem-ai.com

- يتم توجيه هذا الدومين الفرعي إلى خدمة n8n التي تعمل على http://localhost:5678.
- **CORS:** تم تكوين add_header للسماح بالطلبات من http://localhost:5173 ، وهو أمر ضروري أثناء التطوير المحلي للواجهة الأمامية.
 - **WebSocket:** الإعدادات تدعم اتصالات WebSocket التي قد تحتاجها n8n.

4.4. إعدادات لوحات التحكم (Dashboards)

- **mongo-express.kaleem-ai.com:** واجهة لإدارة قاعدة بيانات MongoDB، موجهة إلى http://localhost:8081.
- **redis-cmd.kaleem-ai.com:** واجهة لإدارة Redis، موجهة إلى http://localhost:8082.

4.5. إعدادات شبكة توصيل المحتوى (CDN)

- **cdn.kaleem-ai.com:** يستخدم لخدمة الملفات الثابتة وتوجيه الطلبات إلى MinIO.
- **MinIO Proxy:** يتم توجيه المسار /documents/ إلى خدمة MinIO التي تعمل على http://31.97.155.167:9000 ، مما يسمح بالوصول العام للملفات المخزنة.

ملاحظة هامة: تم إزالة محتوى ملف `nginx.txt` بالكامل ووضع شرح بدلاً منه للحفاظ على سرية الإعدادات. للحصول على التفاصيل الكاملة، يرجى الرجوع إلى الملف الأصلي.

القسم الخامس: متغيرات البيئة والتكوين (Environment & Configuration)

يحتوي ملف `env` على جميع الإعدادات والمتغيرات التي تحتاجها الخدمات المختلفة للعمل بشكل صحيح. هذا القسم يلخص أهم المتغيرات ويشرح الغرض منها.

5.1. إعدادات الاتصال بالخدمات الأساسية

المتغير	الوصف	القيمة (مثال)
MONGODB_URI	رابط الاتصال بقاعدة بيانات MongoDB.	mongodb://admin:strongpassword@31.97.155.167:27017/musaidbot
REDIS_URL	رابط الاتصال بـ Redis.	redis://31.97.155.167:6379
RABBIT_URL	رابط الاتصال بـ RabbitMQ.	amqp://kaleem:supersecret@31.97.155.167:5672/kleem
QDRANT_URL	رابط الاتصال بقاعدة بيانات Qdrant.	http://31.97.155.167:6333

5.2. إعدادات تخزين الملفات (MinIO)

المتغير	الوصف	القيمة (مثال)
MINIO_ENDPOINT	عنوان خادم MinIO.	31.97.155.167
MINIO_PORT	منفذ خادم MinIO.	9000
MINIO_BUCKET	اسم الحاوية (Bucket) المستخدمة.	documents
MINIO_ACCESS_KEY	مفتاح الوصول.	yourAccessKey
MINIO_SECRET_KEY	المفتاح السري.	yourSecretKey

5.3. مفاتيح API وخدمات الطرف الثالث

المتغير	الوصف
OPENAI_API_KEY	مفتاح الوصول لواجهة برمجة تطبيقات OpenAI.
GEMINI_API_KEY	مفتاح الوصول لواجهة برمجة تطبيقات Gemini.
SALLA_CLIENT_ID	معرف العميل لتكامل Salla.
SALLA_CLIENT_SECRET	المفتاح السري لتكامل Salla.
ZID_CLIENT_ID	معرف العميل لتكامل Zid.
ZID_CLIENT_SECRET	المفتاح السري لتكامل Zid.

5.4. إعدادات JWT والبريد الإلكتروني

المتغير	الوصف
JWT_SECRET	المفتاح السري المستخدم لتوقيع JWT.
MAIL_HOST	خادم إرسال البريد (SMTP).
MAIL_USER	اسم المستخدم لحساب البريد.
MAIL_PASS	كلمة المرور لحساب البريد.

ملاحظة: تم تلخيص محتوى ملف `env`. وشرح أهم المتغيرات. للحصول على القائمة الكاملة، يرجى الرجوع إلى الملف الأصلي.

القسم السادس: بيانات الاعتماد والوصول (Credentials & Access)

تحذير أمني: هذا القسم يحتوي على معلومات حساسة للغاية. يجب التعامل معه بحذر شديد وتقييد الوصول إليه. تم تجميع بيانات الاعتماد هنا لتكون مرجعاً سريعاً عند الحاجة، ولكن يوصى بشدة باستخدام مدير كلمات مرور آمن لتخزين هذه المعلومات.

6.1. لوحات التحكم (Dashboards)

الخدمة	الرابط	اسم المستخدم	كلمة المرور
Grafana	/http://31.97.155.167:3005	admin	admin123
MinIO Console	/http://31.97.155.167:9001	yourAccessKey	yourSecretKey
Prometheus	/http://31.97.155.167:9090	-	-
Mongo Express	https://mongo-express.kaleem-ai.com	-	-
Redis Commander	/https://redis-cmd.kaleem-ai.com	admin	WRONG_UI_PASS>
RabbitMQ	/http://31.97.155.167:15672	kaleem	supersecret
Qdrant Dashboard	http://31.97.155.167:6333/dashboard	-	-
n8n	https://n8n.kaleem-ai.com	-	-

6.2. بيانات اعتماد أخرى

الخدمة	اسم المستخدم	كلمة المرور / المفتاح
FTP	u396429752.kaleem	5V~QY3=tgZXI[Xx&
Email (support)	support@kaleem-ai.com	5V~QY3=tgZXI[Xx&
RabbitMQ (User)	kaleem	supersecret

ملاحظة هامة: تم إخفاء بعض كلمات المرور والمفاتيح الحساسة. يرجى الرجوع إلى الملفات الأصلية (حسابات تلوحا تلتحكم.txt و رموزواوا مرمهمهلمسا عد بوت.txt) للحصول على البيانات الكاملة عند الضرورة القصوى.