وثيقة مواصفات وتوثيق رسمي

التوثيق التقني والتشغيلي الشامل

Project Desk — المعمارية والبيانات والواجهات والأمن والتشغيل والجودة

معرّف الوثيقة PD-TECH-DOCS	**الإصدار** 1.0
تاريخ خط الأساس 2026-08-12	**التصنيف** مرجع داخلي / قابل للتسليم

مبنية على التنفيذ الفعلي للمستودع ومميّزة بوضوح بين المنفذ والجزئي والمخطط.

خريطة المحتويات

الأقسام الرئيسية والفرعية للوثيقة.

DOCUMENTATION INDEX

1. طريقة الاستخدام
2. الملفات العشرة
3. مسارات قراءة مقترحة
4. خريطة التنفيذ والوثيقة الحاكمة
5. قرارات وحدود لا يجوز فقدها
6. مصادر الشيفرة التي راجعتها الحزمة
7. وثائق مرتبطة خارج الحزمة
8. قواعد تحديث الوثائق
9. فحص اتساق سريع
10. سجل التغيير

SYSTEM OVERVIEW

1. الغرض والنطاق
2. تصنيف حالة الوظائف
3. المستخدمون والأدوار
4. رحلات العمل الأساسية
5. خريطة الواجهة
6. حدود مهمة
7. مؤشرات الصحة
8. روابط الوثائق التقنية

ARCHITECTURE

1. القرار المعماري
2. الحزم والإصدارات الأساسية
3. Bootstrap والـmiddleware
4. حدود الوحدات المنفذة
5. معاملات واتساق وتزامن
6. الزمن والتقويم
7. نمط الواجهة والحالة
8. البحث والقراءات

9. Adapter boundaries
10. قرارات وحدود معلنة

DATA MODEL

1. اصطلاحات
2. مخطط العلاقات العالي
3. الهوية والبنية الأساسية
4. الدليل والمشاريع
5. المتطلبات والعمل
6. التخطيط والاجتماعات
7. الحوكمة
8. الملفات وكراسة المتطلبات
9. قوالب الفواتير والبيانات التاريخية
10. العمليات والتدقيق
11. سياسات الحذف وFK
12. فهارس مهمة

API AND ROUTES

1. العقد العام
2. مسارات الواجهة العليا
3. المشاريع والعملاء والفريق
4. المهام والمتطلبات
5. التخطيط والحوكمة
6. الوثائق
7. قوالب الفواتير
8. مركز البيانات
9. النسخ والاستعادة
10. الإعدادات والمصادقة
11. مثال عميل JSON داخلي

SECURITY AND PERMISSIONS

1. نموذج الثقة
2. المصادقة والجلسات
3. مصفوفة الصلاحيات العامة
4. Object-level authorization
5. حماية الإدخال والملفات

6. أمن النسخ والاستعادة
7. ترويسات المتصفح وTLS
8. التدقيق والخصوصية
9. قائمة hardening الإنتاج
10. تهديدات وحدود متبقية

OPERATIONS AND RECOVERY

1. الغرض والنطاق
2. مكونات التشغيل ومواقع البيانات
3. العمليات المجدولة والأوامر
4. روتين التشغيل
5. دورة النسخة .pdesk
6. الملفات والفحص والتنظيف
7. الاستيراد والتصدير كعملية تشغيلية
8. سيناريوهات التعافي
9. المراقبة والتنبيه
10. اختبار التعافي وقائمة الإغلاق

TESTING AND QA

1. سياسة الدليل
2. بيئة الاختبار
3. أوامر الجودة

Backend: format check + PHPStan + PHPUnit

كل بوابات الشيفرة باستثناء build/browser

اختبارات Laravel فقط أثناء التشخيص

Frontend

Browser؛ يتطلب تطبيقاً محلياً وقاعدة اختبار seeded

4. خريطة تغطية Backend
5. تدفقات المتصفح
6. خط CI الفعلي
7. استراتيجية بيانات الاختبار
8. بوابات القبول قبل الدمج
9. قبول الإصدار
10. فجوات وخطة تحسين QA

11. تشخيص فشل الاختبار

DEVELOPER GUIDE

1. نقطة البداية
2. المتطلبات المحلية
3. إعداد نسخة تطوير جديدة
4. خريطة المستودع
5. دورة الطلب
6. إضافة ميزة مجال جديدة
7. البيانات والمعاملات
8. الوقت والتقويم
9. الملفات
10. قوالب الفاتورة
11. React وInertia
12. البحث والتنبيهات
13. الأمن والأسرار
14. أوامر يومية

فحص routes والمخطط

الجودة

تشغيل مهام المجال يدوياً للتشخيص

15. تشخيص التطوير
16. تعريف الإنجاز للتغيير

DEPLOYMENT

1. نموذج النشر المدعوم
2. متطلبات الخادم
3. بناء artifact
4. إعدادات البيئة
5. نظام الملفات والصلاحيات
6. إعداد خادم الويب
7. scheduler والخدمات
8. أول نشر
9. ترقية إصدار قائم
10. Smoke tests بعد النشر

11. الرجوع

12. المراقبة والسعة

13. hardening قبل Go-Live

14. قائمة إصدار نهائية

DOCUMENTATION GOVERNANCE

1. الغرض

2. مجموعة الوثائق الحاكمة

3. مصدر الحقيقة

4. قواعد التغيير

5. تعريف اكتمال التوثيق

6. دورة المراجعة

DOCUMENTATION INDEX

1. طريقة الاستخدام

هذه الحزمة توثق التنفيذ الذي تمت مراجعته من شيفرة Project Desk بتاريخ 2026-08-12، ولا تمثل وحدها رقم إصدار أو شهادة قبول. ابدأ بنظرة النظام، ثم اختر الدليل حسب دورك. عند اختلاف وثيقة مع التنفيذ تكون الأولوية: migrations وroutes/Requests/Policies/Services ثم الاختبارات القابلة للتشغيل، ثم هذه الوثائق، ثم المستندات التاريخية.

رموز الحالة المستخدمة:

الحالة	المعنى
منفذ `implemented`	يوجد مسار شيفرة حقيقي واختبارات/عقد مناسب بحسب الجرد؛ يلزم مع ذلك CI حديث للإصدار
جزئي `partial`	جزء عامل، لكن التكامل أو التغطية أو السلوك الموحد غير مكتمل
مخطط `planned`	نية/توصية فقط، ليست وعداً ولا يجوز عرضها كوظيفة حالية
غير مدعوم	خارج حدود الإصدار الحالي أو يحتاج إعادة تصميم

2. الملفات العشرة

الملف	الجمهور	المحتوى
SYSTEM_OVERVIEW_AR.md	الجميع/المنتج	هدف النظام، الوحدات، الأدوار، الرحلات والحدود
ARCHITECTURE_AR.md	المعماريون والمطورون	modular monolith، الطبقات، الطلب، الوقت، الحالة والتكاملات
DATA_MODEL_AR.md	المطورون/DBA	الجداول والنماذج والعلاقات والقيود والأرشفة والتزامن
API_AND_ROUTES_AR.md	Frontend/Backend/QA	routes الواجهة وvalidation، payloads، HTTP، الأخطاء والاستجابات
SECURITY_AND_PERMISSIONS_AR.md	الأمن/المنتج/المطورون	المصادقة، matrix الصلاحيات، الملفات، الأسرار والتدقيق والمخاطر
OPERATIONS_AND_RECOVERY_AR.md	التشغيل/الدعم	scheduler، النسخ `.pdesk` وlegacy، الاستعادة، الملفات، المراقبة وrunbooks
TESTING_AND_QA_AR.md	QA/المطورون/الإصدار	الجرد، الأوامر، gates، browser، CI، الفجوات والتشخيص
DEVELOPER_GUIDE_AR.md	المطورون	الإعداد، بنية المستودع، إضافة ميزة، معاملات/وقت/ملفات/React
DEPLOYMENT_AR.md	DevOps/التشغيل	filesystem، Nginx، env، build، النشر والترقية والرجوع وGo-Live
DOCUMENTATION_INDEX_AR.md	الجميع	الفهرس، أثر المتطلبات، الملكية وقواعد تحديث التوثيق

3. مسارات قراءة مقترحة

المهمة	اقرأ بالترتيب
فهم المنتج وحدوده	System Overview ثم Security ثم API
بدء تطوير	Developer Guide ثم Architecture ثم Data Model ثم Testing
بناء واجهة/تكامل داخلي	API and Routes ثم Security ثم Data Model
تدقيق أمني	Security ثم API ثم Operations ثم Deployment
نشر أول مرة	Deployment ثم Operations ثم Testing ثم Security
حادث أو استعادة	Operations and Recovery ثم Deployment ثم Data Model
قرار إضافة محاسبة	System Overview وحدود Sales ثم SRS؛ الوظيفة الحالية templates فقط
ترحيل لمساعد AI/فريق جديد	الملفات العشرة مع migrations/routes/config/tests وملف البيئة المثال دون أسرار

4. خريطة التنفيذ والوثيقة الحاكمة

المجال	الوضع المختصر	المرجع الأول
مشاريع ومهام ممتدة بين بداية ونهاية	منفذ؛ جدول أسبوعي ببداية الأحد افتراضياً	API، Data Model، Architecture
عملاء وجهات اتصال وفريق	منفذ مع أدوار عامة وعضوية مشروع	API، Security
كراسة متطلبات وإصدارات	منفذ	Data Model، API، Operations للملفات
اجتماعات ومحاضر وجدولة	منفذ، وتنبيهات scheduler	API، Operations
مخاطر وقضايا ووثائق	منفذ	API، Data Model
ملفات خاصة وفحص malware	منفذ؛ production fail-closed مطلوب	Security، Operations
استيراد/تصدير CSV/XLSX	منفذ لموارد محددة ومعاملة ذرية	API، Operations
بحث داخلي	منفذ ومقيد بالصلاحيات	Architecture، Security، Developer Guide
قوالب فواتير وPDF	منفذ، غير محاسبي	System Overview، API، Developer Guide
proposal/receipt/letterhead legacy	مخزن تاريخياً لكنه مخفي/غير مسار منتج	Data Model، API
نسخة `.pdesk` كاملة مشفرة	منفذ لـSQLite/الملفات	Operations، Security
legacy DB backup	جزئي، database-only	Operations
إشعارات المهام والاجتماعات	منفذ عبر scheduler	Operations، Developer Guide
queue/background Jobs مجال	غير منفذ؛ scaffold فقط	Architecture، Developer Guide
إعداد التقويم والمنطقة في كل النظام	جزئي؛ بعض المستهلكين يعتمدون config ثابتاً	Architecture، Testing

المجال	الوضع المختصر	المرجع الأول
تعدد المضيفين/HA	غير مدعوم في v1	Deployment، Architecture

5. قرارات وحدود لا يجوز فقدها

1. **قوالب الفواتير ليست نظاماً محاسبياً.** لا توجد دفعات أو أرصدة أو دفتر أستاذ أو قيود أو ضرائب/تحصيل محاسبي.
2. v1 يستخدم SQLite وprivate local storage وأقفالاً محلية؛ النشر الأفقي غير آمن دون إعادة تصميم.
3. datetimes تخزن/تقارن UTC ويعرض الوقت التجاري افتراضياً `Africa/Tripoli`. الأسبوع يبدأ الأحد والعطلة الجمعة/السبت في builder الحالي.
4. إعداد calendar/timezone موجود لكنه لا يغذي كل الخدمات بالتساوي؛ الحالة `partial`.
5. لا توجد Jobs مخصصة أو outbox؛ العمليات الحالية متزامنة أو أوامر scheduler، وActivityLogger متزامن غالباً.
6. global `viewer` مع project pivot `manager` حالة غير موحدة: بعض تحديثات المشروع قد تسمح بها العضوية، بينما رفع الملفات يمنع viewer صراحة. هذه فجوة قرار/اختبار، لا قاعدة مستقرة.
7. التنزيل لا يسمح إلا لملف clean وبصلاحية صحيحة؛ production يجب أن يفشل مغلقاً إن لم يوجد ماسح.
8. restore خطير ومدير فقط مع recent password وعبارة ثابتة وchecksum وnonce وقفل وصيانة وrollback.

6. مصادر الشيفرة التي راجعتها الحزمة

المصدر	ما تم استخراجه
`composer.json`, `package.json`, lockfiles	المنصة والإصدارات والأوامر
`bootstrap/app.php`, providers، `config/*`, `.env.example`	adapters، `/up`، middleware والحدود بلا قيم سرية
`database/migrations`, models/relations	المخطط والقيود والأرشفة/lock versions
`routes/*.php`, Controllers, Form Requests	routes وpayloads والاستجابات
Policies وServices	authorization والمعاملات والحسابات والنسخ/الملفات
Commands و`routes/console.php`	scheduler والعمليات الدورية
`resources/js/pages/components/hooks/layouts`	صفحات Inertia والحالة وRTL/unsaved guard
Tests وCI	تغطية قابلة للتنفيذ وبوابات الجودة والفجوات

الجرد في لحظة المراجعة: 25 service، 31 policy، 17 controller، 31 model، دون `app/Jobs`. أرقام الجرد أداة لاكتشاف الانجراف وليست contract دائم.

7. وثائق مرتبطة خارج الحزمة

توجد وثائق أخرى في `docs/` ويجب التعامل معها كمراجع داعمة والتحقق منها أمام الشيفرة:

- دليل المستخدم العربي
- حوكمة التوثيق
- نطاق المنتج
- البيئة
- المرفقات والاحتفاظ
- النسخ والاستعادة
- جاهزية الإصدار
- تتبع المتطلبات
- خارطة الطريق
- Workflow Status API

تحذير: وثيقة architecture تاريخية قد تذكر `SaveTaskAction` أو outbox؛ التنفيذ الفعلي يستخدم `TaskService` ولا يملك outbox. لا تنقل الادعاء التاريخي إلى SRS أو تشغيل حالي.

8. قواعد تحديث الوثائق

حدث الوثائق في نفس pull request عند أي مما يلي:

التغيير	الملفات الدنيا
route أو payload/error	Testing، API، وSRS/traceability
model/migration/relation	Data Model، Architecture، backup compatibility
Policy/role/membership	API، Security، اختبارات matrix ودليل المستخدم
ملف/MIME/scanner/reference	Testing، Data Model، Operations، Security
backup/restore/encryption	Deployment، Security، Operations واختبار drill
scheduler/Job/queue	Testing، Deployment، Operations، Architecture
timezone/week/calendar	Testing، Developer، Architecture ودليل المستخدم
صفحة/رحلة UI	browser tests، API، System Overview ودليل المستخدم
نطاق invoice/sales	System Overview وAPI وSRS؛ حافظ على وسم غير محاسبي إلا بقرار منتج
متطلبات runtime/env	Environment، Deployment، Developer وCI

كل تحديث يجب أن يسجل: السبب، الحالة قبل/بعد، migration/compatibility، الأثر الأمني/التشغيلي، الاختبارات، وقرار rollback. لا تضع secret example فعلياً.

9. فحص اتساق سريع

قبل نشر الحزمة أو تسليمها لمساعد AI آخر:

- [ ] الملفات العشرة موجودة وروابطها تعمل.
- [ ] route list وmigrations وPolicies تطابق جداول API/data/security.
- [ ] لا ادعاء عن outbox/Jobs/HA/CSP عامة غير موجودة.
- [ ] invoice موصوف كقالب غير محاسبي في كل موضع.
- [ ] `.pdesk` وlegacy database-only والفحص fail-closed موثقة بلا مفاتيح.
- [ ] UTC وTripoli والأحد والجمعة/السبت وحدود التكامل الجزئي ظاهرة.
- [ ] أوامر setup/test/deploy لا تحتوي credential ولا أمر مسح إنتاج.
- [ ] نتائج الاختبار مرتبطة بـcommit وتشغيل حديث، لا برقم قديم فقط.
- [ ] diagrams إن وجدت يسبقها/يتبعها وصف نصي كامل ولا يعتمد الفهم عليها.
- [ ] الترميز UTF-8 والعربية وMarkdown tables/code fences سليمة.

10. سجل التغيير

التاريخ	النطاق	الملاحظة
2026-08-12	إنشاء الحزمة التقنية العربية	توثيق code-first للتنفيذ الحالي مع تمييز المنفذ/الجزئي/المخطط والحدود

SYSTEM OVERVIEW

> مرجع تقني مستخرج من التنفيذ الفعلي — 12 أغسطس 2026

> حالة الوثيقة: تصف الإصدار الموجود في المستودع، ولا تفترض خصائص غير منفذة.

1. الغرض والنطاق

Project Desk تطبيق ويب داخلي أحادي الشركة لإدارة العمل والمشاريع باللغة العربية وباتجاه RTL. يجمع في نشر واحد:

- المشاريع، العملاء، جهات الاتصال، الفريق وعضويات المشاريع؛
- المتطلبات وكراسة المواصفات ذات الإصدارات؛
- المهام وجدولها وسجل الإسناد والجدول الأسبوعي؛
- الخط الزمني والاجتماعات والحضور والمحاضر؛
- المخاطر والمشكلات والمرفقات وسجل النشاط؛
- البحث والتنبيهات والإعدادات وحالات سير العمل؛
- استيراد/تصدير CSV وXLSX، ملخصات PDF، والنسخ والاستعادة؛
- **قوالب فواتير غير محاسبية** مع بنود وحساب معاينة وPDF.

لا توجد واجهة API عامة مستقلة؛ صفحات Inertia وبعض نقاط JSON الداخلية تخدم التطبيق نفسه بجلسة Laravel وسياساته.

2. تصنيف حالة الوظائف

المجال	الحالة	الوصف والحدود
الهوية والوصول	منفذ	دخول داخلي، تحقق البريد، استعادة كلمة المرور، Passkeys، 2FA، تعطيل/أرشفة الحساب، جلسات مشفرة اختيارياً
المشاريع والعملاء والفريق	منفذ	CRUD وأرشفة واستعادة وصلاحيات على مستوى السجل والمشروع
المهام والمتطلبات	منفذ	تواريخ إلزامية للمهمة، حالات قابلة للتهيئة، روابط مهمة/متطلب، قفل تفاؤلي
التخطيط والاجتماعات	منفذ	خط زمني، اجتماع كمصدر واحد مرتبط ببند زمني، حضور ومحضر ومرفق اختياري
كراسة المتطلبات	منفذ	ملف بإصدارات، حالة، إصدار حالي واحد، أرشفة واستعادة
المخاطر والمشكلات	منفذ	مخاطر 1–5، معالجة؛ مشكلات بشدة وحل مطلوب عند الإغلاق
الملفات	منفذ جزئياً تشغيلياً	تحقق بنيوي، تخزين خاص، عقد ماسح. يتطلب ماسح malware حقيقياً في الإنتاج
قوالب الفواتير	منفذ	إنشاء/تعديل/نسخ/أرشفة/استعادة/PDF؛ العميل والمشروع والتواريخ سياق معاينة اختياري
المحاسبة والتحصيل	خارج النطاق	لا أرصدة، قيود، مدفوعات، حالات تحصيل، أعمار ديون أو تقارير مالية
عروض/إيصالات/خطابات	بيانات تاريخية فقط	جداول/نماذج legacy محفوظة للتوافق؛ غير قابلة للعرض أو الربط عبر مسارات القوالب
مركز البيانات	منفذ	استيراد clients/tasks فقط؛ تصدير clients/projects/tasks؛ معاينة ثم commit ذري
النسخ والاستعادة	منفذ لملف v1	SQLite وحزمة `.pdesk` مشفرة مع الملفات؛ legacy SQLite جسر ترحيل database-only
التنبيهات	منفذ	Inbox دائم للمهام والاجتماعات، مزامنة مجدولة، تفضيلات شخصية تقيد سياسة المدير
النشر متعدد العقد	مخطط/غير مدعوم	ملف v1 هو خادم Linux واحد وSQLite WAL وتخزين محلي خاص
Public API / Webhooks	غير منفذ	نقاط JSON داخلية فقط ولا يوجد عقد تكامل خارجي أو رموز API

3. المستخدمون والأدوار

الأدوار العامة المخزنة في `users.global_role` هي:

الدور	الاستخدام المتوقع
`admin`	إدارة كاملة للشركة، المستخدمين، الإعدادات، مركز البيانات، النسخ والاستعادة، وكل المشاريع والقوالب
`project_manager`	إنشاء مشاريع وعملاء؛ إدارة المشاريع التي يديرها؛ إدارة قوالب الفواتير التي أنشأها
`member`	قراءة المشاريع التي ينتمي إليها، ورفع الملفات حسب دور المشروع، وتغيير حالة المهمة المسندة إليه
`viewer`	المقصود قراءة فقط ولا يرفع ملفات؛ لكن عضوية مشروع بدور `manager` تستطيع حالياً عبور بعض Policies للتعديل، وهي فجوة اتساق يجب حسمها

تضيف عضوية المشروع دوراً محلياً `viewer` | `member` | `manager` وحالة `active`. صلاحية سجل بعينه لا تُستنتج من الدور العام وحده؛ تطبق Policies وscopes للرؤية. العضوية لا تضيق الدور العام دائماً في التنفيذ الحالي: pivot `manager` قد يرفع global `viewer` في تحديث المشروع/موارده، بينما `uploadFile` يمنعه صراحة؛ لذلك لا تعتمد على وصف «قراءة فقط» قبل توحيد السياسات واختبارها.

4. رحلات العمل الأساسية

4.1 من العميل إلى التنفيذ

1. ينشئ المدير عميلاً وجهات اتصال.
2. ينشئ مشروعاً ويختار العميل والحالة والمدير والفريق.
3. يسجل المتطلبات أو يرفع إصداراً من كراسة المتطلبات.
4. ينشئ المهام ببداية ونهاية، ويمكن ربطها بالمتطلبات وإسنادها لعضو نشط.
5. تظهر المهام والاجتماعات في التخطيط والجدول الأسبوعي؛ تعرض لوحة المتابعة مؤشرات مأخوذة من البيانات نفسها.
6. تسجل المخاطر والمشكلات، ويربط الملف بالمشروع أو مهمة أو متطلب داخله.

4.2 اجتماع ومحضر

الاجتماع ليس نسخة منفصلة من الجدول: تنشئ الخدمة `timeline_entries.kind=meeting` ثم سجل `meetings` واحداً مرتبطاً به، وتحفظ الحضور. المحضر واحد لكل اجتماع، ويقبل ملخصاً وقرارات وإجراءات ومرفقاً آمناً اختيارياً.

4.3 قالب فاتورة

ينشئ Admin أو Project Manager قالباً مستقلاً كمسودة. رقم القالب يولده الخادم، ويحتوي بنداً واحداً على الأقل، وعملة من LYD/USD/EUR، وخصماً وضريبة ضمن 0–100. يمكن تحديد عميل/مشروع/تاريخ للمعاينة فقط. الإخراج PDF موسوم كمسودة/مؤرشف؛ لا ينشئ مستحقاً أو دفعة.

5. خريطة الواجهة

المسار	صفحة React	الوظيفة
`/dashboard`	`dashboard.tsx`	مؤشرات، قوائم تدخل، توزيع، عبء فريق وجدول أسبوعي
`/projects`	`projects/index.tsx`	قائمة/فلاتر وإنشاء مشروع
`/projects/{id}`	`projects/show.tsx`	تبويبات النظرة العامة والمتطلبات والمهام والخط الزمني والاجتماعات والمخاطر والمشكلات والفريق والوثائق والعميل والنشاط
`/tasks`	`tasks/index.tsx`	قائمة وكانبان، فلاتر، إنشاء وتعديل المهمة وسجل الإسناد
`/clients` وما تحته	`clients/*`	قائمة وإنشاء وتعديل وتفاصيل العميل وجهات اتصاله
`/team`	`team/index.tsx`	إدارة المستخدمين وعرض عبء العمل المرئي
`/sales`	`sales/index.tsx`	مكتبة ومحرر ومعاينة قوالب الفواتير فقط
`/data-center`	`data-center/index.tsx`	import/export والنسخ والاستعادة؛ Admin فقط
`/settings`	`settings/index.tsx`	إعدادات الشركة والتنبيهات والنسخ والتقويم وسير العمل؛ Admin فقط
`/settings/profile`	`settings/profile.tsx`	ملف المستخدم
`/settings/security`	`settings/security.tsx`	كلمة المرور و2FA وPasskeys

المسار	صفحة React	الوظيفة
`/settings/notifications`	`settings/notifications.tsx`	تفضيلات التنبيه الشخصية

الواجهة تستخدم Inertia forms و`fetch` لنقاط JSON، وتعرض flash عبر Sonner. الخطاف `use-unsaved-changes` يحمي التنقل وإغلاق الحوارات و`beforeunload` عند وجود مسودة غير محفوظة. المظهر light/dark/system محفوظ في `localStorage` وCookie، وليس هناك store عالمي خارجي.

6. حدود مهمة

- الحذف الفيزيائي ليس مسار العمل المعتاد؛ الأرشفة تحفظ التاريخ. تنظيف الملفات اليتيمة استثناء مضبوط بعد مهلة.
- البحث قاعدة بيانات محلية بحد خمسة نتائج لكل نوع، وليس محرك بحث خارجي.
- PDF وXLSX/CSV تنفذ تزامنياً في الطلب الحالي؛ لا توجد Jobs تطبيقية في `app/Jobs` حالياً، رغم تهيئة queue database.
- إعداد التقويم قابل للتخزين، لكن builder الأسبوعي الحالي يستخدم الأحد كبداية والجمعة/السبت عطلة من `config/project-desk.php`؛ راجع قسم الزمن في دليل المعمارية.
- ملف الإنتاج المدعوم v1 هو خادم واحد؛ MySQL/PostgreSQL موجودان في إعداد Laravel العام لكن نسخ/استعادة Project Desk المباشرة ترفض غير SQLite.

7. مؤشرات الصحة

- `GET /up`: صحة Laravel الأساسية.
- نجاح scheduler كل دقيقة، وآخر `DataJob` للنسخ، ونتائج مزامنة التنبيهات مؤشرات تشغيلية لازمة.
- أحداث `activity_logs` تحمل `request_id` و`correlation_id` لتتبع الطلب.
- سجل `storage/logs/laravel.log` قناة التطوير الافتراضية؛ يجب ضبط التدوير والمراقبة في الإنتاج.

8. روابط الوثائق التقنية

ابدأ من فهرس التوثيق، ثم راجع المعمارية، نموذج البيانات، المسارات والعقود، والأمن والصلاحيات.

ARCHITECTURE

> يصف هذا الملف البنية المنفذة، مع تمييز الحدود التشغيلية التي لم تُغلق بعد.

1. القرار المعماري

النظام **Modular Monolith** في مستودع ونشر واحد:

طبقة	التقنية/المسؤولية
المتصفح	React 19 + TypeScript + Inertia 3 + Tailwind CSS 4 + Radix primitives، RTL عربي
النقل	مسارات Laravel web، جلسة وInertia responses، CSRF وJSON داخلي وملفات تنزيل
HTTP	Form Requests، Controllers، Middleware، Policies/Gates
التطبيق	Services ومعاملات وتدقيق وقفل تفاؤلي وحسابات
المجال/البيانات	Eloquent Models، SQLite WAL، جداول علائقية وJSON محدود
المخرجات	تخزين محلي خاص، CSV streamed، PhpSpreadsheet، mPDF
التشغيل	database cache/session/queue، Scheduler، نسخ `.pdesk`، سجل Laravel

المسار النصي للطلب هو: **المتصفح** ← ← **Controller** ← **Form Request/Policy** ← **Route binding** ← **Web middleware** ← **Inertia** ← **Eloquent/Storage** ← **Service/transaction** أو **JSON** أو **تنزيل**.

> **مخطط Mermaid:** المصدر النصي محفوظ أدناه، ويُقرأ مع الشرح النصي أو الجدولي المحيط به.

```
flowchart LR
    B["React / Inertia"] --> M["Laravel Web Middleware"]
    M --> R["Routes + Form Requests + Policies"]
    R --> C["Controllers"]
    C --> S["Application Services"]
    S --> D[("SQLite WAL")]
    S --> F["Private File Storage"]
    S --> O["PDF / XLSX / CSV"]
    Q["Scheduler"] --> S
```

الرسم تلخيص فقط؛ الجدول السابق يحدد مسؤولية كل عقدة كاملة، ويظل التطبيق والخلفية في عملية نشر واحدة.

2. الحزم والإصدارات الأساسية

- PHP `^8.3` وLaravel `^13.17`؛
- `inertiajs/inertia-laravel ^3.0` و`@inertiajs/react ^3.0.0`؛
- Vite `^5.7.2`، TypeScript `^19.2.0`، React/React DOM `^8`؛
- 2FA، Laravel Passkeys، Fortify؛
- mPDF `^8.3` وPhpSpreadsheet `^5.9`؛
- Larastan، PHPUnit 12، مستوى 7، ESLint، Pint وPrettier؛
- Node 22.12+ و11.16.0 pnpm.

3. Bootstrap والـmiddleware

`bootstrap/app.php` يسجل stack الويب بهذا الترتيب الإضافي:

1. `RequestContext`: يقبل معرفات منقحة أو يولد UUID، ويعيد `X-Request-Id` و`X-Correlation-Id`.
2. `HoldRestoreReadLock`: يأخذ قفل قراءة مشتركاً؛ أثناء restore يعيد 423.
3. `HandleInertiaRequests`: يشارك المستخدم والقدرات والتنبيهات وحالة sidebar.
4. preload headers للأصول.
5. `SecurityHeaders`: referrer/permissions/COOP/CORP، frame deny، nosniff، وHSTS في production HTTPS.
6. `AuthenticateSession`: يربط الجلسة بتغير بيانات المصادقة.

مجموعة الأعمال محمية بـ`auth`, `verified`, `active`. `EnsureActiveUser` يسجل خروج الحساب المعطل/المؤرشف ويبطل الجلسة والـCSRF token.

4. حدود الوحدات المنفذة

الوحدة	الجداول/النماذج الرئيسية	خدمات محورية
Identity	users, passkeys, sessions	Fortify، `UserSessionSecurity`
Directory	clients, contacts	Controllers، scopes `visibleTo/manageableBy`
Projects/Work	projects, project_members, tasks, assignment_events	`TaskService`, `ProjectMetrics`, `WeeklyScheduleBuilder`
Requirements	requirements, requirement_task, books/versions	`RequirementService`, `RequirementBookService`
Planning	timeline_entries, meetings, attendees, minutes	`MeetingService`
Governance	risks, issues	`OptimisticLock` + Controllers
Documents	file_objects, attachment_links	`ProjectFileService`, scanner, retention
Invoice Templates	sales_documents, sales_line_items	`SalesDocumentService`, calculator, PDF
Data Operations	data_jobs, import_errors	CSV/XLSX services، backup services
Audit/Notifications	activity_logs, notifications	`ActivityLogger`, `NotificationCenterService`
Settings/Workflow	system_settings, workflow_statuses	corresponding services

الاعتماد النصي: الهوية والدليل يغذيان المشاريع؛ المشاريع هي حد الصلاحية الذي تعتمد عليه المهام والمتطلبات والتخطيط والحوكمة والوثائق؛ لوحة المتابعة والبحث تقرآن من الجميع دون إنشاء مصدر حقيقة بديل؛ مركز البيانات يستدعي خدمات البيانات؛ التدقيق يكتب مع عملية المجال.

> **مخطط Mermaid:** المصدر النصي محفوظ أدناه، ويُقرأ مع الشرح النصي أو الجدولي المحيط به.

```
flowchart TD
    I["Identity"] --> P["Projects"]
    D["Directory"] --> P
    P --> W["Tasks"]
    P --> R["Requirements"]
    P --> T["Timeline + Meetings"]
    P --> G["Risks + Issues"]
    P --> F["Documents"]
    D --> V["Invoice Templates"]
    P -. "optional preview" .-> V
    W --> X["Dashboard / Search / Notifications"]
    T --> X
    R --> X
```

الأسهم المتقطعة تعني سياق معاينة اختياري فقط؛ ربط قالب الفاتورة بمشروع لا يمنح الوصول للقالب ولا يجعل القالب معاملة مشروع محاسبية.

5. معاملات واتساق وتزامن

- عمليات الإنشاء/التعديل المركبة تستخدم `DB::transaction` وتكتب `activity_logs` داخل المعاملة نفسها.
- `lockForUpdate` يحمي السجل أو التسلسل عند التعديل؛ SQLite مضبوط على `transaction_mode=IMMEDIATE` و`busy_timeout=5000`.
- `lock_version` يطبق على المشاريع، المهام، المتطلبات، إصدارات الكراسة، القوالب، المخاطر، المشكلات، البنود الزمنية، الاجتماعات والمحاضر.
- عدم تطابق الإصدار يعيد غالباً 422 بحقل `lock_version`، بينما المشروع والقالب يستخدمان 409 في مواضع محددة.
- استيراد المهام يقارن snapshot للإصدارات ثم يعيد قفل السجلات قبل commit؛ لا نجاح جزئياً.
- مولد رقم قالب الفاتورة يتطلب أن يكون داخل transaction ويقفل `document_sequences`.
- النسخ الكامل وتنظيف الملفات يتسلسلان بـ`FileInventoryLock`؛ الاستعادة تستخدم maintenance mode وقفل ملف حصرياً.

6. الزمن والتقويم

القاعدة	التنفيذ
تخزين اللحظات	Form Requests تحول مدخلات business timezone إلى UTC؛ Eloquent يعيد datetime
التوقيت التجاري الافتراضي	`Africa/Tripoli` من `BUSINESS_TIMEZONE`
تواريخ بلا وقت	start/end للمشروع وissue/due للقالب من نوع `date`
أسبوع الجدول الحالي	الأحد (`week_starts_on=0`) حتى السبت
عطلة العرض الحالية	الجمعة والسبت (`[5,6]`)
إعدادات المدير	`calendar.week_start/weekend_days` مخزنة وتستخدم في جدولة النسخ الأسبوعية؛ builder الأسبوعي الحالي يعتمد config الثابت، لذا التكامل مع العرض **جزئي**

`assigned_at` قرار الإسناد، ولا يغير `start_at/due_at`. `completed_at` مشتق من semantic حالة المهمة: يثبت عند `done` ويزال عند الخروج منه.

7. نمط الواجهة والحالة

- `resources/js/app.tsx` يحل الصفحات تلقائياً ويطبق layouts حسب الاسم.
- Inertia shared props: المستخدم، `canCreateTask`، قدرات مركز البيانات/الإعدادات، التنبيهات وحالة sidebar.
- الصفحات تحفظ الحالة المحلية بـReact؛ إرسال النماذج عبر Inertia `useForm`، وJSON عبر `fetch` مع CSRF/session.
- Wayfinder يولد مسارات TypeScript؛ ملفات التوليد مستثناة من ESLint.
- `use-unsaved-changes` يحرس Inertia navigation وhistory وreload وإغلاق الحوارات.
- شاشات المشروع الكبيرة تحمل تبويباً واحداً فقط وتجزئه (عادة 50 صفاً) لتقليل payload.

8. البحث والقراءات

البحث يستخدم SQLite `LIKE` بعد escape لـ`%` و`_`، ويطلب حرفين على الأقل، ويحد كل نوع بخمس نتائج. الأنواع: project/task/client/requirement/invoice template/document/team. كل query يطبق visibility scope قبل البحث.

`ProjectMetrics` هو المصدر المشترك لتقدم المشروع وصحته والمرحلة التالية. `WeeklyScheduleBuilder` يبني أشرطة المهام والاجتماعات ويقصها عند حدود الأسبوع. لا تدخل إجماليات قوالب الفواتير في مؤشرات dashboard.

9. Adapter boundaries

- `MalwareScanner`: `none | command | callback`. في production يجب أن يكون configured وإلا يفشل الرفع مغلقاً.
- Laravel `Storage`: الملفات الخاصة تحت `storage/app/private` افتراضياً.
- PDF: mPDF متزامن إلى memory ثم response private/no-store.
- XLSX: PhpSpreadsheet مع حدود ZIP/حجم/ورقة/قالب؛ CSV stream وparser ثابت.
- Mail افتراضياً `log` محلياً؛ يجب تهيئة mailer إنتاجي للتحقق وإعادة كلمة المرور.

10. قرارات وحدود معلنة

- لا microservices ولا multi-tenancy ولا public API في v1.
- لا WebSocket/Echo realtime؛ الصفحات تعيد تحميل البيانات بعد الطفرات، والتنبيهات يزامنها scheduler.
- queue مهيأة database لكن الكود الحالي لا يعرّف `app/Jobs`; العمليات الثقيلة ما زالت متزامنة.
- PostgreSQL/MySQL connections عامة من Laravel وليست ملف نشر معتمداً؛ backup adapter الحالي SQLite-only.
- legacy commercial models والجداول موجودة للحفاظ على البيانات التاريخية فقط، ويمنع route binding الوصول لها.

DATA MODEL

> المصدر: migrations ونماذج Eloquent وعلاقاتها كما هي في 12 أغسطس 2026.

1. اصطلاحات

- المفتاح الأساسي الافتراضي `id` integer، مع timestamps ما لم يذكر خلاف ذلك.
- `*` في الجداول التالية يعني قيداً مهماً، وليس nullable/fillable notation.
- معظم كيانات الأعمال لا تستخدم `SoftDeletes`; الأرشفة حقل `archived_at` صريح.
- `lock_version` يبدأ 1 ويزيد عند كل طفرة محمية.

2. مخطط العلاقات العالي

العلاقات النصية الكاملة:

- العميل يملك جهات اتصال ومشاريع، ويمكن اختياره اختيارياً في قوالب الفواتير.
- المشروع يملك أعضاء ومهام ومتطلبات وخطاً زمنياً ومخاطر ومشكلات وكراسة متطلبات.
- المهمة والمتطلب علاقة many-to-many؛ المهمة لها سجل إسناد.
- الاجتماع يملك timeline entry واحداً، حضوراً ومحضراً واحداً اختيارياً.
- الملف له روابط attachment متعددة، وقد يكون ملف كراسة أو محضر أو مهمة بيانات.
- قالب الفاتورة له بنود، ومنشئ، وسياق عميل/مشروع اختياري.

> **مخطط Mermaid:** المصدر النصي محفوظ أدناه، ويُقرأ مع الشرح النصي أو الجدولي المحيط به.

```
erDiagram
    CLIENTS ||--o{ CONTACTS : has
    CLIENTS o|--o{ PROJECTS : context
    USERS ||--o{ PROJECT_MEMBERS : joins
    PROJECTS ||--o{ PROJECT_MEMBERS : has
    PROJECTS ||--o{ TASKS : has
    PROJECTS ||--o{ REQUIREMENTS : has
    TASKS }o--o{ REQUIREMENTS : links
    PROJECTS ||--o{ TIMELINE_ENTRIES : plans
    TIMELINE_ENTRIES ||--o| MEETINGS : specializes
    MEETINGS ||--o| MEETING_MINUTES : records
    PROJECTS ||--o{ RISKS : has
    PROJECTS ||--o{ ISSUES : has
    PROJECTS ||--o| REQUIREMENT_BOOKS : owns
    REQUIREMENT_BOOKS ||--o{ REQUIREMENT_BOOK_VERSIONS : versions
    FILE_OBJECTS ||--o{ ATTACHMENT_LINKS : linked
    SALES_DOCUMENTS ||--o{ SALES_LINE_ITEMS : contains
    USERS ||--o{ SALES_DOCUMENTS : creates
```

الرسم لا يعرض كل FK؛ الجداول التالية هي المرجع التفصيلي.

3. الهوية والبنية الأساسية

`users`

الحقل	النوع/القيد	المعنى
name, email	string؛ email unique	هوية المستخدم
email_verified_at	datetime nullable	بوابة `verified`
password	hash مخفي	كلمة مرور Fortify
phone, job_title	nullable	ملف وظيفي
global_role	string default member	admin/project_manager/member/viewer
status	string default active	active/inactive عملياً
archived_at	timestamp nullable	إبطال تاريخي للحساب
notification_preferences	JSON nullable	خيارات المستخدم
*_two_factor	text/datetime nullable	أسرار وحالة 2FA، مخفية عن serialization

علاقات: projects عبر `project_members`، passkeys، notifications، assignedTasks.

الجداول القياسية المصاحبة: `password_reset_tokens`, `sessions`, `passkeys`, `cache`, `cache_locks`, `jobs`, `job_batches`, `failed_jobs`, `migrations`.

`workflow_statuses`

`entity_type + code` فريد. الحقول `label`, `semantic(open/in_progress/done/cancelled)`, `color`, `position`, `is_active`. ترتبط بها projects/tasks/requirements. لا يوجد delete endpoint.

`system_settings`

`group + key` فريد؛ `value` JSON حتى للقيم null؛ `is_secret` موجود في النموذج لكن مجموعات الإعداد الحالية تكتب قيماً غير سرية فقط. المجموعات المعروضة: general/company/notifications/automatic_backup/calendar.

4. الدليل والمشاريع

`clients`

`code` فريد؛ الاسم مطلوب؛ email/phone/address اختيارية؛ `status`; `archived_at`; `created_by` nullable. العلاقات: contacts, projects, historical salesDocuments. نطاق الرؤية يسمح للمدير بمن أنشأه أو بمن يرتبط بمشروع مرئي.

`contacts`

FK `client_id` cascade؛ name مطلوب؛ role/email/phone اختيارية؛ `is_primary`, `is_active`. تحقق التطبيق يمنع جهة أساسية غير نشطة، ويتأكد أن primary contact المختار للمشروع تابع للعميل.

`projects`

المجموعة	الحقول/القيود
الهوية	`code` nullable, `name`، unique
الدليل	manager nullable، primary_contact nullable، client nullable nullOnDelete
التخطيط	start/end date nullable، priority، status restrict مع end ≥ start في validation
التزامن/التاريخ	`lock_version`, `archived_at`

`project_members`

`project_id + user_id` فريد؛ `project_role` default member؛ `status` default active. حذف المشروع cascade؛ حذف المستخدم restrict. مدير المشروع والمستخدم المنشئ يضافان كمديرين نشطين عند الإنشاء.

5. المتطلبات والعمل

`requirements`

FK مشروع؛ `code` فريد داخل المشروع؛ title/description/acceptance_criteria؛ priority؛ status restrict؛ owner nullable؛ lock_version؛ archived_at. الكود الفارغ يولد `REQ-00001` بعد الإدراج.

`tasks`

FK مشروع؛ code فريد داخله؛ title/description؛ status؛ priority؛ assignee nullable؛ `assigned_at` مستقل؛ `start_at` و`due_at` مطلوبان؛ completed_at؛ estimated_minutes؛ notes؛ lock_version؛ archived_at. الخدمة تولد `TSK-00001`، وتضمن due ≥ start وتزيل assigned_at عند إزالة المسؤول.

`requirement_task`

مفتاح مركب requirement_id/task_id؛ كلاهما cascade. التطبيق لا يقبل الربط خارج المشروع نفسه أو إلى متطلب مؤرشف.

`task_assignment_events`

task cascade؛ from/to user nullable؛ recorded_by restrict؛ assigned_at؛ recorded_at؛ note. يسجل حدثاً فقط عند تغير المسؤول، وليس كل حفظ.

6. التخطيط والاجتماعات

`timeline_entries`

project؛ `kind`; title؛ starts_at nullable؛ ends_at؛ status؛ owner؛ note؛ metadata JSON؛ archived_at؛ lock_version.
أنواع الإدخال العام: milestone/delivery/review/deadline/phase/event؛ `meeting` محجوز لخدمة الاجتماعات.

`meetings` و`meeting_attendees`

meeting يملك `timeline_entry_id` archived_at، agenda، URL، location، organizer nullable، unique cascade
وattendees. lock_version يربط meeting/user بصورة فريدة مع status:
invited/accepted/declined/tentative/attended/absent.

`meeting_minutes`

`meeting_id` unique؛ summary مطلوب؛ decisions/action_items؛ file_object_id nullable؛ recorded_by؛ recorded_at؛
lock_version. لا يوجد سجل محضر ثان للاجتماع؛ المسار upsert.

7. الحوكمة

`risks`

project؛ title/description؛ probability وimpact (1–5 في validation)؛ status
open/monitoring/mitigated/accepted/closed؛ owner؛ mitigation؛ due_at؛ archived_at؛ lock_version. درجة العرض =
impact × probability.

`issues`

project؛ title/description؛ severity low/medium/high/critical؛ status open/in_progress/resolved/closed؛ owner؛
due_at؛ resolution؛ archived_at؛ lock_version. يتطلب resolved/closed وجود resolution.

8. الملفات وكراسة المتطلبات

`file_objects`

الحقل	المعنى
disk, storage_key unique	مكان خاص يولده الخادم
original_name, mime_type, extension	metadata منقحة
size_bytes, checksum_sha256	سلامة وحصص
scan_status	pending/safe/structurally_safe/quarantined حسب المسار

الحقل	المعنى
uploaded_by, uploaded_at	أصل الملف

`attachment_links`

يربط file_object بمشروع، ويمكن أن يحدد **واحداً** من task/requirement/requirement_book_version/meeting_minutes وفق مسار الخدمة. `archived_at` يؤرشف الرابط فقط. بعض حصرية الأهداف تفرضها الخدمة وليست CHECK constraint في migration.

`requirement_books` و`requirement_book_versions`

Book واحد unique لكل مشروع. Version فريد بـ(book, version_number)، وله title/status/file/uploader/uploaded_at/is_current/lock_version/archived_at. الحالات: draft/under_review/approved/superseded. الخدمة تضمن إصداراً حالياً نشطاً واحداً منطقياً، لكن قاعدة البيانات لا تملك partial unique index لـ`is_current`.

9. قوالب الفواتير والبيانات التاريخية

`sales_documents`

الجدول اسمه تاريخياً تجاري، لكن سطح المنتج الحالي يقبل فقط:

- `type=invoice`؛
- `status=draft|archived`؛
- number unique مولد من `document_sequences`؛
- title؛ client/project/issue_date/due_date اختيارية؛
- reference؛ currency LYD/USD/EUR؛ discount/tax؛ notes؛ snapshots؛ lock_version؛ created_by.

`SalesDocument::resolveRouteBindingQuery` و`scopes` يحجبان proposal/receipt/letterhead والحالات المحاسبية القديمة. لا تحذف بيانات legacy.

`sales_line_items`

document cascade؛ name/description؛ quantity decimal(12,3) >0؛ unit؛ unit_price decimal(14,2) ≥0؛ position.
الإجماليات لا تخزن؛ يعيد `SalesCalculator` subtotal/discount/tax_base/tax/total باستخدام BCMath.

جداول legacy

`proposal_details`, `receipt_details`, `letter_details` و`source_document_id` ما زالت موجودة للتوافق التاريخي، لكنها غير مستخدمة في مسارات المنتج الحالية. `document_sequences` يحتفظ بـdocument_type/year/next_number unique.

10. العمليات والتدقيق

`data_jobs` و`import_errors`

Job: type، resource_type، format، status، file_object، created_by، summary JSON، error_message، started/completed. Error: job، sheet/row/field/code/message. حالات الاستيراد تتدرج من processing إلى validated/succeeded/failed حسب المسار.

`activity_logs`

actor nullable، project nullable، action، subject_type/id، before/after JSON، request_id، correlation_id، IP، user_agent، created_at فقط. لا يعرض التطبيق endpoint حذف أو تعديل؛ هو append-only على مستوى التطبيق.

`notifications`

Laravel database notifications: UUID، type، notifiable morph، data، read_at. Project Desk يستخدم type ثابت `project-desk.deadline` ومعرفات مستقرة مشتقة من user/source.

11. سياسات الحذف وFK

- حذف project يسبب cascade لموارد المشروع، لكنه ليس endpoint عادياً؛ المنتج يؤرشف.
- المستخدم المشار إليه في عضوية/أحداث/ملفات غالباً restrict، بينما manager/owner nullable يستخدم nullOnDelete.
- client المرتبط بمشروع nullOnDelete، لكن sales historical FK restrict في migration.
- حذف file_object cascade لـattachment_links؛ مراجع الكراسة/المحضر/data jobs تستخدم nullOnDelete أو restrict حسب الجدول.
- تنظيف اليتيم لا يعتبر رابطاً مؤرشفاً يتيماً؛ كل durable reference يحمي الملف.

12. فهارس مهمة

توجد فهارس على نطاقات الحالة/الأرشفة، تواريخ المهام، المسؤول والحالة، المشروع/التاريخ للخط الزمني، المخاطر والمشكلات، job، scan status، activity subject/actor/project/correlation، type/status، ومخزون المرفقات. اختبار الحجم يغطي 10,000 مهمة واستقرار عدد queries لمقاييس الحافظة.

API AND ROUTES

> لا يوجد Public REST API في الإصدار الحالي. هذه عقود Web/Inertia وJSON داخلية محمية بالجلسة وCSRF؛ أي عميل خارجي يحتاج عقداً وإصداراً ومصادقة منفصلة قبل الاعتماد.

1. العقد العام

1.1 Middleware والاستجابة

جميع مسارات الأعمال تقريباً تستخدم `web + auth + verified + active`. الاستثناءات: صفحة البداية؛ مسارات Fortify للضيف؛ profile/notification preferences تحتاج auth+active ولا تشترط verified؛ health `/up`.

نوع الطلب	نجاح	فشل شائع
Inertia page	HTML أولي أو Inertia JSON props	404، 403، redirect/login
Inertia mutation	302/303 إلى صفحة مع flash toast	422 403، validation bag
JSON مع `Accept: application/json`	`{data: ...}` أو payload موثق؛ create غالباً 201 + Location	`{message, errors}` عند 422؛ 401/403/404؛ stale 409؛ restore 423
تنزيل	PDF/XLSX/CSV/private file response	403 أو 404 دون كشف storage key

كل response ويب يضم `X-Request-Id` و`X-Correlation-Id`. الطلب يقبل معرفاً يطابق `[A-Za-z0-9][A-Za-z0-9._:-]{0,99}` وإلا يولد UUID.

1.2 رموز الأخطاء

HTTP	الدلالة في النظام
401	لا يوجد مستخدم مصادق في نقطة تتوقعه
403	Policy/Gate أو حساب معطل؛ تنزيل scan غير safe
404	سجل خارج النطاق/parent mismatch/نوع legacy محجوب/مجموعة غير مدعومة
409	lock_version قديم في مشروع أو قالب فاتورة في مواضع محددة
422	Form Request، lock_version قديم في معظم الموارد، ملف/استيراد/حالة مجال غير صالحة
423	الاستعادة تمسك قفل الكتابة ولا تقبل طلبات متزامنة
429	login/2FA/passkey/upload/restore rate limit
503	maintenance mode أثناء restore

لا تعتمد على نص الخطأ وحده؛ تعامل مع status و`errors.<field>`. تعاد الصفحة أو المعاينة عند conflict، ثم يرسل `lock_version` الجديد.

2. مسارات الواجهة العليا

Method	URI	الاسم	النتيجة
GET	/	home	welcome
GET	/dashboard	dashboard	Dashboard Inertia

Method	URI	الاسم	النتيجة
GET	`/search?q=`	search	JSON نتائج مرئية؛ 2–80 حرفاً، وإلا فارغ
POST	`/notifications/{uuid}/open`	notifications.open	mark read ثم redirect لوجهة يعاد التحقق منها
GET	`/up`	—	health baseline

3. المشاريع والعملاء والفريق

3.1 المشاريع

Method	URI	الغرض
GET/POST	`/projects`	list/create
GET/PUT	`/projects/{project}`	show/update
POST	`/projects/{project}/archive`	archive
POST	`/projects/{project}/restore`	restore
GET	`/projects/{project}/summary.pdf`	PDF ملخص مصرح ومدقق

فلاتر list: `h=danger|attention|healthy,scope=active|archived,sort=end_date|start_date|name|priority|created_at,direction=asc|desc`. pagination ثابت 20.

عقد create/update الأساسي:

```
{
  "code": "PRJ-042",
  "name": "اسم المشروع",
  "description": null,
  "client_id": 3,
  "primary_contact_id": 8,
  "manager_id": 5,
  "status_id": 2,
  "priority": "high",
  "start_date": "2026-08-12",
  "end_date": "2026-09-30",
  "members": [{"id": 5, "role": "manager"}, {"id": 7, "role": "member"}],
  "lock_version": 4
}
```

`lock_version` update فقط. code unique max40؛ end ≥ start؛ status مشروع فعال؛ priority low/medium/high/critical؛ client/manager/member نشطون؛ contact تابع للعميل؛ members بلا تكرار وبأدوار manager/member/viewer. لا يسمح نقل موارد المشروع عبر nested routes.

صفحة show تقبل `tab=overview|requirements|tasks|timeline|meetings|risks|issues|team|documents|client|activity`, و`archived=1`, و`activity_page`/`tab_page`. يحمل الخادم التبويب النشط فقط.

3.2 العملاء وجهات الاتصال

Method	URI	الغرض
GET	`/clients`, `/clients/create`, `/clients/{id}`, `/clients/{id}/edit`	صفحات
POST/PUT	`/clients`, `/clients/{id}`	create/update
POST	`/clients/{id}/archive` أو `/clients/{id}/restore`	تاريخ دون حذف
POST/PUT	`/clients/{id}/contacts[/{contact}]`	create/update contact
POST	`/clients/{id}/contacts/{contact}/archive` أو `.../restore`	تعطيل/إعادة تفعيل

فلاتر: Client payload: `code` unique max40، name، nullable email/phone/address(max5000)، status active/inactive q، status active/inactive/archived، archived active/only/all، per_page 1–100. Contact: name، role، email، phone، primary; `is_primary/is_active` booleans يجب أن يكون active.

3.3 الفريق

Method	URI	الغرض
GET/POST	`/team`	list/create user
PUT	`/team/{member}`	update
POST	`/team/{member}/archive` أو `/team/{member}/restore`	archive/restore

Payload: name/email unique/phone/job_title/global_role/status؛ عند الإنشاء password + confirmation. عند التعديل password اختياري. تعديل المستخدم لنفسه في email/role/password يتطلب current_password وتأكيداً حديثاً؛ Admin فقط يدير الحسابات ولا يستطيع أرشفة نفسه.

4. المهام والمتطلبات

4.1 المهام

Method	URI	الغرض
GET	`/tasks`, `/tasks/create`, `/tasks/{task}/edit`	list/form
POST/PUT	`/tasks`, `/tasks/{task}`	create/update كامل
PATCH	`/tasks/{task}/status`	تغيير الحالة فقط
POST	`/tasks/{task}/archive` أو `/tasks/{task}/restore`	archive/restore

Payload الكامل:

```
{
  "project_id": 1,
  "title": "تنفيذ الواجهة",
  "description": null,
  "status_id": 6,
  "priority": "high",
  "assignee_id": 7,
  "assigned_at": "2026-08-12T09:00",
  "start_at": "2026-08-13T08:00",
  "due_at": "2026-08-18T17:00",
  "estimated_minutes": 1200,
  "notes": null,
  "requirement_ids": [4, 5],
  "assignment_note": "بدء المرحلة",
  "lock_version": 3
}
```

start/due مطلوبان وdue ≥ start؛ assignee اختياري لكن يجب أن يكون مدير المشروع أو عضواً نشطاً بدور manager/member؛ المتطلبات من المشروع نفسه؛ لا نقل مهمة بعد إنشائها؛ 1–100000 estimated. طلب status هو `{status_id, lock_version}` ويمكن للمسند إليه تنفيذه وإن لم يملك التعديل الكامل.

4.2 المتطلبات

النمط لكل nested resource هو GET collection، POST collection، GET item، PUT item، POST item/archive، POST item/restore تحت `/projects/{project}/requirements`.

Payload: `code` اختياري max40 وفريد داخل المشروع، title، description max20000، acceptance_criteria max50000، priority، status_id لحالة requirement فعالة، owner_id عضو نشط، وlock_version للتعديل/الأرشفة/الاستعادة. الخادم يولد code عند غيابه.

4.3 كراسة المتطلبات

Method	URI	Payload مختصر
GET	`/projects/{p}/requirement-book`	`archived` اختياري
POST	`/projects/{p}/requirement-book/versions`	multipart: title، version_number؟، status؟، note؟، is_current؟، file
PUT	`.../versions/{v}`	lock_version + title/status/note/is_current fields
POST	`.../{v}/make-current` أو `.../archive` أو `.../restore`	`{lock_version}`

File max من `UPLOAD_MAX_FILE_KB`; النوع يخضع لخدمة الملفات. status من draft/under_review/approved/superseded. لا يمكن إلغاء current مباشرة؛ عيّن غيره. لا يمكن أرشفة آخر إصدار حالي دون انتقال صالح.

5. التخطيط والحوكمة

5.1 خط الزمن

Nested تحت `/projects/{p}/timeline-entries`: GET/POST collection، GET/PUT item، POST archive/restore. Query: per_page≤100، to، from، kind، archived boolean.

Payload: kind milestone/delivery/review/deadline/phase/event؛ title؛ starts_at؛ ends_at nullable ≥ start؛ status planned/in_progress/completed/cancelled؛ owner active member؛ note؛ metadata object؛ lock_version للتعديل. `kind=meeting` ممنوع هنا.

5.2 الاجتماعات والمحاضر

Nested تحت `/projects/{p}/meetings`: POST archive/restore، GET/PUT item، GET/POST collection، وPUT `/{meeting}/minutes`.

Meeting payload:

```
{
  "title": "اجتماع الاعتماد",
  "starts_at": "2026-08-15T10:00",
  "ends_at": "2026-08-15T11:00",
  "status": "planned",
  "organizer_id": 5,
  "location": "قاعة 1",
  "meeting_url": "https://meet.example/room",
  "agenda": "...",
  "note": null,
  "attendees": [{"user_id": 5, "attendance_status": "accepted"}],
  "lock_version": 2
}
```

organizer/attendees، URL http/https، ends > starts. أعضاء نشطون. Minutes payload: summary مطلوب max100000؛ decisions/action_items؛ إما `file_object_id` آمن أو multipart `attachment` لا كلاهما؛ lock_version عند وجود محضر سابق.

5.3 المخاطر والمشكلات

كل منهما يوفر POST archive/restore، GET/PUT item، GET/POST collection تحت المشروع. List يقبل status، archived، per_page≤100.

- Risk: title، description، probability/impact 1–5، status open/monitoring/mitigated/accepted/closed، owner عضو، lock_version، due_at، mitigation.
- Issue: title، description، severity low/medium/high/critical، status open/in_progress/resolved/closed، owner، resolution، lock_version، due_at مطلوب للحالتين resolved/closed.

6. الوثائق

Method	URI	العقد
GET	`/projects/{p}/files`	JSON list؛ `archived`, `per_page≤100`
GET	`/projects/{p}/file-targets`	JSON targets؛ query type/q/per_page
POST	`/projects/{p}/files`	multipart file + target_type project/task/requirement + target_id حسب النوع
POST	`/projects/{p}/files/{f}/archive` أو `.../restore`	رابط project العام

Method	URI	العقد
POST	`/projects/{p}/files/{f}/links/{link}/archive` أو `.../restore`	رابط هدف بعينه
GET	`/files/{fileObject}/download`	تنزيل خاص؛ safe + رابط نشط + authorization

الأنواع المسموحة PDF/DOCX/XLSX/CSV/JPEG/PNG/WebP؛ الحجم الافتراضي 25MB؛ extension/MIME/signature/ZIP active content تفحص. response تنزيل `private, no-store`, `nosniff`, وCSP sandbox.

7. قوالب الفواتير

Method	URI	العقد
GET/POST	`/sales`	list/create
GET/PUT	`/sales/{template}`	JSON detail/update
POST	`/sales/{template}/archive` أو `.../restore` أو `.../duplicate`	`{lock_version}`
GET	`/sales/{template}/pdf`	PDF خاص ومدقق

status draft/archived، project، q: List filters؛ pagination 20. Legacy rows تعيد 404 حتى لو ID موجود.

Create/update payload:

```
{
  "type": "invoice",
  "title": "قالب خدمات التطوير",
  "status": "draft",
  "client_id": null,
  "project_id": null,
  "issue_date": null,
  "due_date": null,
  "reference": null,
  "currency": "LYD",
  "discount_rate": "0",
  "tax_rate": "15",
  "notes": null,
  "line_items": [
    {"name": "تطوير", "description": null, "quantity": "1.000", "unit": "خدمة", "unit_price": "1000.00"}
  ],
  "lock_version": 1
}
```

`number`, `source_document_id`, `proposal`, `receipt`, `letter` prohibited. line_items min1؛ quantity >0 حتى 3 decimals؛ price ≥0؛ rates≤100؛ due لا تسبق issue؛ إذا وجد client وproject فيجب أن يتبعه. `lock_version` prohibited في create ومطلوب في update.

8. مركز البيانات

Admin فقط.

Method	URI	ملاحظات
GET	`/data-center`	صفحة Inertia
GET	`/data-center/jobs[/{id}]`	JSON pagination/detail
GET	`/data-center/csv/{resource}/template` و`/.../export`؛ ونظيرهما تحت `/xlsx`	resource clients/projects/tasks للتصدير/القوالب
POST	`/data-center/csv/{resource}/preview`	clients/tasks فقط؛ multipart CSV ≤10MB default
POST	`/data-center/xlsx/{resource}/preview`	clients/tasks فقط؛ template v1، ورقة واحدة
POST	`/data-center/imports/{job}/commit`	`{checksum_sha256}` لنفس ملف المعاينة
GET	`/exports/xlsx/{resource}`	export scoped للرؤية؛ ليس صفحة Admin حصراً

CSV headers:

- clients: `code,name,email,phone,address,status`؛
- projects export: `code,name,description,client_code,manager_email,status_code,priority,start_date,end_date`؛
- tasks:

`le,code,title,description,status_code,priority,assignee_email,assigned_at,start_at,due_at,estimated_minutes,notes`

المعاينة لا تكتب سجلات العمل. تعرض `DataJob` و`import_errors`; commit يتطلب checksum، status validated، مطابق، snapshot إصدارات صالح، وينفذ transaction لجميع الصفوف.

9. النسخ والاستعادة

Method	URI	الحماية/العقد
POST	`/data-center/backups`	إنشاء `.pdesk`؛ Admin
POST	`/data-center/backups/upload`	multipart بامتداد `.pdesk` أو `.sqlite` أو `.sqlite3` أو `.db`.
POST	`/data-center/backups/{f}/validate`	تأكيد كلمة مرور حديث 900ث؛ يعيد restore_nonce
POST	`/data-center/backups/{f}/restore`	password confirmation + throttle؛ confirmation/checksum/nonce
GET	`/data-center/backups/{f}/download`	safe private download + checksum header

Restore payload:

```
{
  "confirmation": "RESTORE PROJECT DESK",
  "checksum_sha256": "64-hex",
  "restore_nonce": "64-hex.64-hex"
}
```

nonce أحادي الاستخدام، مرتبط بالمستخدم والملف والبصمة، TTL افتراضي 600 ثانية. الاستعادة تسجل خروج الجلسة بعد النجاح.

10. الإعدادات والمصادقة

- `/system-settings`: groups. GET all، PUT `/{group}` partial fields، DELETE `/{group}` reset. Admin فقط general/company/notifications/automatic_backup/calendar.
- `/settings/workflow-statuses/{project|task|requirement}`: GET وPUT/PATCH **مجموعة كاملة**؛ يجب إرسال كل status مرة واحدة؛ code/entity_type immutable؛ لا delete.
- `/settings/profile`: GET/PATCH؛ تغيير email يتطلب current password + recent confirmation ويعيد التحقق ويلغي الجلسات الأخرى.
- `/settings/security`: GET بعد password confirmation؛ `/settings/password` PUT throttled.
- `/settings/notifications`: GET/PATCH `{enabled, overdue_tasks, upcoming_tasks, meetings, lead_hours}`؛ lead لا يتجاوز سياسة النظام.
- Fortify: login/logout، forgot/reset password، email verification، confirm password، 2FA lifecycle/recovery codes وPasskeys. Self-registration غير مفعّل.

11. مثال عميل JSON داخلي

استخدم Cookie session وCSRF؛ المثال إيضاح عقد لا API token:

```
GET /search?q=بوابة HTTP/1.1
Accept: application/json
X-Request-Id: ui-search-42
X-CSRF-TOKEN: ...
Cookie: project_desk_session=...
```

النتيجة:

```
{
  "data": [{"id":"project-1","type":"project","type_label":"مشروع","title":"بوابة العملاء","subtitle":"PRJ-001","href":"/projects/1"}],
  "meta": {"query":"بوابة","total":1}
}
```

لا تخزن أو تسجل CSRF/session/password/2FA secrets في عميل أو log.

SECURITY AND PERMISSIONS

1. نموذج الثقة

النظام تطبيق داخلي أحادي الشركة. المتصفح غير موثوق؛ جميع القرارات الحساسة في الخادم عبر authentication، Form Requests، nested parent checks، scopes، Policies/Gates، ومعاملات. إخفاء زر React ليس حد أمان.

الأصول الحساسة: credentials وmaterial 2FA/passkey، بيانات المشاريع والعملاء، الملفات الخاصة، قاعدة APP_KEY، SQLite، مفتاح `.pdesk`، session/cookies، وسجل النشاط. لا تضعها في Git أو وثائق عامة أو خدمات AI غير معتمدة.

2. المصادقة والجلسات

- Laravel Fortify guard `web` وusername=email lowercase.
- يقبل login فقط حساب `status=active` و`archived_at=null` مع password صحيح.
- login و2FA خمس محاولات/دقيقة؛ passkeys عشر/دقيقة بحسب credential/session+IP.
- Email verification إلزامي لمسارات الأعمال.
- 2FA يتطلب confirmation وكلمة مرور؛ Passkeys تعتمد RP host من `APP_URL` وallowed origin نفسه.
- `SESSION_DRIVER=database`, lifetime 120 دقيقة، encryption مفعّل في `.env.example`, HttpOnly true، serialization JSON، SameSite Lax. الإنتاج يجب أن يفعّل Secure cookie وHTTPS.
- تغيير كلمة المرور/email/الدور الحساس يلغي الجلسات الأخرى ويغير remember token. الاستعادة تمسح sessions وpassword reset tokens وتدور remember tokens.
- الحساب المعطل يطرد فوراً عبر `active` middleware.

في 12 :password default، production+ مع mixed case/letters/numbers/symbols وuncompromised. في local/testing تتبع قاعدة Laravel الافتراضية الأخف.

3. مصفوفة الصلاحيات العامة

الجدول يصف المقصود العام، لكن عضوية المشروع قد تضيف قدرة محلية ولا تضيق النطاق دائماً في التنفيذ الحالي. خصوصاً، `ProjectPolicy::update` لا يمنع global `viewer` إذا كانت عضويته active بدور manager، بينما `uploadFile` يمنعه صراحة. الخلايا الموسومة «فجوة» أدناه تحتاج قراراً وتوحيداً قبل اعتبار Viewer قراءة فقط قطعياً.

القدرة	Admin	Project Manager	Member	Viewer
عرض المشاريع المرئية	كل المشاريع	managed/member	member	member
إنشاء مشروع/عميل	نعم	نعم	لا	لا
تعديل مشروع	نعم	manager_id أو project_role=manager	فقط إذا project_role=manager	ممكن إذا project_role=manager؛ فجوة
إدارة مهام المشروع	نعم	في مشروع يديره	فقط إن كان project_role=manager	ممكن عبر update project؛ فجوة
تغيير حالة مهمة مسندة	نعم	حسب المشروع/الإسناد	نعم للمهمة المسندة	لا
رفع ملف	نعم	manager أو active project member manager/member	project role member	لا
المتطلبات/الخط الزمني/الحوكمة/الاجتماعات	نعم	داخل مشروع يديره	فقط إن كان manager محلياً	قد يعدل كmanager محلي؛ فجوة
إدارة المستخدمين	نعم	لا	لا	لا
مركز البيانات والنسخ	نعم	لا	لا	لا

القدرة	Admin	Project Manager	Member	Viewer
إعدادات الشركة وحالات workflow	نعم	لا	لا	لا
عرض/إنشاء قوالب الفواتير	كل القوالب/نعم	قوالبه فقط/نعم	لا	لا
تعديل قالب فاتورة	draft يملكه أي منشئ/كلها كAdmin	draft أنشأه فقط	لا	لا

صلاحية المشروع

الرؤية: Admin، `manager_id`، أو عضوية project active. التعديل: المشروع غير مؤرشف وAdmin أو manager_id أو عضوية active بدور manager. لا يفحص فرع العضوية الدور العام، وهذا مصدر فجوة Viewer المذكورة. الاستعادة لغير Admin تشترط عضوية active مناسبة حتى لو ظل manager_id.

العملاء

Admin يرى الكل. غيره يرى من أنشأه أو عميل مشروع مرئي. Project Manager يدير من أنشأه أو عميل مشروع يديره. Contact يرث Policy العميل.

قوالب الفواتير

صلاحيتها مستقلة عن المشروع: Project Manager لا يرى إلا `created_by=self`. اختيار مشروع مرئي كسياق لا يمنح الآخرين وصولاً. `member/viewer` ممنوعان. binding يقبل invoice draft/archived فقط.

4. Object-level authorization

- كل nested route يفحص أن child يخص project الموجود في المسار؛ المخالفة 404 لتقليل تسريب الوجود.
- queries تستخدم `Project::visibleTo`, `Client::visibleTo/manageableBy`, `SalesDocument::visibleTo`.
- البحث، exports، dashboard، التنبيهات والملفات تبدأ من project IDs المرئية.
- تنزيل الملف يتطلب scan=safe، رابطاً نشطاً، ورؤية مشروع واحد على الأقل.
- SystemSetting، DataJob، وWorkflowStatus Admin-only.
- Policies ترفض تلقائياً المستخدم inactive/archived عبر `before` في معظم الموارد.

5. حماية الإدخال والملفات

Form/data

- CSRF من Laravel web middleware؛ mass assignment عبر fillable/validated fields.
- فلاتر sort/enum allowlist؛ search العام يهرب wildcards `%` و`_`.
- URLs للاجتماع/الموقع http/https فقط بحسب الحقل.
- optimistic lock يمنع overwrite الصامت.

- CSV/XLSX يرفض formula injection (`=,+,-,@,tabs`) مع استثناء phone موجب صالح؛ export يحول القيم الخطرة إلى نص آمن.
- XLSX يرفض external links، macros، أكثر من ورقة، نسخة template خاطئة، ZIP bomb بالحجم/entries.

مرفقات المشاريع

تدفق الحماية:

1. Policy وrate limit per user/project.
2. scanner availability في production؛ غياب تكامل حقيقي يرفض الرفع.
3. allowlist extension/MIME وحجم وتوقيع/بنية؛ مفاتيح التخزين عشوائية خاصة.
4. quota للمشروع والمستخدم وعدد الملفات تحت cache lock.
5. عقد `MalwareScanner`: command يمرر absolute path كوسيط دون shell interpolation، أو callback يطبق interface.
6. clean فقط ينتقل `safe`; scanner failure; `quarantined`=infected=`structurally_safe` وغير قابل للتنزيل.
7. التنزيل يعيد authorization ويستخدم `private,no-store`, `nosniff`, `CSP sandbox`.

الوضع الافتراضي local `MALWARE_SCANNER_DRIVER=none` يفيد التطوير فقط. الإنتاج غير جاهز حتى يثبت command/callback حقيقي وفحوص clean/infected/failure.

6. أمن النسخ والاستعادة

- `.pdesk` يحتوي ZIP payload لكنه يخزن مشفراً في chunks حجمها 1MiB بـAES-256-GCM authenticated.
- header versioned مع key ID غير سري؛ المفتاح byte-32 منفصل في `BACKUP_ENCRYPTION_KEY`; المفاتيح السابقة للدوران فقط.
- manifest يسجل checksums وقائمة الملفات وinventory؛ paths allowlisted، ويمنع duplicate/traversal/زيادة expanded size/entries.
- validate قبل restore يتحقق من checksum، schema، الملفات، ووجود المدير الحالي active.
- nonce، checksum، exact phrase، requires recent password موقّع أحادي الاستعمال، وthrottle 3/10 دقائق افتراضياً.
- restore يدخل maintenance، ينتظر requests، ينشئ `pre_restore`, يعزل WAL/SHM والملفات، ويعمل rollback عند الفشل.
- legacy SQLite uploads تحول إلى encrypted `.pdesk` مع `legacy_database_only=true`; لا تعيد bytes ملفات غير موجودة.

لا تخزن مفتاح النسخة مع الحزمة. انسخ الحزم إلى off-host/immutable storage، واختبر restore شهرياً في نسخة معزولة.

7. ترويسات المتصفح وTLS

كل response يضيف:

- `X-Content-Type-Options: nosniff`؛
- `X-Frame-Options: DENY`؛
- `Referrer-Policy: strict-origin-when-cross-origin`؛
- `Permissions-Policy: camera=(), microphone=(), geolocation=(), payment=()`؛
- `Cross-Origin-Opener-Policy: same-origin`؛
- `Cross-Origin-Resource-Policy: same-origin`؛

- HSTS سنة مع subdomains عند production+HTTPS.

لا توجد Content-Security-Policy عامة للصفحات في middleware الحالي؛ يوجد CSP sandbox للتنزيلات/PDF. إعداد CSP صفحة شامل **تحسين أمني مخطط** ويحتاج اختبار Inertia/Vite/fonts قبل التفعيل.

8. التدقيق والخصوصية

`ActivityLogger` يسجل IP، request/correlation، before/after، subject، action، project، actor و user-agent. أحداث login/logout/failure و2FA و Passkey تسجل دون password/secret/recovery credential. فشل login لحساب غير معروف لا ينشئ subject اصطناعياً.

اعتبارات:

- before/after قد يحتوي بيانات عمل؛ امنح عرض activity فقط لمرئي المشروع، واضبط retention خارج التطبيق إذا لزم قانونياً.
- لا يوجد UI عام لحذف audit، وهو append-only تطبيقياً، لكن DB admin يستطيع التعديل؛ الحماية غير قابلة للعبث cryptographically غير منفذة.
- request/correlation يسمحان الربط مع reverse proxy logs؛ حافظ على نفس المعرف بعد تنقيحه.

9. قائمة hardening الإنتاج

- `APP_ENV=production`, `APP_DEBUG=false`, أسرار من secret manager.
- TLS وتحويل HTTP؛ Secure/HttpOnly/SameSite cookies؛ اختبر HSTS والنطاق.
- Mailer موثوق للتحقق/reset؛ لا تستخدم log mailer.
- ماسح malware حقيقي fail-closed وتحديث signatures ومراقبة الأعطال.
- مفتاح backup مخصص، restore drill، off-host replication.
- least-privilege OS user، صلاحيات `storage` و SQLite فقط، لا expose `storage/app/private`.
- PHP/Nginx upload limits متطابقة مع application limits.
- تدوير logs وحجب الأسرار، مراقبة 401/403/409/423/429/5xx وأحداث scanner/restore.
- scheduler و queue تحت systemd/Supervisor؛ أوقف custom workers وقت restore.
- CI، Composer/npm audit من build artifacts، lockfiles موثقة.
- نسخ SQLite على local durable disk فقط، لا NFS/SMB.

10. تهديدات وحدود متبقية

الخطر	الوضع الحالي	الإجراء
XSS/asset injection	React escaping + validation؛ لا CSP صفحة شامل	أضف CSP tested ونظف أي HTML مستقبلي
رفع ضار	عقد وتنفيذ fail-closed production	تهيئة ماسح حقيقي وإثباته قبل Go

الخطر	الوضع الحالي	الإجراء
DB/file loss	`.pdesk` كامل ومشفر	RPO/RTO مراقبة + off-host + drills
سرقة جلسة	HttpOnly/SameSite/encryption available	TLS + Secure + rotation + incident runbook
إساءة Admin	audit فقط؛ لا فصل موافقات	least privilege ومراجعة دورية؛ dual-control للاستعادة إن تطلبت المخاطر
Horizontal race	ملف single-host v1 وflock	لا توسع قبل distributed locks/backup adapter
Audit tampering by DB operator	غير محمي cryptographically	export/WORM أو signing إذا أصبح مطلب امتثال

OPERATIONS AND RECOVERY

1. الغرض والنطاق

هذا الدليل مخصص لمشغل Project Desk في الإصدار الحالي ذي المضيف الواحد وقاعدة SQLite. يشرح التشغيل الدوري، النسخ الاحتياطي، الاستعادة، معالجة الملفات، والمواقف الطارئة كما تنفذها الشيفرة. لا يحل محل سياسة استمرارية أعمال مؤسسية، ولا يجعل SQLite أو التخزين المحلي مناسبين تلقائياً لنشر أفقي متعدد العقد.

حالة القدرات:

القدرة	الحالة	الملاحظة التشغيلية
نسخ `.pdesk` كامل ومشفر	منفذ	قاعدة البيانات والملفات المشار إليها، مع manifest وchecksums
رفع نسخة والتحقق منها قبل الاستعادة	منفذ	فحص بنية وحجم ومسارات وتشفير ومخطط وصلاحيات
استعادة ذرية مع رجوع	منفذ	وضع صيانة، قفل حصري، نسخة ما قبل الاستعادة، staging للملفات
استيراد قاعدة legacy `.db/.sqlite`	منفذ جزئياً	تتحول إلى `.pdesk` موسومة database-only؛ الملفات القديمة غير الموجودة لا تستعاد
نسخ تلقائي حسب الإعدادات	منفذ	يحتاج scheduler دائم العمل
نقل النسخ إلى موقع خارجي	مخطط/مسؤولية البنية	التطبيق يخزن محلياً؛ يجب إضافة نسخ off-host خارج التطبيق
High availability متعدد المضيفين	غير مدعوم في v1	الأقفال وSQLite والتخزين محلية

2. مكونات التشغيل ومواقع البيانات

العنصر	المصدر/الموقع الافتراضي	قاعدة التشغيل
قاعدة البيانات	`database/database.sqlite` عند اختيار SQLite	ملف محلي دائم، ليس NFS/SMB
الملفات الخاصة	قرص `local` تحت `storage/app/private`	لا تعرضه خوادم الويب مباشرة

العنصر	المصدر/الموقع الافتراضي	قاعدة التشغيل
الملفات المؤقتة	`storage/app/private/tmp` وما ينشئه كل تدفق	يحذفها التطبيق بعد النجاح/الفشل؛ راقب البقايا
النسخ الاحتياطية	المسار المعرّف في إعدادات/تهيئة Project Desk	اقصر الوصول على مستخدم الخدمة
السجلات	قنوات Laravel في `storage/logs` افتراضياً	دوران واحتفاظ وتنبيه خارجي
cache/session/queue	قاعدة البيانات افتراضياً في `.env.example`	تعتمد على جداول Laravel ومتانة SQLite نفسها
أصول الواجهة	`public/build` بعد `pnpm build`	جزء من artifact الإصدار

يجب أن يملك مستخدم PHP-FPM/الخدمة حق القراءة والكتابة على ملف SQLite ومجلده، و`storage` و`bootstrap/cache` فقط. لا تمنحه حق الكتابة على الشيفرة المنشورة.

3. العمليات المجدولة والأوامر

شغّل `php artisan schedule:run` كل دقيقة عبر cron أو شغّل `php artisan schedule:work` تحت مدير خدمات. الجدول الفعلي:

المهمة	التوقيت	منع التداخل	الغرض
`project-desk:automatic-backup`	كل دقيقة	60 دقيقة	يفحص موعد النسخ المضبوط ثم ينشئ النسخة المستحقة مرة واحدة
`project-desk:sync-notifications`	كل دقيقة	10 دقائق	ينشئ/يحدّث تنبيهات المهام والاجتماعات الدائمة
`project-desk:prune-orphaned-files`	يومياً 03:30 بتوقيت العمل	60 دقيقة	يزيل الكائنات اليتيمة بعد مهلة الاحتفاظ

الأوامر المتاحة للمشغل:

```
php artisan project-desk:automatic-backup
php artisan project-desk:sync-notifications
php artisan project-desk:prune-orphaned-files
php artisan project-desk:provision-admin
php artisan project-desk:ensure-app-key
```

`provision-admin` تفاعلي وآمن لإنشاء أول مدير؛ لا توثق أو تشحن بيانات دخول ثابتة. `ensure-app-key` لا يستبدل مفتاحاً قائماً. لا تدور `APP_KEY` أو مفتاح النسخ بلا خطة ترحيل واختبار استعادة.

لا توجد Jobs تطبيقية مخصصة حالياً؛ إعداد queue موجود للبنية والإطار، لكن أعمال المجال الحالية متزامنة أو تنفذها أوامر scheduler. إذا أضيفت Jobs لاحقاً يجب تشغيل `queue:work` ومراقبته وإيقافه أثناء الاستعادة.

4. روتين التشغيل

4.1 كل يوم

1. تحقق أن scheduler نفذ خلال آخر دقيقتين، وأن مزامنة التنبيهات لم تفشل.
2. راجع أخطاء 5xx وعمليات 401/403/409/423/429 غير المعتادة ومعرّفات الطلب/الارتباط.
3. راقب مساحة القرص: SQLite وWAL والملفات الخاصة والنسخ والسجلات.

4. تحقق من حالة ماسح البرمجيات الضارة؛ فشل الماسح في الإنتاج يمنع قبول التنزيل الآمن للملف.
5. تحقق من أحدث نسخة تلقائية وchecksum وحجمها، ثم من نجاح نقلها إلى موقع خارجي إن كان النقل منفذاً بالبنية.
6. راجع فشل البريد إن كانت تنبيهات التحقق/إعادة كلمة المرور مطلوبة.

4.2 أسبوعياً

1. نفّذ تحققاً قابلاً للقراءة على نسخة `pdesk.` مختارة من واجهة البيانات.
2. راجع الملفات في quarantine أو `structurally_safe` وأسباب فشل الفحص.
3. راجع الحسابات المعطلة/المؤرشفة، المديرين، ومديري المشاريع.
4. افحص نمو `activity_logs` و`security_events` و`notifications` وفق سياسة الاحتفاظ.
5. أكد أن الوقت والمنطقة `BUSINESS_TIMEZONE` وتوقيت المضيف/NTP صحيحان.

4.3 شهرياً أو قبل إصدار كبير

نفّذ تمرين استعادة في بيئة معزولة من نسخة إنتاج حقيقية منزوعة الوصول الخارجي. قس زمن الاكتشاف، فك الحزمة، التحقق، الاستعادة، smoke tests، وقرار العودة. سجّل RPO وRTO الفعليين؛ التطبيق لا يَعِد بقيم ثابتة.

5. دورة النسخة `pdesk.`

5.1 الإنشاء

الحزمة الكاملة تحتوي manifest، لقطة SQLite متسقة، والملفات الدائمة المشار إليها. التشفير AES-256-GCM موثق في الحزمة، ويُطبّق على chunks مصادق عليها بحجم 1 MiB. يتحقق المنشئ من checksums والحجوم والمسارات. في الإنتاج يلزم `BACKUP_ENCRYPTION_KEY` مخصص؛ يمكن تعريف مفاتيح سابقة للقراءة أثناء التدوير دون عرضها في الوثائق أو السجلات.

خط البيانات مكتوب نصياً:

```
pdesk. داخلية -> تشفير مصادق -> ملف ZIP حزمة -> checksumsو manifest -> قفل/لقطة متسقة -> جمع المراجع الدائمة -> تحقق نهائي
```

لا تعتمد على وجود الملف فقط؛ النسخة لا تعد صالحة حتى ينجح التحقق ويمكن فتحها بالمفتاح المتاح.

5.2 الرفع والتحقق

يدقق التطبيق في:

- الامتداد والبصمة والتشفير والإصدار وحدود الحزمة والملفات.
- منع absolute paths و`..` وsymlinks وتجاوز العدد/الحجم ونسب الضغط الخطرة.
- checksums للقاعدة والملفات وتوافق مخطط قاعدة البيانات.
- وجود مدير فعال حتى لا تنتج استعادة تقفل النظام.
- checksum النسخة مقارنة بما أكد المستخدم قبل التنفيذ.

التحقق لا يغير بيانات العمل. احتفظ بالملف المرفوع في نطاق خاص وطبّق مهلة احتفاظ محددة.

5.3 شروط الاستعادة

الاستعادة عملية مدير فقط، وتتطلب مجتمعة:

- تأكيد كلمة مرور حديث ضمن النافذة المضبوطة، افتراضياً 900 ثانية.
- العبارة الحرفية `RESTORE PROJECT DESK`.
- checksum مطابق للنسخة التي تم التحقق منها.
- nonce موقّع أحادي الاستخدام، افتراضياً 600 ثانية.
- اجتياز throttle الاستعادة، افتراضياً 3 محاولات في 10 دقائق.
- قاعدة SQLite ومضيف واحد وقابلية الحصول على القفل الحصري.

5.4 تسلسل الاستعادة والرجوع

الوصف الكامل للتدفق:

1. يعيد الخادم التحقق من النسخة والمدخلات، لا يثق بمعاينة قديمة.
2. يدخل وضع الصيانة ويأخذ قفل `flock` حصرياً؛ يجب إيقاف workers مخصصة إن وجدت.
3. ينشئ نسخة `pre_restore` من الحالة الحالية.
4. يفك القاعدة والملفات إلى staging ويتحقق منها، دون الكتابة المباشرة فوق الحالة الحية.
5. يعزل SQLite مع ملفات `-wal` و`-shm` ويبدّل القاعدة والملفات.
6. يشغّل اختبارات السلامة المطلوبة ويثبت الحالة الجديدة.
7. عند أي فشل يعيد قاعدة وملفات ما قبل الاستعادة ويبلغ بحالة واضحة؛ لا تحذف أدلة الفشل قبل التحقيق.
8. عند النجاح يبطل الجلسات/بيانات المصادقة ذات الصلة، يخرج من الصيانة، ثم يلزم تسجيل الدخول مجدداً.

بعد النجاح نفّذ smoke tests على تسجيل الدخول، dashboard، مشروع وملف آمن، بحث، وقالب فاتورة PDF. راجع أحدث activity/security event ومعرف العملية.

5.5 النسخ القديمة

ملفا `.sqlite` و`.db`. المقبولان legacy يمثلان قاعدة فقط. يحولهما التطبيق إلى `.pdesk`. مشفرة تحمل `legacy_database_only=true`. لا يمكن اختراع مرفقات لم تكن داخل الملف؛ لذلك:

- احتفظ بنسخة منفصلة من شجرة الملفات القديمة قبل الترحيل.
- افحص سجلات الملفات/المرفقات المفقودة بعد الاستعادة.
- لا تسوق استعادة legacy على أنها استعادة كاملة.

6. الملفات والفحص والتنظيف

حالة الملف التشغيلية تمر تقريباً بالتسلسل التالي:

```
ربط -> scanner failure أو infected أو clean -> malware فحص -> MIME/signature/quota/rate/رفع مؤقت -> تحقق امتداد
حذف بعد المهلة -> orphan دائم أو
```

الحالة	السلوك	إجراء المشغل
`clean`/آمن	قابل للربط والتنزيل المصرح	لا إجراء إلا المراقبة
`infected`	quarantine وغير قابل للتنزيل	اعزل المصدر وراجع الحادث، لا تتجاوز النتيجة يدوياً بلا سياسة
`structurally_safe` بعد فشل scanner	ليس آمناً للتنزيل	أصلح الماسح وأعد الفحص وفق مسار التطبيق
orphan داخل المهلة	محفوظ مؤقتاً	طبيعي أثناء تدفق/فشل حديث
orphan بعد المهلة	مرشح pruning	راجع dry-run/السجل عند التحقيق

المراجع الدائمة التي تمنع الحذف تشمل روابط المرفقات حتى المؤرشفة، إصدارات كراسة المتطلبات، محاضر الاجتماعات، ومخرجات/مدخلات data jobs. الأمر يستخدم قفلاً ويعزل rollback/quarantine لتجنب سباق الإنشاء أو الاستعادة.

في الإنتاج لا تستخدم scanner driver `none`. يمكن استخدام command أو callback حسب التهيئة؛ اختبر content، exit codes، timeout، contract، وفشل الاتصال على أنه fail-closed.

7. الاستيراد والتصدير كعملية تشغيلية

- Preview CSV/XLSX لا يغير البيانات؛ Commit يعيد التحقق من checksum ونسخ السجلات وينفذ معاملة all-or-nothing.
- الاستيراد الحالي للعملاء والمهام فقط. التصدير/القالب يدعم العملاء والمشاريع والمهام حسب نوع العملية.
- XLSX نسخة قالب 1 وورقة واحدة؛ ترفض الصيغ والماكرو والمراجع الخارجية وحزم ZIP الخطرة.
- إذا ظهر `409` فقد تغير الملف/المعاينة أو `lock_version` لسجل؛ أعد preview ولا تجبر الدمج.
- لا تعدل ملف job في التخزين ولا قاعدة البيانات يدوياً لتجاوز حالة مقفلة؛ احتفظ بالأدلة وشخّص السبب.

8. سيناريوهات التعافي

العارض	التشخيص الآمن	التعافي
التطبيق يعرض 503	افحص maintenance mode وlogs والقفل والاستعادة الجارية	إن لم توجد عملية حية، أكمل/ارجع runbook ثم `artisan up` بعد سلامة DB فقط
`database is locked`	افحص عمليات PHP/scheduler/backup والاستعادة ومساحة القرص	أوقف الكاتب غير الشرعي، لا تحذف WAL/SHM يدوياً أثناء عمله، أعد المحاولة بعد القفل
نسخة تلقائية لا تظهر	افحص scheduler، الإعدادات، timezone، المسار والمفتاح والمساحة	أصلح السبب وشغل الأمر مرة، ثم تحقق من النسخة
الملفات لا تنزل	افحص authorization وحالة scan ووجود object وchecksum	أصلح الماسح/استعد الملف من `.pdesk`؛ لا تجعل private disk عاماً
تنبيهات غائبة	افحص scheduler والتفضيلات والسياسة والمواعيد UTC	شغّل sync، ثم أصلح الجدولة لا إشعاراً يدوياً فقط
استيراد عالق/فاشل	افحص job/errors/checksum/record versions	صحح الملف وأعد preview؛ المعاملة تمنع أنصاف الاستيراد
login يفشل بعد restore	تأكد أن الاستعادة اكتملت والجلسات أبطلت والمدير فعال	استخدم حساباً صالحاً؛ عند فقد الجميع استخدم إجراء recovery موثق/`provision-admin` من console الموثوق

العارض	التشخيص الآمن	التعافي
مساحة القرص منخفضة	جزّئ SQLite/WAL، private files، backups، logs، tmp	أوقف عمليات كبيرة، انقل نسخاً قديمة وفق الاحتفاظ، prune عبر الأمر؛ لا تحذف كائنات حية يدوياً

9. المراقبة والتنبيه

الحد الأدنى خارج التطبيق:

- HTTP health على `/login` أو نقطة تحقق مناسبة دون تسجيل أسرار، وزمن الاستجابة ونسبة 5xx.
- نبض scheduler وآخر نجاح لكل أمر ومدة التنفيذ.
- مساحة/inodes وصحة SQLite ووقت آخر نسخة ناجحة وحجمها ونجاح off-host copy.
- فشل scanner ونسبة الملفات غير `clean`، وفشل البريد والqueue إن استعملت.
- security events الحساسة: محاولات login المتكررة، تغيير 2FA/Passkeys، استعادة، صلاحيات، وتعطيل حسابات.
- تنبيه على انجراف الساعة، انتهاء TLS، وأخطاء البناء/الإصدار.

لا تسجل كلمات المرور أو session/CSRF tokens أو مفاتيح backup أو بيانات 2FA/Passkey. استخدم request/correlation IDs للربط بين proxy وLaravel.

10. اختبار التعافي وقائمة الإغلاق

قبل اعتبار تمرين الاستعادة ناجحاً:

- [ ] checksum والحزمة والمفتاح تم التحقق منها في بيئة معزولة.
- [ ] تم توثيق وقت البداية والنهاية وRPO/RTO الفعليين.
- [ ] migration/schema والحساب الإداري والجلسات بعد الاستعادة سليمة.
- [ ] عدد/عينات المشاريع والمهام والعملاء والقوالب متسقة.
- [ ] عينات الملفات نظيفة وقابلة للتنزيل بصلاحية صحيحة.
- [ ] PDF قالب فاتورة ينتج؛ لم يضف الاختبار أي سلوك محاسبي غير موجود.
- [ ] scheduler والتنبيهات والبحث يعملون بعد الخروج من الصيانة.
- [ ] تم اختبار مسار فشل واحد على الأقل وإثبات rollback.
- [ ] النسخة `pre_restore` وأدلة التمرين خضعتا لسياسة الاحتفاظ.
- [ ] تم تدوين الملاحظات والإجراءات التصحيحية والمالك والموعد.

TESTING AND QA

1. سياسة الدليل

المصدر الحاكم هو الاختبار القابل لإعادة التنفيذ في نفس commit وبيئة CI، لا رقم نجاح قديم. جرد الشيفرة الحالي يحتوي 239 طريقة اختبار Feature و17 طريقة Unit، أي 256 طريقة مكتشفة نصياً، إضافة إلى 10 تدفقات متصفح يشغلها `run-all.mjs`. هذه أرقام جرد وليست شهادة أن آخر تشغيل نجح؛ يجب إرفاق رابط CI أو سجل مؤرخ بكل إصدار.

حالة منظومة الجودة:

الطبقة	الحالة	الأداة/النطاق
Unit	منفذ	12 PHPUnit؛ حسابات ومقاييس وجدولة أسبوعية ونسخ وفحص malware
Feature/Integration	منفذ على نطاق واسع	Laravel HTTP/DB/policies/files/backups/imports/settings
Browser workflows	منفذ	Playwright Chromium عبر 10 scripts متسلسلة
تحليل PHP الساكن	منفذ	Larastan/PHPStan level 7
أنواع TypeScript	منفذ	`tsc --noEmit` مع إعداد صارم
lint/format	منفذ	Pint، ESLint، Prettier
فحص dependencies	منفذ في CI لـ Composer	`composer audit --locked`؛ يلزم أيضاً سياسة دورية للنظام وnpm/pnpm
تغطية سطرية بحد أدنى	غير منفذ	CI يعطل coverage؛ لا توجد عتبة تمنع الدمج
اختبار حمل/أداء مستمر	جزئي	يوجد `PerformanceVolumeTest`، وليس benchmark إنتاجياً أو SLO
DAST/اختبار اختراق	غير منفذ في المستودع	مطلوب قبل نشر حساس

2. بيئة الاختبار

`phpunit.xml` يضبط بيئة معزولة تقريباً: `APP_ENV=testing`، SQLite في الذاكرة، cache/session من نوع array، queue متزامنة، mail array، وhashing منخفض الكلفة. لذلك:

- لا تستنتج أداء SQLite على القرص أو سلوك WAL/أقفال المضيف من اختبارات in-memory وحدها.
- اختبارات backup/restore التكاملية التي تنشئ ملفات مؤقتة هي الدليل الأهم لمسار القرص، ويجب إبقاؤها معزولة.
- queue sync لا يختبر worker/retry/failure transport؛ حالياً لا توجد Jobs مجال مخصصة أصلاً.
- اختبارات المتصفح تستخدم قاعدة seeded بعد `migrate:fresh --seed`، لذلك ممنوع تشغيلها على قاعدة عمل.

متطلبات CI المعلنة: PHP 8.4، Ubuntu مع `bcmath,curl,dom,fileinfo,gd,intl,mbstring,pdo_sqlite,sqlite3,xml,zip`، Node 22.12.0، pnpm 11.16.0، Chromium. يدعم Composer PHP `^8.3`، لكن build الحالي Vite 8 يحتاج toolchain Node حديثاً؛ لا تعتمد على Node 18 في التطوير أو الإصدار.

3. أوامر الجودة

من جذر `project-desk`:

```
# Backend: format check + PHPStan + PHPUnit
composer test

# build/browser كل بوابات الشيفرة باستثناء
composer run ci:check

# اختبارات Laravel فقط أثناء التشخيص
php artisan test
php artisan test --filter=SqliteBackupRestoreIntegrationTest

# Frontend
pnpm run format:check
pnpm run lint:check
pnpm run types:check
pnpm run build

# Browserاختبار قاعدة وتطلب تطبيقاً محلياً ؛ seeded
pnpm run browser:check
```

`composer test` يبدأ بـ`config:clear` ثم Pint check وPHPStan والاختبارات. لا تشغل أوامر الإصلاح مثل `pint` أو `eslint --fix` كجزء من التحقق الصامت إن أردت كشف diff غير مقصود.

تنبيه حالة الأحرف: مجلد المصدر الموجود هو `tests/browser` بينما script في `package.json` يشير حالياً إلى `tests/Browser/run-all.mjs`. يعمل ذلك عادة على Windows غير الحساس للحالة، وقد يفشل على Linux الحساس لها. هذه فجوة يجب حسمها قبل الاعتماد على CI Linux؛ التوثيق لا يعتبر browser gate ناجحاً دون تشغيل فعلي.

4. خريطة تغطية Backend

المجال	الاختبارات الممثلة	ما يجب أن تثبته
المصادقة والحساب	مجلدات `Feature/Auth` و`Feature/Settings`، `ProvisionAdminCommandTest`	login، verification، password، 2FA/passkeys، session invalidation، provisioning
المشاريع/المهام	`ProjectWorkflowTest`, `TaskWorkflowTest`, `Dashboard*`, `ProjectMetrics*`	archive/restore، CRUD، صلاحيات، status، dates، metrics، lock/version conflicts
الفريق	`TeamWorkflowTest`, `SharedNavigationAbilitiesTest`	الأدوار والحالة والعضوية وقدرات navigation
العملاء	`ClientContactCrudTest`	العميل وجهات الاتصال والتحقق والأرشفة
الحوكمة	`GovernanceResourcesTest`, `TimelineMeetingWorkflowTest`	متطلبات ومخاطر وقضايا واجتماعات ومحاضر
المستندات والملفات	`ProjectDocumentWorkflowTest`, `ProjectFileSecurityTest`, `ProjectFileTargetLinkTest`, `OrphanedFileRetentionTest`	authorization، ربط الأهداف، scanning، download، orphan pruning
قوالب الفاتورة	`SalesDocumentWorkflowTest`, `SalesDocumentAuthorizationTest`, `SalesCalculatorTest`, `PdfExportAuthorizationTest`	draft/archive/restore/duplicate، الإجماليات، PDF، عدم التسريب

المجال	الاختبارات الممثلة	ما يجب أن تثبته
Data Center	`DataCenterCsvTest`, `DataCenterXlsxTest`	preview/commit، الصيغ الخبيثة، checksum، all-or-nothing
النسخ والاستعادة	`SqliteBackupControllerTest`, `SqliteBackupRestoreIntegrationTest`, `SqliteBackupManagerTest`, `RestoreWriteFenceTest`, `AutomaticBackupCommandTest`	تشفير/manifest، صلاحيات، nonce/phrase/checksum، قفل وrollback وautomatic schedule
الأمن والتدقيق	`SecurityHeadersTest`, `SecurityActivityAuditTest`, `ActivityLogContextTest`	headers، request/correlation context، activity/security events
التنبيهات والبحث	`NotificationCenterTest`, `GlobalSearchTest`	permission-scoped results، stable notifications، preferences
إعدادات النظام/workflow	`SystemSettingsTest`, `WorkflowStatusSettingsTest`, `WeeklyScheduleBuilderTest`	validation، immutable codes، week rules والحدود المعروفة
malware	`CommandMalwareScannerTest`	command result/timeout/failure contract
حجم	`PerformanceVolumeTest`	عدم الانهيار على fixture أكبر؛ لا يمثل capacity proof

عند إضافة حالة عمل جديدة، أضف اختبار Policy/Request/Service وليس اختبار صفحة سعيد فقط. اختبر 401، 403، 404 لمنع enumeration، 409 للنسخة المتعارضة، 422 للتحقق، و423/429 حيث ينطبق.

5. تدفقات المتصفح

يشغّل `tests/browser/run-all.mjs` هذه التدفقات بالترتيب:

الملف	الهدف العام
`smoke.mjs`	login والتنقل الأساسي
`critical-journey.mjs`	رحلة مشروع/مهمة حرجة
`sales-smoke.mjs`	قوالب الفاتورة وPDF دون محاسبة
`documents-data-settings-smoke.mjs`	المستندات وData Center والإعدادات
`governance-smoke.mjs`	المتطلبات والمخاطر والقضايا/الحوكمة
`notifications-smoke.mjs`	مركز التنبيهات
`notification-preferences-smoke.mjs`	تفضيلات التنبيه وحدودها
`accessibility-responsive-smoke.mjs`	مؤشرات وصول واستجابة أساسية
`ui-task-regressions.mjs`	انحدارات واجهة المهام والجدول
`unsaved-dialogs-smoke.mjs`	حوار التغييرات غير المحفوظة والتنقل

الـrunner ينشئ حالة مصادقة مؤقتة بصلاحية ملف `0600` ويحذفها في `finally`. عند الفشل تحفظ CI مجلد `storage/app/browser-tests` وسجل الخادم 14 يوماً. يجب التأكد أن evidence لا يتضمن session/cookies أو بيانات شخصية قبل مشاركته.

اختبارات الوصول هنا smoke وليست تدقيق WCAG كامل. الفحص اليدوي المطلوب يشمل: keyboard-only، ترتيب focus، dialog focus trap/return، قارئ شاشة عربي، zoom 200%، contrast، RTL/LTR mixed content، ورسائل الأخطاء المرتبطة بالحقول.

6. خط CI الفعلي

تسلسل `.github/workflows/ci.yml`:

1. checkout بمرجع action مثبت وبدون credentials باقية.
2. إعداد PHP 8.4 وNode 22.12 وpnpm 11.16.
3. إنشاء `.env` وSQLite للاختبار فقط.
4. `composer validate` ثم install و`composer audit --locked`.
5. توليد مفتاح مؤقت وmigrate.
6. `pnpm install --frozen-lockfile` ثم format/lint/types/build.
7. `composer test`.
8. تثبيت Playwright Chromium.
9. `migrate:fresh --seed` لfixtures المتصفح، تشغيل خادم محلي ثم browser workflows.
10. رفع evidence عند الفشل.

يوجد timeout 20 دقيقة وconcurrency تلغي التشغيل الأقدم لنفس المرجع. قبل اعتماد الخط كحاجز إصدار، أصلح/اختبر اختلاف `browser`/`Browser` على Ubuntu.

7. استراتيجية بيانات الاختبار

- استخدم factories/seeders وحسابات وهمية؛ لا تنسخ production PII إلى CI.
- احتفظ بحالات boundary: بداية/نهاية اليوم UTC/Tripoli، أسبوع يبدأ الأحد، عطلة الجمعة/السبت، DST لمناطق بديلة إن سمحت الإعدادات بها لاحقاً.
- أنشئ ملفات fixtures صغيرة صالحة وخبيثة: MIME مضلل، signature خاطئ، ZIP bomb محدود، XLSX formula/macro/external checksum، path traversal، ref خاطئ.
- اختبر `lock_version` بعميلين وpreview/commit بعد تغيير سجل.
- اختبر أدواراً متقاطعة، خصوصاً global `viewer` مع project membership `manager`: بعض Policies تسمح تحديث المشروع عبر العضوية بينما upload يمنع viewer صراحة. هذا تضارب فعلي يجب تحويله إلى قرار متطلب واختبار موحد.
- اختبر archived resources وعلاقاتها الدائمة حتى لا يحذف orphan pruner ملفاً ما زال مرجعاً.

8. بوابات القبول قبل الدمج

- [ ] `composer validate --strict --no-check-publish` وaudit بلا ثغرات غير مقبولة.

- [ ] Pint check وPHPStan level 7 ناجحان.
- [ ] Prettier وESLint وTypeScript ناجحة.
- [ ] build production ناجح من lockfiles على Node المدعوم.
- [ ] كل Unit/Feature ناجحة ولا توجد skip جديدة بلا تبرير/issue.
- [ ] browser workflows ناجحة على نظام حساس لحالة الأحرف.
- [ ] migration جديدة تختبر fresh install وupgrade وbackup/restore compatibility.
- [ ] تغيير صلاحيات يملك positive وnegative وcross-project tests.
- [ ] تغيير ملف يملك MIME/signature/scanner/quota/orphan tests.
- [ ] تغيير تاريخ يختبر UTC والمنطقة وحدود الأسبوع.
- [ ] تغيير UI يراجع empty/error/loading states، keyboard، responsive، RTL وunsaved guard.
- [ ] لا توجد أسرار أو fixtures حقيقية أو browser auth state في artifact.

9. قبول الإصدار

بالإضافة إلى بوابات الدمج:

1. شغّل CI كاملاً على commit المرشح وسجّل SHA والرابط والوقت والنتائج.
2. نفذ `migrate:fresh --seed` وsmoke في staging مشابه للإنتاج.
3. اختبر upgrade من نسخة مدعومة، لا fresh install فقط.
4. أنشئ `.pdesk` في staging واستعدها في نسخة بيئة أخرى واختبر rollback متعمداً.
5. اختبر scanner الحقيقي وmailer وscheduler وdisk permissions وTLS/cookies.
6. تحقق من invoice template PDF بالعربية والعملات المدعومة، ومن عدم ظهور ledger/payment/accounting workflows.
7. راجع release notes، تغييرات البيئة، runbook والرجوع، وموافقة مالك المنتج/الأمن/التشغيل.

10. فجوات وخطة تحسين QA

الأولوية	الفجوة	الإجراء المقترح
P0	case mismatch في مسار browser script	توحيد الاسم وتشغيل CI Ubuntu قبل الإصدار
P0	لا evidence حديث مضمّن تلقائياً في الوثائق	حفظ ملخص JUnit/CI وSHA مع كل release
P1	تضارب global viewer/project manager	قرار صلاحية مركزي، تعديل Policies واختبارات matrix
P1	إعدادات calendar/timezone غير موصولة بكل الخدمات	contract tests من setting إلى weekly schedule/backup/UI
P1	لا حد coverage	جمع coverage أولاً ثم عتبة تدريجية للمجالات الحرجة
P1	backup/scanner يعتمد على قرص/command فعليين	integration suite على filesystem وscanner sandbox في CI آمنة
P2	وصول وأداء جزئيان	axe/قارئ شاشة يدوي، budgets، وload profile على staging
P2	لا DAST/SBOM gate موثق	إضافة dependency policy، SBOM، وDAST مصادق محدود

11. تشخيص فشل الاختبار

الفشل	افحص أولاً	لا تفعل
Vite/Node syntax أو engine	`node --version` و`pnpm lock`؛ استخدم Node 22.12 كما CI	لا تغير lockfile بمدير حزم آخر
PHP extension missing	`php -m` وقائمة CI، خصوصاً `intl/bcmath/sqlite/zip`	لا تتجاوز الاختبار من production image ناقصة
test يعمل منفرداً لا ضمن suite	state/leak/time/factory/order	لا تثبت ترتيب الاختبارات كحل
SQLite locked	عمليات متوازية وملفات مؤقتة وWAL	لا تحذف WAL أثناء كاتب نشط
browser لا يجد الملف على Linux	حالة `Browser` مقابل `browser`	لا تعتبر نجاح Windows دليلاً على Linux
browser selector متقلب	semantics/accessibility locator وانتظار الحالة	لا تضف sleep ثابتاً طويلاً
timezone assertion	UTC normalization و`BUSINESS_TIMEZONE` وCarbon clock	لا تعتمد على ساعة الجهاز بلا تثبيت
snapshot/fixture يحتوي سراً	evidence وstorage state وlogs	لا ترفعه قبل التنقيح

DEVELOPER GUIDE

1. نقطة البداية

Project Desk تطبيق PHP 8.3 / Laravel 13+ مع Inertia 3 وReact 19 وTypeScript وVite 8 وTailwind 4. المعمارية modular monolith: طلب الويب يمر عبر Route ثم Form Request/Policy ثم Controller وخدمة مجال ثم Eloquent/SQLite، ويعيد صفحة Inertia أو redirect/JSON. لا توجد microservices أو outbox أو domain event bus، ولا توجد `app/Jobs` مخصصة حالياً؛ `TaskService` وخدمات المجال هي التنفيذ الحقيقي، وليس أي تصميم قديم يذكر `SaveTaskAction`.

الحد الوظيفي المهم: المبيعات الحالية هي **قوالب فواتير غير محاسبية**. لا تضف ledger أو دفعات أو أرصدة أو ترحيل قيود تحت اسم صيانة عادية؛ ذلك منتج/نطاق جديد يحتاج SRS وقراراً صريحاً.

2. المتطلبات المحلية

الأداة	المتطلب
PHP	`^8.3`؛ CI يستخدم 8.4
Extensions	bcmath, curl, dom, fileinfo, gd, intl, mbstring, pdo_sqlite, sqlite3, xml, zip
Composer	v2 ومن lockfile
Node	استخدم 22.12 أو أحدث مدعوم من المشروع؛ Node 18 غير مناسب لـVite 8
pnpm	11.16.0 عبر Corepack
قاعدة التطوير	SQLite محلية، ملف واحد ومضيف واحد

تحقق قبل البدء:

```
php -v
php -m
composer --version
node --version
corepack enable
corepack prepare pnpm@11.16.0 --activate
pnpm --version
```

إذا غاب `intl` أو امتداد آخر، أصلح PHP CLI نفسه؛ نجاح PHP-FPM منفصل لا يعني أن أوامر Artisan/CI المحلية سليمة.

3. إعداد نسخة تطوير جديدة

المسار المختصر:

```
Copy-Item .env.example .env
New-Item -ItemType File -Path database/database.sqlite -Force
composer install
php artisan project-desk:ensure-app-key
php artisan migrate
php artisan db:seed --class=WorkflowStatusSeeder
php artisan project-desk:provision-admin
pnpm install --frozen-lockfile
pnpm run build
```

أو `composer setup` ينفذ التسلسل تقريباً، ويتوقف تفاعلياً عند provisioning أول مدير. لا تضع بيانات مدير افتراضية في commit أو توثيق.

للتطوير:

```
composer run dev
```

أو شغّل Laravel وVite كل واحد في terminal مستقل. شغل scheduler في terminal ثالث عند تطوير التنبيهات/النسخ:

```
php artisan schedule:work
```

لا تستخدم `migrate:fresh --seed` على قاعدة عمل؛ هو مخصص للاختبار/fixtures.

4. خريطة المستودع

المسار	المسؤولية
`app/Models`	25 نموذج Eloquent في جرد النسخة الحالية
`app/Http/Controllers`	31 Controller؛ تحويل HTTP وتنسيق الاستجابة
`app/Http/Requests`	validation وauthorization الأولي وعقود payload
`app/Policies`	17 Policy؛ صلاحيات الكائنات والمشروع
`app/Services`	31 خدمة؛ معاملات وتدفقات المجال والحساب/النسخ/الملفات

المسار	المسؤولية
`app/Console/Commands`	backup، notifications، orphan pruning، app key، admin provisioning
`app/Support`	أدوات مشتركة للسياق والتواريخ والأمان/التنسيق بحسب الاستخدام
`database/migrations`	مصدر حقيقة المخطط والقيود والفهارس
`database/seeders`	حالات workflow وfixtures التطوير/الاختبار
`routes/web.php`	routes الأساسية
`routes/settings.php`	الملف الشخصي والأمن والتنبيهات
`routes/workflow.php`	إعداد حالات workflow
`routes/console.php`	scheduler
`resources/js/pages`	صفحات Inertia
`resources/js/components`	عناصر مشتركة وواجهات المجال
`resources/js/hooks`	سلوك مشترك مثل unsaved changes
`resources/js/actions`, `routes`, `wayfinder`	مخرجات/تكامل Wayfinder لعقود routes typed
`tests/Feature`, `tests/Unit`, `tests/browser`	اختبارات الخادم والمتصفح
`config/project-desk.php`	حدود الملفات والنسخ والوقت والسياسات التطبيقية

الأعداد أعلاه snapshot تساعد على اكتشاف ملف منسي، وليست contract؛ حدثها أو احذفها عند تغير الجرد.

5. دورة الطلب

الوصف النصي الكامل:

`المتصفح -> middleware السياق/الرؤوس/التحقق/المصادقة -> route model binding -> Form Request -> Policy -> Controller -> Service داخل transaction عند الكتابة -> Model/SQLite والملف الخاص -> Activity/Security log -> Inertia redirect أو JSON`

القواعد:

1. لا تعتمد على إخفاء زر في React؛ كل عملية تملك authorization خادمياً.
2. استخدم Form Request للأنواع والحدود والرسائل، وPolicy لحق الفاعل في الكائن.
3. اجعل Controller رفيعاً؛ الحسابات والتغييرات المركبة في Service.
4. اجمع الكتابات المرتبطة في `DB::transaction` واستخدم row/operation lock حيث يوجد سباق.
5. سجّل before/after والسياق عبر آلية ActivityLogger الحالية دون أسرار.
6. عد بخطأ محدد: 403 منع، 404 إخفاء/غير موجود، 409 تعارض نسخة، 422 423، validation قفل، throttle 429.

6. إضافة ميزة مجال جديدة

اتبع هذا الترتيب العملي:

1. اكتب contract وحالة `implemented/partial/planned` والحدود، خاصة الأدوار والأرشفة والوقت.
2. أضف migration forward آمنة مع foreign keys/unique/indexes و`lock_version` إن كان السجل قابلاً للتحرير المتزامن.
3. أضف Model بعلاقات وcasts وfillable محددة؛ تجنب mass assignment لحقول الملكية/الدور غير المصرحة.
4. أضف Policy واختبارات matrix تشمل cross-project وarchived وinactive/global role + project role.
5. أضف Form Request، ثم Service transaction، ثم Controller وroutes مسماة.
6. مرر props الدنيا إلى Inertia؛ لا ترسل صفوفاً غير مصرح بها ثم تخفيها في الواجهة.
7. استخدم Wayfinder بدلاً من كتابة URLs متفرقة، وشغّل build/types لإعادة التوليد والتحقق.
8. أضف Unit للحساب الخالص، Feature للعقود/DB/Policy، ومتصفحاً للرحلة الحرجة فقط.
9. حدث docs التقنية وSRS ودليل المستخدم في نفس التغيير.

لا تنشئ status codes حرة؛ workflow statuses بيانات مضبوطة، وcode/entity type غير قابلين للتغيير عبر الإعدادات الحالية.

7. البيانات والمعاملات

- Migration هي الحقيقة؛ لا تعدل SQLite يدوياً لتطوير ميزة.
- archive منطقي مقصود في المشاريع والموارد ذات الصلة؛ راجع scopes وroute binding حتى لا تعيد legacy/archived بلا قصد.
- حقول `lock_version` تتطلب إرسال النسخة التي قرأها العميل؛ عند mismatch أرجع 409 ودع المستخدم يعيد التحميل/الدمج.
- Data Center preview/commit يعتمد checksum وrecord versions وall-or-nothing؛ لا تحول commit إلى upsert جزئي صامت.
- Activity log متزامن غالباً ضمن تدفق الطلب، ولا يوجد outbox. إذا نقل إلى queue مستقبلاً، صمم ضمان التسليم/الترتيب/الفشل صراحة.
- queue tables scaffold من Laravel وليست دليلاً على asynchronous domain processing.

عند إضافة migration:

```
php artisan make:migration add_example_to_projects_table
php artisan migrate
php artisan migrate:status
```

اختبر fresh install وupgrade من نسخة مدعومة وbackup/restore. لا تجعل rollback blind هو خطة استرجاع إنتاج؛ بعض migrations غير قابلة للعكس دون فقد.

8. الوقت والتقويم

- datetimes التجارية تدخل بتوقيت العمل وتخزن/تقارن UTC، ثم تعرض في المنطقة المحددة.
- الافتراضي `BUSINESS_TIMEZONE=Africa/Tripoli`.
- builder الأسبوعي الحالي يعتمد `config/project-desk.php`: بداية الأسبوع الأحد (`0`) وعطلة الجمعة/السبت (`[5,6]`).

- توجد إعدادات system calendar/timezone، لكنها ليست موصولة بالكامل بكل builders وUI/automatic backup. عامل ذلك كـ`partial` ولا تدّع أن تغيير الإعداد يغير كل السلوك.
- اختبر midnight، حدود الأسبوع، قيم date-only، والاجتماعات/التنبيهات عند التحويل إلى UTC.

لا تستخدم `now()` الموزع عشوائياً في منطق قابل للاختبار؛ مرر clock/ثبّت Carbon حيث يلزم.

9. الملفات

لا تكتب upload مباشرة في `public` أو تثق بالامتداد. المسار الحالي يفرض allowlist للامتداد/MIME/signature، حدود الحجم والعدد والمساحة/rate، تخزيناً خاصاً بمفتاح عشوائي، ثم malware scan. في الإنتاج فشل/غياب الماسح fail-closed.

لإضافة نوع ملف:

1. حدّث allowlists المتسقة في config/validator/signature inspector.
2. أضف fixtures صحيحة ومضللة واختبارات scanner/quota/download.
3. أثبت أن التنزيل لا يتم إلا لحالة clean وبPolicy صحيحة.
4. أضف المرجع الجديد إلى orphan retention وbackup manifest، بما فيه archived resource.
5. اختبر restore وchecksum ومسارات Unicode دون path traversal.

10. قوالب الفاتورة

العقد الحالي:

- نوع document المقبول `invoice` فقط؛ الحالة `draft` أو `archived`.
- عميل/مشروع/issue/due date اختياري، مع صف واحد على الأقل.
- العملات LYD/USD/EUR والنسب 0..100، والحساب في الخادم باستخدام BCMath.
- create/update/duplicate/archive/restore وPDF منفذة.
- جداول proposal/receipt/letterhead legacy محفوظة للتوافق، لكن routes/scopes الحالية تخفيها؛ لا تعيدها تلقائياً.

أي طلب دفع أو balance أو chart of accounts أو tax filing أو reconciliation خارج هذا المجال. سمِّ UI «قوالب فواتير» لا «المحاسبة».

11. React وInertia

- التطبيق RTL عربي، و`app.tsx` يهيئ صفحات Inertia وtheme.
- الحالة محلية عبر React وfetch/`useForm` Inertia؛ لا يوجد global state store أو WebSocket حالياً.
- shared props يجب أن تكون صغيرة ومصرحاً بها.
- hook التغييرات غير المحفوظة يحرس Inertia navigation و`popstate` و`beforeunload` ويعرض dialog؛ استخدمه في النماذج الجديدة.
- theme محفوظ في localStorage وcookie؛ راع SSR/flash عند تغييره.
- استخدم semantic HTML وfocus management، Radix primitives ورسائل field errors، واختبر keyboard وRTL والهواتف.
- فرّق loading/empty/error/permission/archived/conflict states؛ لا تعرض failure عام عندما يملك الخادم 409 أو 422 مفيداً.

12. البحث والتنبيهات

البحث الداخلي permission-scoped؛ query بين 2 و80 حرفاً، مع escaping لـLIKE wildcards وحد 5 نتائج لكل نوع. عند إضافة نوع نتيجة، طبق scope في query نفسه واختبر cross-project، ولا تجمع كل البيانات ثم ترشحها في PHP.

التنبيهات الدائمة ينتجها scheduler بمعرف ثابت لتجنب التكرار للمهام المتأخرة/القادمة والاجتماعات. تفضيلات المستخدم يمكنها تضييق سياسة النظام ولا تتجاوز نافذة lead العليا. لا تنشئ المنتج نفسه من UI وscheduler في آن واحد.

13. الأمن والأسرار

- احتفظ بـ`.env`. خارج Git، ولا تعرض `APP_KEY` أو backup keys أو credentials في exception/screenshot/log.
- لا تبدل `APP_KEY` في بيئة قائمة دون خطة للجلسات والحقول المشفرة.
- استخدم `BACKUP_ENCRYPTION_KEY` مخصصاً في الإنتاج ومفاتيح previous للقراءة أثناء rotation المنضبط.
- لا تستخدم mail `log` أو scanner `none` كإعداد إنتاج.
- Global `viewer` ليس مطبقاً بصورة متسقة مع project membership: `ProjectPolicy::update` قد يسمح لعضو pivot manager حتى لو global viewer، بينما `uploadFile` يمنعه صراحة. لا توسع هذا السلوك بالنسخ؛ احسم المتطلب ووحّد Policies واختباراتها.

14. أوامر يومية

```
# فحص routes والمخطط
php artisan route:list
php artisan migrate:status

# الجودة
composer test
pnpm run format:check
pnpm run lint:check
pnpm run types:check
pnpm run build

# تشغيل مهام المجال يدوياً للتشخيص
php artisan project-desk:sync-notifications
php artisan project-desk:automatic-backup
php artisan project-desk:prune-orphaned-files
```

مسارات الإصلاح تكتب ملفات (`pint`, `prettier --write`, `eslint --fix`)؛ راجع diff ولا تستخدمها لتغطية تغيير غير مقصود.

15. تشخيص التطوير

المشكلة	الفحص
صفحة 419	CSRF/cookie/domain/proxy وsession driver؛ لا تعطل CSRF
403 غير متوقع	project pivot status/role، user status/archive، global role وPolicy
409	`lock_version` أو checksum/record version؛ أعد القراءة ولا تجبر الكتابة

المشكلة	الفحص
route/types قديمة	شغل Vite/Wayfinder build وتحقق من imports وحالة الأحرف
build يفشل	Node 22.12+ و pnpm 11.16 وfrozen lockfile
Artisan يفشل	PHP CLI extensions و`.env` وSQLite write permissions
ملف غير قابل للتنزيل	scan status وPolicy وobject/checksum؛ لا تجعل القرص عاماً
إشعار لا يظهر	scheduler/timezone/policy/preferences وUTC value
تغير إعداد التقويم بلا أثر	التكامل partial؛ تتبع consumer الفعلي في `config/project-desk.php` والخدمة

16. تعريف الإنجاز للتغيير

- [ ] Request وPolicy وService تفصل المسؤوليات وتمنع cross-project access.
- [ ] المعاملة/القفل/audit/timezone والأرشفة معالجة صراحة.
- [ ] migration وindexes والعلاقات واختبار upgrade صحيحة.
- [ ] Unit/Feature/browser بقدر الخطر، وكل quality gates ناجحة.
- [ ] RTL والوصول والاستجابة والتغييرات غير المحفوظة مجربة.
- [ ] الملفات/backup/search/notifications محدثة إن أصبح للميزة أثر فيها.
- [ ] لا أسرار، لا legacy route غير مقصود، ولا توسع محاسبي ضمن قالب الفاتورة.
- [ ] توثيق API/data/security/operations وSRS محدث ووضع القدرة صحيح.

DEPLOYMENT

1. نموذج النشر المدعوم

الإصدار الحالي مصمم لتطبيق ويب عربي داخلي على **مضيف واحد**:

```
خاص محلي storage + محلية SQLite -> PHP-FPM/Laravel -> reverse proxy/Nginx -> HTTPS المستخدمون عبر
```

وعلى نفس المضيف أو تحت نفس القرص/القفل يعمل scheduler. الأصول المبنية تخدم من `public/build`. البريد وماسح malware وخزن النسخ الخارجي تكاملات تشغيلية. لا توجد Jobs مجال مخصصة حالياً، لذلك queue worker ليس ضرورياً للسلوك الحالي، مع بقاء جداول/تهيئة queue scaffold.

النمط	الحالة
Linux أو Windows مضيف واحد بقرص محلي دائم	مدعوم وفق التهيئة والاختبار
Container واحد مع volume دائم لـSQLite وstorage	ممكن، ويحتاج اختبار permissions/signals/backup
عدة PHP replicas على SQLite/storage محلي	غير مدعوم

النمط	الحالة
SQLite على NFS/SMB	غير مدعوم/عالي المخاطر
نشر أفقي مع DB/objects/locks موزعة	مخطط مستقبلي، يحتاج إعادة تصميم adapters والأقفال والاستعادة

2. متطلبات الخادم

- PHP 8.3+؛ استخدم نسخة مدعومة أمنياً. CI الحالي يتحقق على PHP 8.4.
- Extensions: `bcmath,curl,dom,fileinfo,gd,intl,mbstring,pdo_sqlite,sqlite3,xml,zip`.
- Composer 2 لبناء artifact أو تثبيت vendor من lockfile.
- Node 22.12+ و pnpm 11.16 للبناء فقط؛ لا يلزمان runtime إذا شحنت `public/build`.
- Nginx/Apache أو خادم مكافئ مع PHP-FPM و TLS.
- cron/systemd/Supervisor لتشغيل scheduler.
- ماسح malware production حقيقي عبر driver command أو callback.
- SMTP/مزود بريد موثوق إذا استعملت verification/reset.
- مساحة محلية دائمة للقاعدة و WAL و private files والنسخ والسجلات، مع مراقبة وهامش للاستعادة staging.

لا تنشر إذا كان PHP CLI مختلفاً عن PHP-FPM في extensions أو config دون اختبار الاثنين. البيئة المحلية التي ينقصها `intl` أو تعمل بـNode 18 ليست مطابقة للبناء المعتمد.

3. بناء artifact

نفذ في CI نظيفة من lockfiles:

```
composer validate --strict --no-check-publish
composer install --prefer-dist --no-progress --optimize-autoloader
composer audit --locked --no-interaction
pnpm install --frozen-lockfile
pnpm run format:check
pnpm run lint:check
pnpm run types:check
pnpm run build
composer test
```

لا تضمّن `.env` أو SQLite أو `storage/app/private` أو logs أو browser auth state في artifact. ضمّن الشيفرة، `vendor` إن كان نموذج النشر artifact كامل، و`public/build` وملفات lock. سجّل commit SHA و checksums ونتيجة CI.

تنبيه: script المتصفح يشير حالياً إلى `tests/Browser` بينما المجلد `tests/browser`. يجب توحيد حالة الأحرف والتحقق على Linux قبل اعتماد browser gate.

4. إعدادات البيئة

استخدم secret manager أو ملف بيئة مقيد القراءة. الجدول يذكر الأسماء لا القيم:

المجموعة	المتغيرات الأساسية	قاعدة الإنتاج
التطبيق	APP_ENV, APP_KEY, APP_DEBUG, APP_URL, APP_TIMEZONE, BUSINESS_TIMEZONE, locales	production، debug false، URL HTTPS، APP_KEY ثابت وسري، business timezone مقصودة
قاعدة البيانات	DB_CONNECTION, SQLite path عند الحاجة، DB_BUSY_TIMEOUT, DB_JOURNAL_MODE, DB_SYNCHRONOUS, DB_TRANSACTION_MODE	SQLite محلية؛ القيم الافتراضية للمشروع تختبر قبل تغييرها
الجلسة	SESSION_DRIVER, SESSION_LIFETIME, SESSION_ENCRYPT, SESSION_DOMAIN, SESSION_SECURE_COOKIE, SESSION_HTTP_ONLY, SESSION_SAME_SITE	Secure، database مع HttpOnly، HTTPS، SameSite lax افتراضياً، domain ضيق
التخزين/queue/cache	FILESYSTEM_DISK, QUEUE_CONNECTION, CACHE_STORE	private/local والدatabase حسب .env.example؛ لا تجعل private files public
البريد	MAIL_MAILER, host/port/user/password/from	مزود حقيقي، لا log لبيانات حساسة
malware	MALWARE_SCANNER_DRIVER, executable/arguments/timeout أو callback	command/callback حقيقي؛ none مرفوض عملياً في production
الرفع	UPLOAD_MAX_FILE_KB, rate/quota/file limit/orphan retention	متسقة مع Nginx وPHP ومساحة القرص
Data Center	CSV_MAX_*, XLSX_MAX_*	أبق الحدود الدفاعية واختبر الملفات القصوى
backup	BACKUP_MAX_*, BACKUP_DISK, BACKUP_FILE_DISKS, BACKUP_ENCRYPTION_KEY, BACKUP_PREVIOUS_ENCRYPTION_KEYS, restore TTL/attempts	مفتاح مستقل، previous للقراءة أثناء rotation، قرص خاص وoff-host copy
logs	LOG_CHANNEL, LOG_LEVEL	دوران وتنبيه وتنقيح أسرار؛ info أو أشد حسب السياسة

لا تضع أسراراً في VITE_*؛ هذه القيم تدخل bundle عام. لا تغير APP_KEY أو BACKUP_ENCRYPTION_KEY في مكانها بلا اختبار فك البيانات/النسخ ومهلة تدوير.

5. نظام الملفات والصلاحيات

وثّق مسار الإصدار الفعلي، مثلاً /var/www/project-desk/current. القواعد:

- document root هو public فقط.
- مستخدم PHP يكتب إلى storage وbootstrap/cache وملف SQLite **ومجلده**.
- ملفات الشيفرة وenv. لا يكتبها مستخدم الويب؛ env. يقرأه فقط مستخدم الخدمة/النشر المناسب.
- storage/app/private وbackup directory غير قابلين للوصول المباشر عبر Nginx.
- احتفظ بضعفي مساحة أكبر عملية restore تقريباً، لأن staging وpre_restore قد يتعايشان.

مثال أوامر Linux يجب تكييف المستخدم والمسار قبل التنفيذ:

```
chown -R deploy:www-data /var/www/project-desk
chown -R www-data:www-data /var/www/project-desk/storage /var/www/project-desk/bootstrap/cache /var/www/project-desk/database
chmod -R u=rwX,g=rwX,o= /var/www/project-desk/storage /var/www/project-desk/bootstrap/cache /var/www/project-desk/database
chmod u=rw,g=r,o= /var/www/project-desk/.env
```

لا تنسخ المثال حرفياً إن اختلف نموذج الملكية؛ اختبر القراءة/الكتابة كمستخدم PHP الفعلي.

6. إعداد خادم الويب

Nginx يجب أن يطبق TLS ويخدم الملفات الموجودة من `public` ويرسل غيرها إلى `public/index.php`. الحد الأدنى المنطقي:

```
root /var/www/project-desk/current/public;
index index.php;

location / {
    try_files $uri $uri/ /index.php?$query_string;
}

location ~ \.php$ {
    include fastcgi_params;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    fastcgi_pass unix:/run/php/php8.4-fpm.sock;
}

location ~ /\. {
    deny all;
}
```

أضف TLS/HSTS بعد اختبار النطاق، وحدود body/timeouts المتسقة مع upload والـPDF/import. لا تضف alias إلى `storage/app/private`. رؤوس التطبيق موجودة، لكن لا توجد CSP عامة شاملة للصفحات حالياً؛ أي CSP في proxy يجب اختبارها مع Inertia/Vite/الخطوط وPasskeys.

اضبط trusted proxy/protocol في البنية حتى يرى Laravel HTTPS الحقيقي، وإلا قد تتأثر URLs وSecure cookies.

7. scheduler والخدمات

cron القياسي:

```
* * * * * cd /var/www/project-desk/current && php artisan schedule:run >> /dev/null 2>&1
```

يفضل توجيه stdout/stderr إلى logging مراقب بدلاً من فقده. بديل systemd هو `php artisan schedule:work` مع restart policy. المهام: automatic backup كل دقيقة، notification sync كل دقيقة، orphan pruning 03:30 بتوقيت العمل.

لا تشغل نسختين scheduler على نفس نشر v1 بلا فهم الأقفال. إن أضيف queue worker لاحقاً، شغله عبر systemd/Supervisor وحدد retries/timeouts و`queue:restart` في النشر، وأوقفه أثناء restore.

8. أول نشر

1. أنشئ مستخدم الخدمة والمجلدات والـTLS والـsecret file.
2. ضع artifact واضبط symlink `current` إن استعملت releases ذرية.
3. أنشئ SQLite الفارغة والمسارات الخاصة بالصلاحيات الصحيحة.
4. نفذ:

```
php artisan project-desk:ensure-app-key
php artisan migrate --force
php artisan db:seed --class=WorkflowStatusSeeder --force
php artisan optimize
php artisan project-desk:provision-admin
```

1. شغّل PHP-FPM/scheduler، ثم افتح `/up` و`/login` عبر HTTPS.
2. سجل دخول المدير، غير/أكد بياناته، فعّل 2FA/Passkey حسب السياسة، واختبر البريد.
3. اضبط company/general/notifications/automatic backup/calendar، مع العلم أن calendar/timezone integration جزئي في بعض الخدمات.
4. اختبر upload نظيفاً وخبيثاً، PDF قالب فاتورة، بحثاً scoped، نسخة `.pdesk`. واستعادة في بيئة معزولة.

`project-desk:ensure-app-key` لا يستبدل مفتاحاً موجوداً. provisioning لا يجوز أن ينتج credential مشتركاً أو محفوظاً في script.

9. ترقية إصدار قائم

نفذ نافذة تغيير معلنة؛ SQLite والمضيف الواحد يفضلان إيقاف الكتابة أثناء migration:

1. تحقق من CI/artifact/checksum ومتغيرات البيئة الجديدة ومساحة القرص.
2. أنشئ نسخة `.pdesk` كاملة مؤكدة وانقل نسخة off-host.
3. سجل baseline: `/up`، نسخة التطبيق/counts، commit/عينات، آخر scheduler/backup.
4. أدخل الصيانة: `php artisan down`، وأوقف scheduler وأي worker مخصص.
5. انشر artifact جديداً دون مسح SQLite, `.env` shared, `storage`.
6. نفذ `php artisan migrate --force` ثم `php artisan optimize`.
7. أعد تشغيل PHP-FPM لتفريغ OPcache، ثم scheduler/workers.
8. `php artisan up` ونفذ smoke tests.
9. راقب الأخطاء والقفل والقرص والمستخدمين أول فترة بعد النشر.

لا تشغل `migrate:fresh` ولا تستبدل قاعدة الإنتاج. لا تعتمد على `migrate:rollback` إذا كانت migration حذفت/حوّلت بيانات؛ استخدم خطة الإصدار ونسخة `.pdesk` المختبرة.

10. Smoke tests بعد النشر

- [ ] `GET /up` ناجح، و`/login` يحمل assets بلا mixed content.

- [ ] login/logout وCSRF والجلسة وemail flow حسب البيئة.
- [ ] dashboard ومشروع مسموح ومشروع غير مسموح لا يتسربان.
- [ ] إنشاء/تحديث مهمة مع start/end و409 عند نسخة قديمة.
- [ ] upload نظيف ينزل بعد scan؛ infected/failure لا ينزل.
- [ ] بحث لا يظهر بيانات مشروع غير مصرح.
- [ ] قالب فاتورة وPDF يعملان؛ لا تظهر وظائف محاسبة/دفع.
- [ ] scheduler يحدّث التنبيهات والنسخ، وpruner لا يحذف مرجعاً دائماً.
- [ ] إنشاء `pdesk.` والتحقق منها؛ لا تنفذ restore على الإنتاج للاختبار.
- [ ] logs/request IDs بلا أسرار ووقت UTC/business صحيح.

11. الرجوع

قرار الرجوع إذا فشل migration، تعذر login/authorization، ظهر فساد قاعدة/ملفات، scanner أصبح fail-open، أو تجاوز الخطأ/SLO حد التغيير.

التسلسل الآمن:

1. أدخل الصيانة وأوقف الكتاب/جدولة.
2. احتفظ بسجلات ونسخة الحالة الفاشلة للتحقيق؛ لا تكتب فوقها عشوائياً.
3. إن كانت الشيفرة فقط ولم تتغير schema، أعد symlink/artifact السابق وأعد PHP-FPM.
4. إن تغيرت البيانات/schema، استخدم تدفق الاستعادة الرسمي للنسخة pre-deploy/`pre_restore` مع العبارة/checksum/nonce، لا نسخ ملف SQLite وهو حي.
5. استعد الملفات مع القاعدة كحزمة واحدة؛ اعزل WAL/SHM وفق runbook.
6. نفذ smoke tests وأعد scheduler ثم اخرج من الصيانة.
7. وثق timeline، الأثر، RPO/RTO، السبب والإجراء التصحيحي.

12. المراقبة والسعة

راقب `/up` و`/login/`، SQLite lock time، scan/mail failures، scheduler heartbeat، request/correlation IDs، 5xx/latency، WAL/DB size، مساحة/inodes، آخر backup وoff-host copy، وفشل restore/import. التطبيق لا يتضمن منصة observability كاملة.

حدود السعة:

- SQLite تسلسل الكتابة ويحتاج `busy_timeout` وtransactions قصيرة.
- توليد PDF/import/backup/scan متزامن قد يستهلك worker وزمناً؛ راقب timeouts والذاكرة.
- storage المحلي والـflock يمنعان التوسع الأفقي الساذج.
- قبل التوسع: انقل DB إلى خادم مناسب، files إلى object storage خاص، locks/cache/session إلى خدمة مشتركة، backup إلى adapter جديد، ثم أعد اختبارات consistency/restore/authorization.

13. hardening قبل Go-Live

- [ ] `APP_ENV=production`, `APP_DEBUG=false`, TLS وSecure/HttpOnly/SameSite cookies.
- [ ] أسرار فريدة في rotation، secret manager وbreak-glass موثقان.
- [ ] mailer حقيقي وscanner fail-closed مثبت باختبار.
- [ ] private/backup/database غير مكشوفة، least privilege ونسخ off-host.
- [ ] NTP/timezone وPHP/Nginx upload limits متسقة.
- [ ] Composer audit وOS/PHP/Nginx patching وSBOM/فحص artifact حسب السياسة.
- [ ] scheduler مراقب، logs مدوّرة ومنقحة، alerts فعالة.
- [ ] restore drill ناجح ومؤرخ، مع ربط owner وRPO/RTO.
- [ ] مراجعة الأدوار، خصوصاً تضارب global viewer مع project manager، قبل منح الحسابات.
- [ ] CSP عامة إما منفذة ومختبرة أو المخاطرة موثقة؛ لا تدّع وجودها.

14. قائمة إصدار نهائية

- [ ] commit/tag وartifact/checksums ونتائج CI/browser مثبتة.
- [ ] migration upgrade/fresh/restore compatibility اختبرت.
- [ ] release notes وenv delta وrunbook/rollback وملاك القرار جاهزون.
- [ ] `.pdesk` pre-deploy صالحة وoff-host ومتاحة بالمفتاح الصحيح.
- [ ] نافذة الصيانة والإبلاغ والدعم بعد النشر محددة.
- [ ] smoke/security/accessibility/RTL وinvoice PDF مكتملة.
- [ ] scheduler/scanner/mail/backup/monitoring مثبتة من بيئة النشر.
- [ ] لا أسرار أو debug أو test admin أو بيانات browser في artifact.
- [ ] القيود المعلنة واضحة: templates، SQLite، single-host غير محاسبية، calendar integration جزئي.

DOCUMENTATION GOVERNANCE

1. الغرض

تحدد هذه الوثيقة كيفية صيانة كراسة مواصفات المتطلبات والتوثيق التقني ودليل المستخدم، بحيث تظل الوثائق متزامنة مع النظام ولا تتحول إلى وصف تاريخي مضلل.

2. مجموعة الوثائق الحاكمة

الوثيقة	الغرض	مالك المراجعة المقترح	متى تُحدّث؟
`docs/specification/PROJECT_DESK_SRS_IEEE830_AR.md`	خط أساس المتطلبات، الحدود، حالات الاستخدام والتتبع	مالك المنتج + المسؤول التقني	عند تغيير نطاق أو قاعدة عمل أو متطلب غير وظيفي
`docs/specification/REQUIREMENTS_MATRIX.csv`	سجل آلي مختصر للمعرّفات والحالة والأدلة	QA + مالك المنتج	مع كل تغيير لمتطلب أو اختبار
`docs/technical/DOCUMENTATION_INDEX_AR.md`	بوابة الدخول للتوثيق التقني	المسؤول التقني	عند إضافة أو إزالة وثيقة تقنية
`docs/technical/ARCHITECTURE_AR.md`	المعمارية والحدود والتدفقات	المهندس المسؤول	عند تغيير مكون أو اعتماد أو قرار معماري
`docs/technical/DATA_MODEL_AR.md`	الجداول والعلاقات والقيود	مالك قاعدة البيانات	مع كل migration أو تغيير علاقة
`docs/technical/API_AND_ROUTES_AR.md`	المسارات والعقود والأخطاء	مالك الخادم	عند تغيير route/request/response
`docs/technical/SECURITY_AND_PERMISSIONS_AR.md`	المصادقة والصلاحيات والضوابط	مسؤول الأمن	عند تغيير Policy أو middleware أو secret
`docs/technical/OPERATIONS_AND_RECOVERY_AR.md`	التشغيل والمراقبة والنسخ والاستعادة	مسؤول التشغيل	عند تغيير بيئة أو أمر أو سياسة تعافٍ
`docs/technical/TESTING_AND_QA_AR.md`	استراتيجية الاختبار وبوابات الإصدار	QA Lead	عند تغيير suite أو بوابة جودة
`docs/USER_MANUAL_AR.md`	تشغيل النظام من منظور المستخدم	مالك المنتج/التدريب	عند تغيير واجهة أو رحلة مستخدم

3. مصدر الحقيقة

ترتيب الحسم عند التعارض هو: التنفيذ الحالي في `/app`, `/routes`, `/database/migrations`, `/resources`, و`/config`، ثم الاختبارات في `/tests`، ثم الوثائق. لا يكفي وجود كلاس أو جدول لإعلان ميزة؛ يجب إثبات المسار القابل للاستخدام والصلاحية والاختبار المناسب.

4. قواعد التغيير

1. يجب أن يتضمن أي تغيير نطاقي تحديث معرّفات `FR-*` أو `BR-*` أو `DR-*` أو `NFR-*` ذات الصلة.
2. يجب ألا يُعاد استخدام معرّف متطلب محذوف لمعنى جديد؛ يوضع المتطلب القديم كـDeprecated أو Out of scope.
3. يجب أن يحدد كل متطلب حالة `Implemented` أو `Partial` أو `Planned` أو `Out of scope`.
4. يجب تحديث دليل التنفيذ والاختبار في نفس طلب التغيير الذي يغيّر السلوك.
5. يجب عدم تضمين كلمات مرور أو مفاتيح أو بيانات إنتاج أو روابط تخزين خاصة في الوثائق.
6. يجب إعادة بناء PDF/DOCX بعد تعديل المصادر، ثم تدقيق الوصولية والخصوصية وفحص الصفحات بصرياً.

5. تعريف اكتمال التوثيق

- لا توجد متطلبات بلا معرّف أو حالة.
- لا توجد وحدة أو route إنتاجي مهم بلا وصف تقني.
- لا توجد صلاحية حساسة موثقة دون Policy/middleware أو اختبار سلبي مقابل.
- ترتبط الخصائص الحرجة باختبار أو تُصنّف صراحة كفجوة.
- تذكر الحدود الخارجية للإطلاق مثل ماسح البرمجيات الخبيثة، TLS، النسخ خارج المضيف وتمرين الاستعادة.
- تطابق قوالب الفواتير نطاقها غير المحاسبي ولا تظهر كدفتر تحصيل أو نظام محاسبي.

6. دورة المراجعة

- مراجعة خفيفة عند كل إصدار.
- مراجعة تتبع كاملة شهرياً أثناء التطوير النشط.
- مراجعة أمن وتشغيل قبل كل نشر إنتاجي.
- مراجعة شاملة بعد تغيير مخطط البيانات أو الصلاحيات أو النسخ والاستعادة.