

في نظام (Noon Payment) دليل الدفع الإلكتروني AdvFood

بدءاً من الإعداد وحتى تدفق، Noon Payments مع بوابة الدفع AdvFood يوفر هذا الدليل نظرة شاملة على كيفية دمج نظام Webhooks العمل، مروراً بمصادر الكود ذات الصلة، وآليات التعامل مع الأخطاء والـ.

الفهرس

1. نظرة عامة على التدفق
 2. الإعدادات ومتغيرات البيئة
 3. الملفات والمسارات الأساسية
 4. تدفق العمل التفصيلي
 5. السجلات (Logging)
 6. الجداول المتأثرة
 7. نقاط التكامل الإضافية
 8. استكشاف الأخطاء وإصلاحها
 9. توصيات التشغيل
 10. ملفات التوثيق داخل المشروع
 11. خاتمة
-

نظرة عامة على التدفق 1.

1. `checkout/customer-details` المستخدم يدخل بياناته عبر صفحة.
2. `order_number` وتوليد رقم `pending` بحالة دفع (`Order`) يتم إنشاء طلب في النظام.
3. (`checkoutData.postUrl`) للحصول على رابط الدفع `INITIATE` عبر واجهة Noon يتم إنشاء طلب إلى.
4. إلى `Webhook` ويُرسَل `/payment-success` المستخدم إلى صفحة Noon بعد الدفع، يعيد `/api/webhook`.
5. إصدار الفاتورة، وإرسال الطلب للشحن، `paid` يتم تحديث الطلب إلى (`fallback` أو `Webhook` عند تأكيد الدفع) عبر.

الإعدادات ومتغيرات البيئة 2.

`.env`. يجب ضبط المتغيرات التالية في ملف

```
NOON_API_URL=https://api-test.sa.noonpayments.com
NOON_API_KEY=...
NOON_BUSINESS_ID=adv_food
NOON_APPLICATION_ID=adv-food
NOON_SUCCESS_URL=https://advfoodapp.clarastars.com/payment-success
NOON_FAILURE_URL=https://advfoodapp.clarastars.com/payment-failed
```

للمزيد من الأمثلة، راجع

```
README_NOON_PAYMENT.md – NOON_PAYMENT_INTEGRATION_GUIDE.md –
NOON_SUCCESS_ANALYSIS.md.
```

الملفات والمسارات الأساسية 3.

3.1 المتحكم الرئيسي: TestNoonController

المسار: `app/Http/Controllers/TestNoonController.php`
يحتوي على:

| الوصف | الدالة |
|--|---|
| ويرجع رابط الدفع Noon ينشئ طلب | <code>createPayment()</code> |
| يعالج التوجيه بعد الدفع ويبحث عن الطلب | <code>success()</code> |
| عرض صفحة فشل الدفع | <code>fail()</code> |
| اختبارات وتشخيصات | <code>checkApiStatus</code> , <code>testConnection</code> , <code>quickTest</code> , إلخ |

3.2 المسارات في `routes/web.php`

```
Route::get('/pay', [TestNoonController::class,
    'createPayment'])->name('payment.create');
Route::get('/payment-success', [TestNoonController::class,
    'success'])->name('payment.success');
Route::get('/payment-failed', [TestNoonController::class,
    'fail'])->name('payment.fail');
Route::get('/noon/status', [TestNoonController::class,
    'checkApiStatus']);
```

3.3 مسار الـ Webhook (`routes/api.php`)

```
Route::post('/webhook', [PaymentWebhookController::class,
    'handleNoon']);
```

3.4 PaymentWebhookController

المسار: `app/Http/Controllers/PaymentWebhookController.php`

```

        public function handleNoon(Request $request)
        {
            $payload = $request->all();
            $orderReference = $this->extractOrderReference($payload);
            $order = Order::where('payment_order_reference', $orderReference)
                ->orWhere('order_number', $orderReference)
                ->orWhere('id', $orderReference)
                ->latest()
                ->first();

            if ($this->isPaymentSuccessful($payload)) {
                $order->payment_status = 'paid';
                $order->status = 'confirmed';
                $order->payment_order_reference = $orderReference;
                $order->save();
            }

            return response()->json(['status' => 'processed']);
        }

```

تستخرج المرجع المناسب، الدالة `extractOrderReference` والدالة `isPaymentSuccessful` تتحقق من نجاح الدفع (success, paid, resultCode, ...).

3.5 نموذج الطلب (Order Model)

يرتبط الدفع بالشحن من خلال حدث التحديث

```

        static::updated(function ($order) {
            if ($order->wasChanged('payment_status') && $order->payment_status
                === 'paid') {
                (new ShippingService())->createOrder($order);
            }
        });

```

تدفق العمل التفصيلي 4.

1. `checkout/customer-details` جمع بيانات العميل عبر
2. وتخزين العنوان ورقم الطلب `pending` إنشاء الطلب بحالة
3. Noon لإرسال طلب إلى `RestLinkController::initiatePayment` بدء الدفع باستخدام
4. `/payment-success` التحويل بعد الدفع إلى
5. `/api/webhook` على Noon من **Webhook** استقبال
6. إصدار الفاتورة → إرسال الطلب للشحن → **paid** تحديث الحالة إلى

كحل طارئ `allow_direct_update=1` يمكن استخدام Webhook، في حال تأخر الـ

5. السجلات (Logging)

`storage/logs/laravel.log` جميع العمليات تُسجل في

| الحدث | أمثلة على السجلات |
|-------------|--|
| بدء الدفع | Noon Payment Request / Sending request to Noon |
| استلام الرد | Noon Payment Response, Payment Failed, Error Code |
| نجاح الدفع | PAYMENT SUCCESS CALLBACK STARTED, Order marked as paid via webhook |

6. الجداول المتأثرة

| الجدول | الاستخدام |
|------------------|--|
| orders | تحديث <code>payment_status</code> , <code>status</code> , <code>payment_order_reference</code> |
| online_customers | تسجيل بيانات العملاء للتحليل |
| invoices | إصدار تلقائي عند تأكيد الدفع |
| shipping_orders | إنشاء سجل بعد الدفع الناجح وإرسال الطلب |

7. نقاط التكامل الإضافية

RestLinkController

المسار: `app/Http/Controllers/RestLinkController.php`
`initiatePayment` يُنشئ الطلب ويستدعي

جداول التحليل

- **online_customers:** غير مرتبطة مباشرة بالدفع لكنها مهمة للتحليلات
-

استكشاف الأخطاء وإصلاحها 8.

| المشكلة | الرسالة المحتملة | الحل |
|----------------------|---|--|
| Noon بيانات ناقصة | Payment configuration error / Noon API credentials not configured | php وشغل .env. تأكد من إعدادات artisan config:clear |
| فشل إنشاء الطلب | Payment request failed | (amount, reference) تحقق من القيم المرسلة |
| رابط الدفع مفقود | checkout URL missing | تحقق من صلاحيات الحساب أو صحة الطلب |
| حالة الدفع لا تتغير | لم يصل Webhook | إلى Webhook لإرسال Noon تأكد من إعدادات /api/webhook |
| لا Webhook يجد الطلب | Order not found | تحقق من وجود الصحيح merchantOrderReference |

توصيات التشغيل 9.

- Sandbox. اختبر أولاً في بيئة.
 - أثناء الاختبار والإنتاج Laravel راقب سجلات.
 - Webhooks لتلقي SSL و DNS تأكد من إعداد.
 - ومعالجة الأخطاء بسرعة Webhooks عين مسؤولين لمراقبة.
 - . أنشئ تنبيهات تلقائية عند فشل إشعارات الدفع.
-

ملفات التوثيق داخل المشروع 10.

| المحتوى | الملف |
|--------------------------------|-----------------------------------|
| مقدمة وخطوات الإعداد | README_NOON_PAYMENT.md |
| شرح مفصل لعملية الدمج | NOON_PAYMENT_INTEGRATION_GUIDE.md |
| تحليل نجاح الاتصال | NOON_SUCCESS_ANALYSIS.md |
| تقارير المشاكل السابقة وحلولها | NOON_ISSUE_REPORT.md |

خاتمة 11.

من البداية حتى نهاية عملية الدفع وربطها بالشحن AdvFood داخل نظام Noon Payments هذا الدليل يشرح عملية تكامل والفوترة. لضمان سير العملية بسلاسة logs يُنصح بمراجعة هذا المستند عند أي تحديث أو خلل في الدفع، ومتابعة الـ
