

في نظام (Noon Payment) دليل الدفع الإلكتروني AdvFood

بدءً من الإعداد وحتى تدفق، يوفر هذا الدليل نظرة شاملة على كيفية دمج نظام **Noon Payments** مع بوابة الدفع **AdvFood** مع مروءة الكود ذات الصلة، والآليات التعامل مع الأخطاء والـ **Webhooks**.

الفهرس

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

نظرة عامة على التدفق .1

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

الإعدادات ومتغيرات البيئة .2

يجب ضبط المتغيرات التالية في ملف `.env`:

```
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:

الدالة	الوصف
createPayment()	ويرجع رابط الدفع Noon ينشئ طلب
success()	يعالج التوجيه بعد الدفع ويبحث عن الطلب
fail()	عرض صفحة فشل الدفع
اختبارات وتشخيصات	checkApiStatus, testConnection, quickTest, 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` تتحقق من نجاح الدفع (`success, paid, resultCode, ...`). والدالة `isPaymentSuccessful`

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

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

5. السجلات (Logging)

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

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

الجداول المتأثرة . 6

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

نقاط التكامل الإضافية . 7

RestLinkController

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

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	وشنّع <code>env</code> . تأكّد من إعدادات <code>artisan config:clear</code>
فشل إنشاء الطلب	Payment request failed	تحقق من القيم المرسلة (amount, reference)
رابط الدفع مفقود	checkout URL missing	تحقق من صلاحيات الحساب أو صحة الطلب
حالة الدفع لا تتغيّر	Webhook لم يصل	إلى Noon لإرسال Webhook تأكّد من إعداد <code>/api/webhook</code>
Webhook لا يجد الطلب	Order not found	تحقق من وجود الصحيح <code>merchantOrderReference</code>

توصيات التشغيل .9

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

ملفات التوثيق داخل المشروع .10

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

خاتمة .11

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