

 الفهرس

- نظرة عامة على المعمارية
 - الإعدادات ومتغيرات البيئة
 - نموذج البيانات والقواعد
 - تدفق العمل داخل نموذج الطلب
 - خدمة الشحن (ShippingService)
 - نقاط التكامل (Controllers & Routes)
 - خطوات التشغيل النموذجية
 - استكشاف الأخطاء وإصلاحها
 - ممارسات موصى بها
 - ملحق: أهم الرسائل في Logs
 - خاتمة.

نظرة عامة على المعمارية .1

1. إنشاء الطلب: يتم إنشاء الطلب أولاً عبر قنوات السلة أو الروابط، ويُخزن في جدول `orders`.
يتم استدعاء خدمة الشحن تلقائياً لإرسال الطلب إلى شركة `paid`، إلى `payment_status` تأكيد الدفع: عند تغيير التوصيل.
2. خدمة الشحن (ShippingService):
 - بناء الطلب والتحقق من البيانات قبل الإرسال.
 - (الخارجي (إنشاء / استعلام / إلغاء API التواصل مع الـ Webhooks).
 - حفظ نسخة محلية في `shipping_orders`.
 - وتحديث الحالة والسائل `Webhooks` استقبال الـ.
3. للاستعلام عن حالة الشحنة أو إلغائها واستقبال التحديثات Endpoints متابعة الحالة: يوفر النظام.
4. للإشعار عن حالة الشحنة أو إلغائها واستقبال التحديثات `Endpoints` متابعة الحالة: يوفر النظام.

الإعدادات ومتغيرات البيئة .2

ضمن المفتاح config/services.php الإعدادات موجودة في `shipping`. يجب ضبط القيم التالية في `.env`:

```
'shipping' => [  
    'url' => env('SHIPPING_API_URL'),  
    'key' => env('SHIPPING_API_KEY'),  
    'endpoints' => [  
        'create' => env('SHIPPING_API_ENDPOINT_CREATE', '/orders'),  
        'status' => env('SHIPPING_API_ENDPOINT_STATUS',  
                         '/orders/{id}'),  
        'cancel' => env('SHIPPING_API_ENDPOINT_CANCEL',  
                         '/orders/{id}'),  
    ],  
    'cancel_method' => env('SHIPPING_API_CANCEL_METHOD', 'delete'),  
    'send_as_form' => env('SHIPPING_API_SEND_AS_FORM', false),  
],
```

أهم المتغيرات

المتغير	الوصف
SHIPPING_API_URL	شركة الشحن API عنوان
SHIPPING_API_KEY	مفتاح الوصول (Bearer Token)
SHIPPING_API_ENDPOINT_*	نقطة النهاية للإنشاء / الحالة / الإلغاء
SHIPPING_API_CANCEL_METHOD	طريقة الإلغاء (delete أو post)
SHIPPING_API_SEND_AS_FORM	عند التفعيل يرسل الطلب بصيغة form-data
SHIPPING_API_VERIFY_SSL	اختياري، لتعطيل التحقق من SSL

نموذج البيانات والقواعد .3

جدول shipping_orders

يحفظ آخر حالة معروفة لدى شركة الشحن لكل طلب

```
$table->unsignedInteger('order_id');
        $table->string('shop_id', 50);
        $table->string('dsp_order_id', 100)->nullable();
$table->string('shipping_status', 50)->default('New Order');
        $table->string('recipient_name', 191);
        $table->string('recipient_phone', 20);
        $table->text('recipient_address');
        $table->decimal('latitude', 10, 6)->nullable();
        $table->decimal('longitude', 10, 6)->nullable();
        $table->string('driver_name', 191)->nullable();
        $table->string('driver_phone', 20)->nullable();
        $table->decimal('driver_latitude', 10, 6)->nullable();
        $table->decimal('driver_longitude', 10, 6)->nullable();
        $table->decimal('total', 10, 2);
```

```
$table->tinyInteger('payment_type');  
$table->text('notes')->nullable();
```

أو تنفيذ استعلام الحالة Webhook يتم تحديث الجدول تلقائياً عند إرسال الطلب أو استقبال

تدفق العمل داخل نموذج الطلب 4.

يتم الإرسال تلقائياً عند تحقق شرط الدفع

```
static::created(function ($order) {  
if (!empty($order->shop_id) && $order->payment_status === 'paid')  
{  
    (new \App\Services\ShippingService())->createOrder($order);  
}  
});  
  
static::updated(function ($order) {  
if ($order->wasChanged('payment_status') && $order->payment_status  
      === 'paid') {  
    if (empty($order->dsp_order_id) && !empty($order->shop_id)) {  
        (new  
            \App\Services\ShippingService())->createOrder($order);  
    }  
}  
});  
});
```

خدمة الشحن 5. (ShippingService)

المسار: app/Services/ShippingService.php

الوظائف الأساسية

الوظيفة	الوصف
createOrder(\$order)	بناء الطلب والتحقق والإرسال إلى شركة الشحن
getOrderStatus(\$shippingOrderId)	جلب حالة الطلب من API
cancelOrder(\$shippingOrderId)	تنفيذ الإلغاء وتحديث السجلات محلياً
handleWebhook(Request \$request)	استقبال التحديثات من شركة الشحن
تتابع مساعدة	mapPaymentType / buildUrl / flattenArray

6. نقاط التكامل (Controllers & Routes)

المسارات في `routes/api.php`

```
Route::post('/shipping/webhook', [ShippingController::class,
                                  'handleWebhook']);
Route::get('/shipping/status/{dspOrderId}', [ShippingController::class, 'getStatus']);
Route::post('/shipping/cancel/{dspOrderId}', [ShippingController::class, 'cancel']);
```

ShippingController

- `handleWebhook`: استقبال إشعارات المزود.
 - `getStatus`: جلب حالة الشحنة.
 - `cancel`: تنفيذ الإلغاء.
 - `createOrder`: للإرسال اليدوي لغرض الاختبار.
-

خطوات التشغيل النموذجية 7.

1. ضبط متغيرات البيئة (`SHIPPING_API_*`)
 2. إنشاء الطلب عبر النظام
 3. عند الدفع يتم الإرسال تلقائياً
 4. التحقق من السجلات (`laravel.log`)
 5. لفحص الحالة استخدام `/api/shipping/status/{id}`
 6. لتحديث السجلات آلياً مراقبة `Webhooks`
-

استكشاف الأخطاء وإصلاحها 8.

المشكلة	السبب المحتمل	الإجراء المقترن
<code>Invalid Shop (422)</code>	غير <code>shop_id</code> صحيح	تأكد من الرابط الصحيح للمطعم
<code>401 Unauthorized</code>	غير صالح API مفتاح	. تحديث القيم في <code>env</code>
<code>ConnectionException</code>	فشل الشبكة أو SSL	تأكد من الاتصال وإعدادات SSL
لا يصل <code>Webhook</code>	عنوان خاطئ	تأكد من التوجيه إلى <code>.../api/shipping/webhook</code>
لا يوجد سجل في <code>shipping_orders</code>	فشل الإرسال	أو استعلم يدوياً بالحالة <code>logs</code> تتحقق من

ممارسات موصى بها 9.

- تجربة النظام في بيئة اختبارية أولاً
 - بعد كل نشر أو تعديل logs مراقبة الـ
 - صحيح لكل مطعم shop_id التأكد من وجود
 - حذف بيانات الاختبار بعد التحقق
 - إنشاء تنبیهات للأخطاء الحرجة مثل Invalid shop Missing credentials
-

ملحق: أهم الرسائل في الـ Logs 10.

الرسالة	المعنى	الإجراء المقتضي
📦 SHIPPINGSERVICE::createOrder CALLED	بدأ إرسال الطلب	تحقق من نتيجة السجل التالي
✖ Shipping API credentials missing!	إعدادات ناقصة	أكمل بيانات .env
✓ Shipping API Response Received	استجابة ناجحة	تحقق من حفظ السجل
🚚 Order successfully sent...	تم الإرسال بنجاح	راقب shipping_orders
🔴 Validation Error (422)	خطأ تحقق	والبيانات shop_id تأكيد من

خاتمة 11.

من حيث التكوين، وتدفق البيانات، والربط مع الأنظمة الأخرى. **AdvFood** يغطي هذا الدليل كل ما يتعلق بخدمة الشحن في بانتظام للتأكد من سلامة التكامل logs يُنصح بالرجوع إليه عند تنفيذ أي تعديل أو مواجهة مشكلة في الشحن، مع متابعة الـ