

1. نظرة عامة على المعمارية
  2. الإعدادات ومتغيرات البيئة
  3. نموذج البيانات والقواعد
  4. تدفق العمل داخل نموذج الطلب
  5. خدمة الشحن (ShippingService)
  6. نقاط التكامل (Controllors & Routes)
  7. خطوات التشغيل النموذجية
  8. استكشاف الأخطاء وإصلاحها
  9. ممارسات موصى بها
  10. Logs ملحق: أهم الرسائل في الـ
  11. خاتمة
-

## نظرة عامة على المعمارية 1.

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

---

## الإعدادات ومتغيرات البيئة 2.

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

```
'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	(Bearer Token) مفتاح الوصول
SHIPPING_API_ENDPOINT_*	نقاط النهاية للإنشاء / الحالة / الإلغاء
SHIPPING_API_CANCEL_METHOD	(delete أو post) طريقة الإلغاء
SHIPPING_API_SEND_AS_FORM	form-data عند التفعيل يرسل الطلب بصيغة
SHIPPING_API_VERIFY_SSL	SSL اختياري، لتعطيل التحقق من

## نموذج البيانات والقواعد 3.

### shipping\_orders جدول

يحفظ آخر حالة معروفة لدى شركة الشحن لكل طلب

```
$table->unsignedBigInteger('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);  
        }  
    }  
});
```

---

## 5. خدمة الشحن (ShippingService)

المسار: [app/Services/ShippingService.php](#)

### الوظائف الأساسية

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

---

## 6. نقاط التكامل (Controllers & Routes)

**routes/api.php** المسارات في

```
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 {id}/api/shipping/status/ استخدام
6. لتحديث السجلات آلياً Webhooks مراقبة






## استكشاف الأخطاء وإصلاحها 8.

المشكلة	السبب المحتمل	الإجراء المقترح
Invalid Shop (422)	shop_id غير صحيح	تأكد من الربط الصحيح للمطعم
401 Unauthorized	غير صالح API مفتاح	.env تحديث القيم في
ConnectionException	SSL فشل الشبكة أو	SSL تأكد من الاتصال وإعدادات
Webhook لا يصل	عنوان خاطئ	تأكد من التوجيه إلى .../api/shipping/webhook
shipping_orders لا يوجد سجل في	فشل الإرسال	أو استعلم يدوياً بالحالة logs تحقق 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
- **Invalid shop** أو **Missing credentials** إنشاء تنبيهات للأخطاء الحرجة مثل

## Logs ملحق: أهم الرسائل في الـ 10.

الرسالة	المعنى	الإجراء المقترح
 <b>SHIPPINGSERVICE::createOrder CALLED</b>	بدأ إرسال الطلب	تحقق من نتيجة السجل التالي
 <b>Shipping API credentials missing!</b>	إعدادات ناقصة	<b>.env</b> أكمل بيانات
 <b>Shipping API Response Received</b>	استجابة ناجحة	تحقق من حفظ السجل
 <b>Order successfully sent...</b>	تم الإرسال بنجاح	<b>shipping_orders</b> راقب
 <b>Validation Error (422)</b>	خطأ تحقق	والبيانات <b>shop_id</b> تأكد من

## 11. خاتمة

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