

API DeCaPortes v1

Integración del DeCA electrónico (Orden FOM/2861/2012 y Resolución DGTCF de 5 de junio de 2026) con ERP, TMS, WMS y programas de transporte.

Versión	1.0 · 4 de octubre de 2026
URL base	https://decaportes.com/api/v1
Especificación	OpenAPI 3.1 · https://decaportes.com/api/v1/openapi.json
Documentación en línea	https://decaportes.com/api-docs
Contacto	info@decaportes.com · 614 096 823

DeCaPortes™ · Inversiones Sijebe, S.L. · en colaboración con Astotrans. Este documento describe la API en la fecha indicada; la referencia vigente es la documentación en línea.

Empezar en 3 pasos

1. En la oficina de DeCaPortes, **Ajustes** → **API** → **Crear clave** (de pruebas). Se muestra una sola vez: guárdala.
2. Comprueba que funciona: `GET /api/v1/ping` con la cabecera `Authorization: Bearer dp_test_...`.
3. Crea tu primer DeCA de prueba con `POST /api/v1/decas` (ejemplo abajo). Cuando todo vaya bien, cambia a una clave real (`dp_live_`).

URL base: `https://decaportes.com/api/v1` · Solo HTTPS · JSON en UTF-8 (`Content-Type: application/json`) · Especificación: [openapi.json](#) (OpenAPI 3.1)

Autenticación

Cada petición lleva la clave de API en la cabecera:

```
Authorization: Bearer dp_live_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

- Cada clave pertenece a **una empresa** y solo ve y crea documentos de esa empresa.
- Una empresa puede tener varias claves (producción, pruebas, otro programa...) y **revocarlas** al momento desde su oficina.
- La clave **nunca** va en la URL ni en el JSON: si llega ahí, la petición se rechaza.
- DeCaPortes no guarda la clave, solo una huella (SHA-256): si se pierde, se crea otra.

Pruebas (sandbox)

Con una clave `dp_test_` todo va a una **empresa de pruebas** separada de la real:

- Los documentos salen marcados «**PRUEBAS API — SIN VALIDEZ**» y nunca se mezclan con los reales.
- Los vehículos y conductores que envíes se dan de alta solos (en real tienen que existir en DeCaPortes, porque cuentan en la licencia).
- La respuesta lleva `"test": true`.

Las claves de pruebas las puede crear cualquier cliente de DeCaPortes. Las reales necesitan el **módulo de API** (300 €/año + IVA, en cualquier plan): pídelo en info@decaportes.com o en el 614 096 823.

Crear un DeCA

`POST /api/v1/decas`

- **Viaje propio** (lo hace tu empresa): no envíes `carrier` (o envía tu propio NIF). Hacen falta `vehicle.tractor_plate` (y `trailer_plate` si es un conjunto articulado) y `driver.document_id`, de

vehículos y conductores dados de alta.

- **Viaje subcontratado:** envía en `carrier` la empresa que lo hace. La respuesta trae `signing_url`: su conductor firma la carga y la entrega con ese enlace (y pone su matrícula si no la enviaste).
- Si algún punto está fuera de España, el documento es un **eCMR** (si tu licencia lo incluye).
- Los clientes y los lugares se reutilizan si ya existen en tu cuenta (mismo NIF, misma dirección) o se dan de alta solos.

Petición

```
{
  "external_id": "ERP-PED-2026-000123",
  "reference": "Pedido 45872",
  "load_date": "2026-10-05",
  "source_system": "Mi ERP",
  "shipper": { "name": "Frutas del Sur S.L.", "tax_id": "B12345678", "address": "Pol. Ind. El Llano, nave 4", "postal_code": "30400", "city": "Caravaca de la Cruz", "country": "ES" },
  "consignee": { "name": "Distribuciones Centro S.A.", "tax_id": "A87654321" },
  "origin": { "name": "Almacén Frutas del Sur", "address": "Pol. Ind. El Llano, nave 4", "postal_code": "30400", "city": "Caravaca de la Cruz", "country": "ES" },
  "destination": { "name": "Mercamadrid", "address": "Ctra. de Villaverde a Vallecas, km 3,8", "postal_code": "28053", "city": "Madrid", "country": "ES" },
  "vehicle": { "tractor_plate": "1234ABC", "trailer_plate": "R1234BBB" },
  "driver": { "name": "Juan Pérez García", "document_id": "12345678Z" },
  "goods": { "description": "Fruta fresca paletizada", "weight_kg": 18500,
  "packages": 33, "packaging": "Palés" },
  "notes": "Descarga por la puerta 4"
}
```

Campo	Obligatorio	Notas
<code>load_date</code>	Sí	Fecha de carga, AAAA-MM-DD.
<code>shipper</code>	Sí (salvo empresas cargadoras)	Quien contrata el transporte: <code>name</code> , <code>tax_id</code> , <code>address</code> , <code>city</code> (y <code>postal_code</code> , <code>country</code>).
<code>consignee</code>	No	Destinatario. Si no viene, el <code>shipper</code> .
<code>carrier</code>	No	Transportista efectivo. Si no viene, tu empresa (viaje propio).
<code>origin</code> , <code>destination</code>	Sí	<code>address</code> y <code>city</code> obligatorios.
<code>goods.description</code>	Sí	Naturaleza de la mercancía.
<code>goods.weight_kg</code>	Sí*	*O <code>goods.other_measure</code> si el peso exacto no se sabe (p. ej. «30 m ³ »).
<code>vehicle</code> , <code>driver</code>	En viaje propio	<code>tractor_plate</code> , <code>trailer_plate</code> ; <code>document_id</code> (DNI/NIE) y <code>name</code> .
<code>external_id</code>	Recomendado	Tu identificador. No puede haber dos documentos vigentes con el mismo (sí uno nuevo si el anterior se anuló).
<code>reference</code> , <code>notes</code> , <code>source_system</code>	No	Referencia (pedido, albarán), observaciones y nombre de tu programa.

Respuesta (201)

```
{
  "success": true,
  "data": {
    "id": "6f1c2a4e-1b7d-4c55-9e0a-2f3b9d0c8a11",
    "external_id": "ERP-PED-2026-000123",
    "number": "DECA-2026-000123",
    "document_type": "deca",
    "status": "issued",
    "test": false,
    "pdf_url": "https://decaportes.com/api/v1/decas/6f1c.../pdf",
    "qr_url": "https://decaportes.com/api/v1/decas/6f1c.../qr",
    "public_url": "https://decaportes.com/d/AbC123...",
    "signing_url": null,
    "created_at": "2026-10-05T06:12:44.000Z",
    "updated_at": "2026-10-05T06:12:44.000Z",
    "...": "y los datos del documento (shipper, carrier, vehicle, goods...)"
  }
}
```

Guarda en tu programa el `id` y el `number`. El `public_url` es la página de verificación (la del QR): muestra el documento y su estado, sin datos personales innecesarios (el DNI del conductor sale oculto).

Endpoints

Método	Ruta	Para qué
GET	<code>/ping</code>	Comprobar la clave (empresa y modo).
POST	<code>/decas</code>	Crear un DeCA.
GET	<code>/decas</code>	Listar y buscar: <code>external_id</code> , <code>reference</code> , <code>status</code> , <code>from</code> , <code>to</code> (fecha de carga), <code>limit</code> (1–500, por defecto 50) y <code>cursor</code> .
GET	<code>/decas/{id}</code>	Consultar uno (estado, firmas, URLs).
POST	<code>/decas/{id}/cancel</code>	Anular con <code>{"reason": "..."} </code> . No se borra: queda ANULADO con el motivo.
GET	<code>/decas/{id}/pdf</code>	El PDF del documento (el mismo que en la web).
GET	<code>/decas/{id}/qr</code>	El QR de verificación en PNG, o SVG con <code>?format=svg</code> .

Paginación: la lista devuelve `pagination.next_cursor`; pásalo como `cursor` para la página siguiente mientras `has_more` sea `true`.

Estados

status	Significado
<code>issued</code>	Emitido, pendiente de la firma de la carga.
<code>in_transit</code>	Carga firmada: en transporte (<code>loaded_at</code>).
<code>delivered</code>	Entrega firmada: terminado (<code>delivered_at</code>).
<code>cancelled</code>	Anulado (<code>cancellation_reason</code>).

Las firmas de carga y entrega las hacen las personas en la app del conductor o en el enlace de firma; la API no puede firmar en su nombre.

Idempotencia

Envía en cada `POST` la cabecera `Idempotency-Key` con un identificador único (un UUID). Si la red falla y tu programa repite la petición con la **misma** clave, recibes la respuesta original (con la cabecera `Idempotent-Replayed: true`) y **no se crea un segundo DeCA**. Si reutilizas la clave con otro contenido, recibes `IDEMPOTENCY_KEY_REUSED`.

Errores

Siempre el mismo formato, con el `request_id` (también en la cabecera `X-Request-Id`) para que podamos localizar la llamada:

```
{
  "success": false,
  "error": { "code": "VALIDATION_ERROR", "message": "Hay datos que faltan o no son válidos." },
  "details": [ { "field": "goods.weight_kg", "message": "Obligatorio el peso..." } ],
  "request_id": "req_3kF9aQ2xLm0P"
}
```

HTTP	code	Cuándo
400	INVALID_REQUEST	JSON mal formado, parámetro no válido, clave en la URL.
401	UNAUTHORIZED	Sin clave, clave incorrecta o revocada.
403	FORBIDDEN	Módulo de API no activado, licencia caducada o cuenta suspendida.
404	NOT_FOUND	El documento no existe en tu empresa.
409	CONFLICT	Ya hay un documento vigente con ese <code>external_id</code> , o ya estaba anulado.
422	VALIDATION_ERROR	Faltan datos obligatorios del DeCA o no son válidos (ver <code>details</code>).
422	VEHICLE_NOT_REGISTERED · DRIVER_NOT_REGISTERED	La matrícula o el conductor no están dados de alta (solo en real).
422	IDEMPOTENCY_KEY_REUSED	Misma Idempotency-Key con otro contenido.
429	RATE_LIMITED	Demasiadas peticiones.
500 / 503	INTERNAL_ERROR / SERVICE_UNAVAILABLE	Error nuestro: reintentá con la misma Idempotency-Key.

Límites

120 peticiones por minuto y clave. Cada respuesta lleva `X-RateLimit-Limit`, `X-RateLimit-Remaining` y `X-RateLimit-Reset` (segundos hasta que se reinicia).

Ejemplos

cURL

```
curl -X POST https://decaportes.com/api/v1/decas \
-H "Authorization: Bearer $DECAPORTES_API_KEY" \
-H "Content-Type: application/json" \
-H "Idempotency-Key: $(uuidgen)" \
-d '{"external_id":"ERP-PED-2026-000123","load_date":"2026-10-05","shipper":
{"name":"Frutas del Sur S.L.","tax_id":"B12345678","address":"Pol. Ind. El Llano,
nave 4","city":"Caravaca de la Cruz"},"origin":{"address":"Pol. Ind. El Llano, nave
4","city":"Caravaca de la Cruz"},"destination":{"name":"Mercamadrid","address":"Ctra.
de Villaverde a Vallecas, km 3,8","city":"Madrid"},"vehicle":
{"tractor_plate":"1234ABC","trailer_plate":"R1234BBB"},"driver":{"name":"Juan Pérez
García","document_id":"12345678Z"},"goods":{"description":"Fruta fresca
paletizada","weight_kg":18500}}'
```

JavaScript

```
// Node.js 18+ (fetch incluido)
const res = await fetch('https://decaportes.com/api/v1/decas', {
  method: 'POST',
  headers: {
    Authorization: `Bearer ${process.env.DECAPORTES_API_KEY}`,
    'Content-Type': 'application/json',
    'Idempotency-Key': crypto.randomUUID(),
  },
  body: JSON.stringify({"external_id":"ERP-PED-2026-000123","load_date":"2026-10-
05","shipper":{"name":"Frutas del Sur S.L.","tax_id":"B12345678","address":"Pol. Ind.
El Llano, nave 4","city":"Caravaca de la Cruz"},"origin":{"address":"Pol. Ind. El
Llano, nave 4","city":"Caravaca de la Cruz"},"destination":
{"name":"Mercamadrid","address":"Ctra. de Villaverde a Vallecas, km
3,8","city":"Madrid"},"vehicle":
{"tractor_plate":"1234ABC","trailer_plate":"R1234BBB"},"driver":{"name":"Juan Pérez
García","document_id":"12345678Z"},"goods":{"description":"Fruta fresca
paletizada","weight_kg":18500}}}),
});
const r = await res.json();
if (!r.success) throw new Error(`${r.error.code}: ${r.error.message}
(${r.request_id})`);
console.log(r.data.number, r.data.public_url);
```

Python

```
import os, uuid, requests

r = requests.post(
    "https://decaportes.com/api/v1/decas",
    headers={
        "Authorization": f"Bearer {os.environ['DECAPORTES_API_KEY']}",
        "Idempotency-Key": str(uuid.uuid4()),
    },
    json={"external_id": "ERP-PED-2026-000123", "load_date": "2026-10-05", "shipper":
{"name": "Frutas del Sur S.L.", "tax_id": "B12345678", "address": "Pol. Ind. El Llano,
nave 4", "city": "Caravaca de la Cruz"}, "origin": {"address": "Pol. Ind. El Llano, nave
4", "city": "Caravaca de la Cruz"}, "destination": {"name": "Mercamadrid", "address": "Ctra.
de Villaverde a Vallecas, km 3,8", "city": "Madrid"}, "vehicle":
{"tractor_plate": "1234ABC", "trailer_plate": "R1234BBB"}, "driver": {"name": "Juan Pérez
García", "document_id": "12345678Z"}, "goods": {"description": "Fruta fresca
paletizada", "weight_kg": 18500}},
    timeout=30,
).json()
if not r["success"]:
    raise Exception(f"{r['error']['code']}: {r['error']['message']}
({r['request_id']})")
print(r["data"]["number"], r["data"]["public_url"])
```

```

<?php
$ch = curl_init('https://decaportes.com/api/v1/decas');
curl_setopt_array($ch, [
    CURLOPT_POST => true,
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_HTTPHEADER => [
        'Authorization: Bearer ' . getenv('DECAPORTES_API_KEY'),
        'Content-Type: application/json',
        'Idempotency-Key: ' . bin2hex(random_bytes(16)),
    ],
    CURLOPT_POSTFIELDS => '{"external_id":"ERP-PED-2026-000123","load_date":"2026-10-05","shipper":{"name":"Frutas del Sur S.L.","tax_id":"B12345678","address":"Pol. Ind. El Llano, nave 4","city":"Caravaca de la Cruz"},"origin":{"address":"Pol. Ind. El Llano, nave 4","city":"Caravaca de la Cruz"},"destination":{"name":"Mercamadrid","address":"Ctra. de Villaverde a Vallecas, km 3,8","city":"Madrid"},"vehicle":{"tractor_plate":"1234ABC","trailer_plate":"R1234BBB"},"driver":{"name":"Juan Pérez García","document_id":"12345678Z"},"goods":{"description":"Fruta fresca paletizada","weight_kg":18500}}}',
]);
$r = json_decode(curl_exec($ch), true);
if (!$r['success']) throw new Exception($r['error']['code'] . ': ' . $r['error']['message']);
echo $r['data']['number'], ' ', $r['data']['public_url'];

```

```
using System.Net.Http.Headers;
using System.Text;

var http = new HttpClient();
http.DefaultRequestHeaders.Authorization =
    new AuthenticationHeaderValue("Bearer",
    Environment.GetEnvironmentVariable("DECAPORTES_API_KEY"));
var req = new HttpRequestMessage(HttpMethod.Post,
    "https://decaportes.com/api/v1/decas") {
    Content = new StringContent("""{"external_id":"ERP-PED-2026-
000123","load_date":"2026-10-05","shipper":{"name":"Frutas del Sur
S.L.","tax_id":"B12345678","address":"Pol. Ind. El Llano, nave 4","city":"Caravaca de
la Cruz"},"origin":{"address":"Pol. Ind. El Llano, nave 4","city":"Caravaca de la
Cruz"},"destination":{"name":"Mercamadrid","address":"Ctra. de Villaverde a Vallecas,
km 3,8","city":"Madrid"},"vehicle":
{"tractor_plate":"1234ABC","trailer_plate":"R1234BBB"},"driver":{"name":"Juan Pérez
García","document_id":"12345678Z"},"goods":{"description":"Fruta fresca
paletizada","weight_kg":18500}}""", Encoding.UTF8, "application/json")
};
req.Headers.Add("Idempotency-Key", Guid.NewGuid().ToString());
var res = await http.SendAsync(req);
Console.WriteLine(await res.Content.ReadAsStringAsync());
```

Próximamente (v1.1): avisos automáticos a tu programa (webhooks) cuando se firma la carga o la entrega, cambios y sustitución de documentos, y alta de clientes, lugares, vehículos y conductores por la API.

Integración realizada por DeCaPortes

Si tu programa (ERP, TMS, WMS) ya dispone de su propia API, **nuestro equipo de integración se encarga de conectarlo con DeCaPortes**, sin desarrollo por tu parte: nos facilitas su documentación, la analizamos y te proponemos la solución y los plazos, sin compromiso. Solicítalo en info@decaportes.com.

¿Dudas con la integración? Escribenos a info@decaportes.com o llama al 614 096 823.