

Handleiding Mollie api

Een snelle handleiding



17 mei 2019

Digendo

Remco van der Linden

Versienummer: 1.0

Aanpas datum: 17 mei 2019

# Voorwoord

Dit document is een manier om snel het gebruik van de API code door te kunnen lezen, en gebruiken. Hierin word ook de installatie behandelt en het gebruik. Dit document is gemaakt door Remco van der Linden.

Inhoud

[*Voorwoord* 1](#_Toc8991174)

# Voorbereiding

Voordat je de applicatie installeert moet je rekening houden met de volgende stappen:

* Laravel is aangeraden voor gebruik tijdens het gebruik van deze handleiding.
* Er wordt ook rekening mee gehouden dat de applicatie op een lokale webserver getest gaat worden.
* Er wordt verwacht dat je al eens eerder met JSON arrays hebt gewerkt en deze kunt uitlezen
* Composer is verplicht om te installeren voordat je iets anders kunt doen
* Je hebt al een Laravel applicatie gemaakt en geïnstalleerd.
* Maak gebruik van de Visual studio code terminaal.

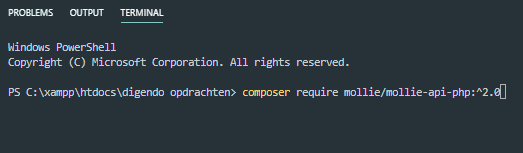
# Het installeren van de Mollie Api

Voor dit onderdeel is het belangrijk om eerst naar de website van Mollie te gaan ([https://docs.mollie.com](https://docs.mollie.com/)) hierin kunt u meer informatie terugvinden als dit niet duidelijk genoeg is, daarnaast kun u meer informatie terugvinden op GitHub (<https://github.com/mollie/mollie-api-php>).

Om de applicatie snel te installeren kopieert u deze code in de console:

**composer require mollie/mollie-api-php:^2.0**

Bijvoorbeeld:



Figuur 1 de zin gekopieerd in de terminal

## Account creëren

Het is ook belangrijk dat je een account hebt gemaakt op de website van sourcetree.

# Betalingen

$mollie = new \Mollie\Api\MollieApiClient();

$mollie->setApiKey("test\_dHar4XY7LxsDOtmnkVtjNVWXLSlXsM");

## IDeal betalingen

De methode die nodig zal zijn voor het opzetten van een betaling kan makkelijk gedaan worden via dit:

$method = $mollie->methods->get(\Mollie\Api\Types\PaymentMethod::IDEAL, ["include" => "issuers"]);

Een array voor een betaling kan er als volgt uitzien:

$payment = $mollie->payments->create([

"amount" => [

"currency" => "EUR",

"value" => "10.00"

],

"description" => "My first API payment",

"redirectUrl" => "https://webshop.example.org/order/12345/",

"webhookUrl" => "https://webshop.example.org/mollie-webhook/",

"method" => \Mollie\Api\Types\PaymentMethod::IDEAL,

"issuer" => $selectedIssuerId, // e.g. "ideal\_INGBNL2A"

]);

## Terugkerende data opslaan

Terugkerende data ziet er als volgt uit:

{

"totalCount": 280,

"offset": 0,

"count": 10,

"data": [

{

"resource": "payment",

"id": "tr\_7UhSN1zuXS",

"mode": "test",

"createdDatetime": "2018-03-16T17:08:53.0Z",

"status": "open",

"expiryPeriod": "PT15M",

"amount": "10.00",

"description": "My first payment",

"metadata": {

"order\_id": "12345"

},

"locale": "nl",

"profileId": "pfl\_QkEhN94Ba",

"links": {

"redirectUrl": "https://webshop.example.org/order/12345/"

}

},

{ },

{ }

],

"links": {

"first": "https://api.mollie.com/v1/payments?count=10&offset=0",

"previous": null,

"next": "https://api.mollie.com/v1/payments?count=10&offset=10",

"last": "https://api.mollie.com/v1/payments?count=10&offset=270"

}

}

Deze data is dan makkelijk om weer op te slaan, dit kan natuurlijk met een Laravel controller via een database query, de code die je bijv. kunt gebruiken bij het opslaan van deze data kan als volgt met:

$payment = $mollie->payments->get($request->id);

*//$save\_payment = payment::where('payment\_id', $request->id)->firstOrFail();*

$save\_payment->mode = $payment->mode;

$save\_payment->amount = $payment->amount->value;

$save\_payment->currency = $payment->amount->currency;

$save\_payment->description = $payment->description;

$save\_payment->method = $payment->method;

$save\_payment->status = $payment->status;

$save\_payment->createat = $payment->createdAt;

$save\_payment->paidat = $payment->paidAt;

$save\_payment->canceledat = $payment->canceledAt;

$save\_payment->expiresat = $payment->expiresAt;

$save\_payment->failedat = $payment->failedAt;

$save\_payment->profile\_id = $payment->profileId;

$save\_payment->consumername = $payment->details->consumerName;

$save\_payment->consumeraccount = $payment->details->consumerAccount;

$save\_payment->consumerbic = $payment->details->consumerBic;

$save\_payment->locale = $payment->locale;

$save\_payment->countrycode = $payment->countryCode;

$save\_payment->save();

# Meer informatie

Voor meer informatie is het aangeraden om de handleidingen door te nemen die gemaakt zijn door Laravel en Mollie om door te hebben hoe je dit allemaal kunt gebruiken. Kijk daarvoor op de volgende websites

<https://laravel.com/docs/5.6> (Laravel)

[https://docs.mollie.com](https://docs.mollie.com/) (Mollie)

# Kwaliteitscontrole en handboek

Het is belangrijk dat er een controle wordt gedaan over welke informatie er binnenkomen en welke informatie daar nodig van is:

Van de volgende code die aangegeven is zal er een voorbeeld aangegeven worden wat er precies nodig is.

{

"totalCount": 280,

"offset": 0,

"count": 10,

"data": [

{

"resource": "payment",

"id": "tr\_7UhSN1zuXS",

"mode": "test",

"createdDatetime": "2018-03-16T17:08:53.0Z",

"status": "open",

"expiryPeriod": "PT15M",

"amount": "10.00",

"description": "My first payment",

"metadata": {

"order\_id": "12345"

},

"locale": "nl",

"profileId": "pfl\_QkEhN94Ba",

"links": {

"redirectUrl": "https://webshop.example.org/order/12345/"

}

},

{ },

{ }

],

"links": {

"first": "https://api.mollie.com/v1/payments?count=10&offset=0",

"previous": null,

"next": "https://api.mollie.com/v1/payments?count=10&offset=10",

"last": "https://api.mollie.com/v1/payments?count=10&offset=270"

}

}

In de code die hiervan teruggestuurd was, is dit het enigste wat echt belangrijk zal zijn voor dit onderdeel:

"resource": "payment",

"id": "tr\_7UhSN1zuXS",

"mode": "test",

"createdDatetime": "2018-03-16T17:08:53.0Z",

"status": "open",

"expiryPeriod": "PT15M",

"amount": "10.00",

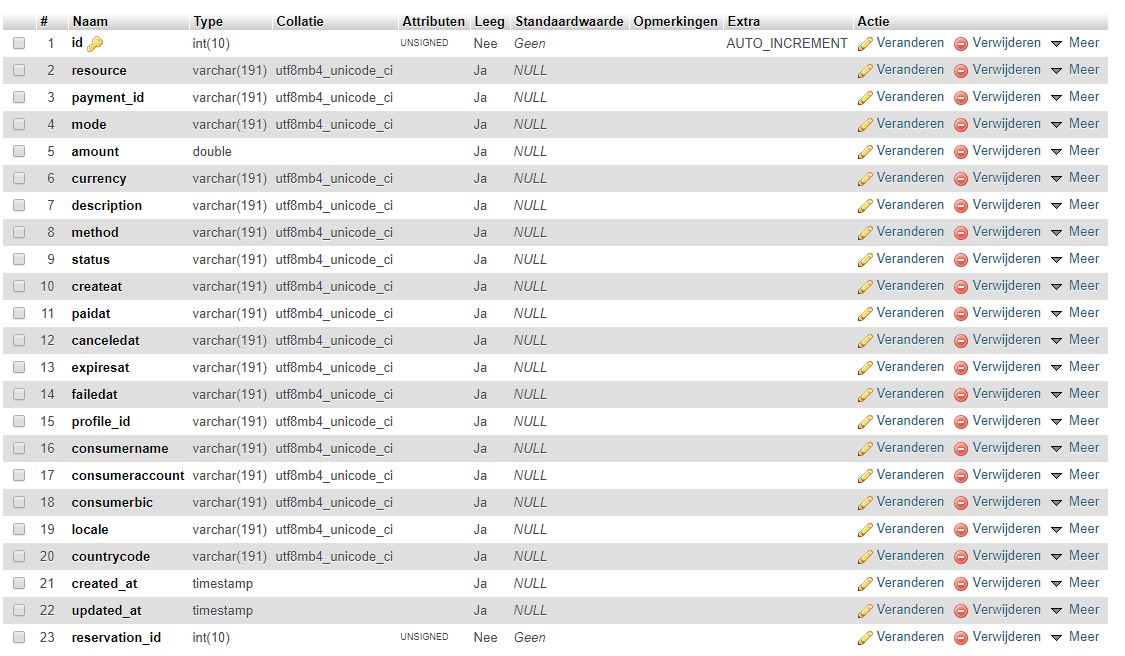
"metadata": {

"order\_id": "12345"

},

Met de code die hier instaat kun je alle data opvragen en aanpassen als dit nodig is.

Voor de database zal er de data opgeslagen worden in de payment, reservation en user tabel opgeslagen worden.



Figuur 2: resultaat van de payment table

Informatie over het product zelf is terug te vinden op papier en is gearchiveerd.