

# Mollie Betaalintegratie Implementatie Instructies

Deze instructies beschrijven hoe je de Mollie betaalfunctionaliteit kunt integreren in je React boekenkastconfigurator op likeabilly.nl.

## Inhoudsopgave

1. [Overzicht van het probleem](#)
2. [Vereisten](#)
3. [Implementatiestappen](#)
4. [Bestandsstructuur](#)
5. [Backend implementatie](#)
6. [Testen](#)
7. [Veelvoorkomende problemen](#)

## Overzicht van het probleem

De huidige implementatie van de Mollie betaalfunctionaliteit werkt niet correct. Hoewel de Mollie script (<https://js.mollie.com/v2/mollie.js>) wordt geladen, wordt het Mollie object niet correct geïnitieerd en verschijnen er geen betaalopties na stap 5 van de boekenkastconfigurator.

## Vereisten

- Een Mollie account met API-sleutels
- Het profiel ID van je Mollie account (begint met `pfl_`)
- React applicatie gebouwd met Vite
- Node.js backend voor het verwerken van betalingen

## Implementatiestappen

### 1. Installeer de Mollie API client

```
npm install @mollie/api-client
```

## 2. Voeg de React componenten toe

Kopieer de volgende bestanden naar je project: - `MolliePayment.jsx` - De hoofdcomponent voor de betaalinterface - `MolliePayment.css` - Styling voor de betaalinterface - `MollieIntegration.js` - Frontend API integratiefuncties - `PaymentPage.jsx` - Wrapper component voor de betaalpagina

## 3. Integreer de betaalpagina in je applicatie

Voeg de `PaymentPage` component toe aan je applicatie na stap 5 van de configurator. Bijvoorbeeld:

```
import React, { useState } from 'react';
import PaymentPage from './components/PaymentPage';

function App() {
  const [currentStep, setCurrentStep] = useState(1);
  const [order, setOrder] = useState({
    height: 202,
    width: 80,
    depth: 40,
    material: 'Spaanplaat',
    shelves: 5,
    price: 149.00
  });
  const [customerInfo, setCustomerInfo] = useState({
    name: "",
    email: "",
    address: "",
    postalCode: "",
    city: "",
    phone: ""
  });

  // Functie om naar de volgende stap te gaan
  const nextStep = () => {
    setCurrentStep(currentStep + 1);
  };

  // Functie om terug te gaan naar de vorige stap
  const prevStep = () => {
    setCurrentStep(currentStep - 1);
  };

  // Render de juiste component op basis van de huidige stap
  const renderStep = () => {
    switch (currentStep) {
      case 1:
        return <Afmetingen order={order} setOrder={setOrder}>
```

```

nextStep={nextStep} />;
  case 2:
    return <Planken order={order} setOrder={setOrder} nextStep={nextStep}
prevStep={prevStep} />;
  case 3:
    return <Materiaal order={order} setOrder={setOrder} nextStep={nextStep}
prevStep={prevStep} />;
  case 4:
    return <Overzicht order={order} nextStep={nextStep} prevStep={prevStep} />;
  case 5:
    return <Gegevens customerInfo={customerInfo}
setCustomerInfo={setCustomerInfo} nextStep={nextStep} prevStep={prevStep} />;
  case 6:
    return <PaymentPage order={order} customerInfo={customerInfo}
onBack={prevStep} />;
  default:
    return <Afmetingen order={order} setOrder={setOrder}
nextStep={nextStep} />;
  }
};

return (
  <div className="app-container">
    {renderStep()}
  </div>
);
}

export default App;

```

## 4. Configureer de Mollie profiel ID

Vervang in `PaymentPage.jsx` de placeholder `pfl_YOUR_PROFILE_ID` met je echte Mollie profiel ID:

```

<MolliePayment
  order={completeOrder}
  onPaymentComplete={handlePaymentComplete}
  onPaymentError={handlePaymentError}
  profileId="pfl_YOUR_PROFILE_ID" // Vervang dit met je echte Mollie profiel ID
/>

```

## 5. Implementeer de backend API endpoints

Implementeer de volgende API endpoints in je backend:

- `POST /api/payments/create` - Voor het aanmaken van een betaling

- GET /api/payments/methods - Voor het ophalen van beschikbare betaalmethoden
- GET /api/payments/status/:id - Voor het controleren van de betaalstatus
- POST /api/webhooks/mollie - Voor het ontvangen van Mollie webhooks

Zie het bestand `backend-example.js` voor een voorbeeld implementatie.

## Bestandsstructuur

```
src/
├── components/
│   ├── MolliePayment.jsx    # Hoofdcomponent voor de betaalinterface
│   ├── MolliePayment.css    # Styling voor de betaalinterface
│   ├── PaymentPage.jsx      # Wrapper component voor de betaalpagina
│   └── ...
├── utils/
│   ├── MollieIntegration.js  # Frontend API integratiefuncties
│   └── ...
└── ...
```

## Backend implementatie

Voor de backend implementatie heb je de Mollie API client nodig. Hier is een voorbeeld van hoe je de betalingen kunt verwerken:

```
const express = require('express');
const { createMollieClient } = require('@mollie/api-client');
const router = express.Router();

// Initialiseer Mollie client met je API key
const mollieClient = createMollieClient({
  apiKey:
    process.env.MOLLIE_API_KEY // Gebruik een omgevingsvariabele voor de API key
});

// API endpoint voor het aanmaken van een betaling
router.post('/api/payments/create', async (req, res) => {
  try {
    const {
      amount,
      description,
      redirectUrl,
      webhookUrl,
      metadata,
      method,
      issuer,
    }
```

```

    cardToken
  } = req.body;

  // Valideer verplichte velden
  if (!amount || !description || !redirectUrl) {
    return res.status(400).json({
      success: false,
      message: 'Verplichte velden ontbreken'
    });
  }

  // Bouw betaalverzoek
  const paymentRequest = {
    amount: amount,
    description: description,
    redirectUrl: redirectUrl,
    webhookUrl: webhookUrl || `${process.env.API_URL}/api/webhooks/mollie`,
    metadata: metadata || {},
    method: method
  };

  // Voeg issuer toe voor iDEAL betalingen
  if (method === 'ideal' && issuer) {
    paymentRequest.issuer = issuer;
  }

  // Voeg cardToken toe voor creditcard betalingen
  if (method === 'creditcard' && cardToken) {
    paymentRequest.cardToken = cardToken;
  }

  // Maak betaling aan via Mollie API
  const payment = await mollieClient.payments.create(paymentRequest);

  // Stuur betaalgegevens terug naar client
  return res.status(200).json({
    success: true,
    id: payment.id,
    status: payment.status,
    _links: payment._links
  });
} catch (error) {
  console.error('Error creating payment:', error);
  return res.status(500).json({
    success: false,
    message: 'Er is een fout opgetreden bij het aanmaken van de betaling',
    error: error.message
  });
}
});

// Implementeer de andere endpoints zoals beschreven in backend-example.js

```

# Testen

Om de implementatie te testen:

1. Zorg ervoor dat alle componenten correct zijn geïmplementeerd
2. Configureer je Mollie API sleutels en profiel ID
3. Start je applicatie en doorloop de boekenkastconfigurator
4. Na stap 5 zou je de betaalpagina moeten zien met de betaalopties
5. Test de verschillende betaalmethoden (gebruik testmodus in ontwikkelomgeving)

## Veelvoorkomende problemen

### Mollie object wordt niet geïnitieerd

Controleer of: - De Mollie.js script correct wordt geladen - Je het juiste profiel ID gebruikt (begint met `pfl_`, niet je API sleutel) - De script wordt geladen voordat je het Mollie object initialiseert

### Betaalopties verschijnen niet

Controleer of: - De MolliePayment component correct wordt gerenderd - Er geen JavaScript fouten zijn in de console - De betaalmethoden correct worden opgehaald van de backend

### Betaling kan niet worden aangemaakt

Controleer of: - Je de juiste API sleutel gebruikt in je backend - Alle verplichte velden worden meegegeven in het betaalverzoek - De webhook URL bereikbaar is voor Mollie

### Betaling wordt niet verwerkt na 3D Secure authenticatie

Controleer of: - De redirectUrl correct is geconfigureerd - Je de betaalstatus correct controleert na terugkeer van de Mollie betaalpagina