



Adding embedded frames to a website

An integration guide for developers

Version 2.1 – December 2022

**Table of contents**

About this guide	3
For more information	3
Introduction	4
Key features	4
The benefits of using embedded frames	4
How it works	5
What you need	5
What you need to do	5
Integrating Embedded Frames	6
Creating a payment intent	6
Adding the backend script	8
Downloading the JavaScript library and creating a Cashflows object	9
Creating the payment fields and payment form	9
Initializing the Cashflows object	9
Apple Pay Domain Association File	11
Testing and going live	11
Further Information	11

Copyright

2022 © Cashflows or Cashflows Europe Limited

While every effort has been made to ensure the accuracy of the information contained in this publication, the information is supplied without representation or warranty of any kind, is subject to change without notice and does not represent a commitment on the part of Cashflows Europe Limited. Cashflows Europe Limited, therefore, assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from this material or any part thereof, or any supplementary materials subsequently issued by Cashflows Europe Limited. Cashflows Europe Limited has made every effort to ensure the accuracy of this material.



About this guide

Welcome to this guide about integrating the Cashflows Embedded Frames payment fields. This is an integration guide for website developers. This guide assumes a high level of knowledge of client- and server-side programming.

This guide is for you if you are a developer working on a website for an online business. It may also be useful if you are a business owner who:

- Uses Cashflows services for processing online card payments.
- Wants to add a page to your website that looks and feels just like your other webpages but is hosted by us.

Note: This guide assumes that you have read [Getting Started with the Cashflows Gateway API](#). This guide is available from the [Cashflows Help Centre](#).

Important: The capitalisation of API calls is important. Always use the same capitalisation as the examples that we provide.

For more information

The Cashflows Help Centre

Our [Cashflows Help Centre](#) has all of our published guides plus answers to questions about Cashflows products. If you can't find the answer to your question, you can request a new article via your relationship manager or contact one of our support teams.

You can also find explanations of the responses from our payment processing (acquiring) platform. These codes can help you to understand why and how a payment may not have been successfully processed.

Implementation support

For implementation support before your integration goes live:
Email: implementations@cashflows.com

Technical support

If your account is live with us and you need technical help:
Email: techsupport@cashflows.com



Introduction

To provide a seamless customer experience, we offer the option to add Embedded Fields that blend into your website and checkout pages but crucially the payment fields themselves are still hosted by us. We call these Embedded Fields.

Embedded Fields are a set of iFrames that we provide for you to incorporate into a website. By connecting our Embedded Fields to your website, you are integrating with the Cashflows Gateway. With Embedded Fields, when your customers arrive at the final checkout page of your website, a series of secure payment fields is displayed. Customers enter their card details into these fields and select the payment button.

By using Embedded Frames, you can display a secure payment page or the payment fields from a page within your business website. To the customer, it gives the impression that the business is capturing their card data. In fact we take care of the processing, including security aspects and many of the technical complexities.

Key features

Embedded Frames offers:

- The potential to accept all major types of card payment including Visa, Mastercard and Amex
- Handling of zero value authorisations and verifications
- The latest version of 3D Secure services from Visa and Mastercard are built in
- A downloadable industry-standard JavaScript library for easy integration and instant updates
- Pages that use embedded frames are responsive and compatible with mobile devices
- Full customisation to match the look and feel of any website

The benefits of using embedded frames

- Fewer abandoned payments: embedded frames can be customised to look and feel the same as any business website. This can encourage customer confidence all the way through the payment process and reduce the number of abandoned payments
- Speed: The longer a transaction takes and the more steps a customer needs to complete, the higher the chance of a customer giving up and abandoning a payment. With embedded frames, the customer journey is quick
- Control: complete control over the checkout and payment page user interface and user experience without any worries over handling or exposing card details
- PCI compliance: customers can safely enter their card details and pay with confidence

How it works

1. You embed each payment field within the checkout page as an iFrame.
2. Each payment field that you place within the checkout page is injected with an iFrame by Cashflows
3. The customer then enters their payment details directly on their checkout page.
4. After each field is completed, the data is sent directly to the Cashflows Gateway API where it is held securely in memory.
5. After all the required fields are completed and the customer presses the pay or verify button, the Cashflows Gateway concatenates the supplied payment data into a payment job request.
6. The Cashflows Gateway sends this request either for authorisation or to the issuer's bank to request the extra security checks in the form of Storing Customer Authentication (SCA)
7. A JavaScript method is called and can be used to either:
 - a. Display a success or failure message or
 - b. Redirect the customer back to a website or web page that you specify

What you need

Before you start make sure you have:

- Collected your API key from Cashflows Go. For information, refer to [Getting started with the Cashflows Gateway API \(PDF\)](#) available from the [Cashflows Help Centre](#)
- Collected your Configuration ID from Cashflows Go. For information, refer to [Getting started with the Cashflows Gateway API \(PDF\)](#) available from the [Cashflows Help Centre](#)
- Downloaded the JavaScript Library. For information, see [Downloading the JavaScript library and creating a Cashflows object](#)

What you need to do

So that we can process an online payment via the Embedded Fields, your backend server needs to send an API request to our online payment servers to initiate your intent to create a transaction.

We then send you a Payment Job Reference and the IntentToken which are linked to each other. You need to create the payment fields and the payment form and then link these to the Cashflows JavaScript Library in order to create and complete the payment request using this token.

To enable you to test your embedded frames before going live, we have an integration environment where you can simulate different payment scenarios. For information see [Testing and going live](#).

If you need an integration account, please email: implementations@cashflows.com.

Tip: We recommend that you use UTF-8 for encoding before you submit.

You must Hash the message with the API key that you collected from Cashflows Go.

You must use the same hashing technique that we do so that we can match our hash with yours – we use SHA512. The hash assures message consistency and protects the payment request from being tampered with during transfer to our Cashflows gateway (a man-in-the-middle attack). Your message needs to have the following headers:

Important: Every parameter and value that you send, and every parenthesis, must be included in the hash.



Integrating Embedded Frames

The process for integrating embedded frames involves the following steps:

1. Create a Payment Intent. For information, see [Creating a Payment Intent](#).
2. Add the backend script. For information, see [Adding the backend script](#).
3. Download the JS Library and create a Cashflows object. For information, see [Downloading the JS Library and Create a Cashflows Object](#).
4. Create the payment fields and the payment form. For information, see [Creating the payment fields and payment form](#).
5. Initialize the Cashflows object. For information, see [Initializing the Cashflows object](#).

Creating a payment intent

You first need to create a payment intent using an API call to Cashflows. For this, you need the API credentials obtained from the portal (your API key and Configuration ID) as well some additional parameters and values as described below:

Required fields

Field	Description
amountToCollect	The total value of the payment transaction.
currency	The transaction currency.
paymentMethodsToUse	The payment type for the transaction.

You can also include address parameters. For information, see [Address and shipping data](#).

Example: Create new payment intent:

- Integration test environment:
POST <https://gateway-int.cashflows.com/api/gateway/payment-intents/>
- Production environment
POST <https://gateway.cashflows.com/api/gateway/payment-intents/>

Request

```
{ "amountToCollect": "10.00", "currency": "GBP", "locale": "en_GB",  
  "PaymentMethodsToUse": ["Card"] }
```

Response

```
{ "data": { "PaymentJobReference": "Examplepaymentjobreference", "Token":  
  "yourpaymentintenttoken", "paymentStatus": "Pending" } }
```

Address and shipping data

We recommend that you supply as much cardholder data as possible to us as this will help with the authentication and authorisation process and reduce the number of declined transactions or related issues.

Here is some example JSON code that shows how the billing and shipping address data should be structured when passing this information to the Cashflows Gateway. You should add this data to the parameters that you are hashing and sending to the Cashflows Gateway when you create the Payment Intent.

```
"BillingAddress": {  
  "Title": "Mr.",  
  "FirstName": "Virgil",  
  "MiddleName": "Van",  
  "LastName": "Dijk",  
  "CountryIso3166Alpha2": "826",  
  "AddressLine1": "Business Centre Road",  
  "AddressLine2": "Unit 7",  
  "ZipCode": "XX12 1XX",  
  "City": "TestCity",  
  "StateProvince": null,  
  "PhoneNumber1": "+44 9999 123456",  
},  
"ShippingAddress": {  
  "Title": "Mr.",  
  "FirstName": "John",
```

```
"MiddleName": null,  
"LastName": "Doe",  
"CountryIso3166Alpha2": 826,  
"AddressLine1": "Addressline1",  
"AddressLine2": "Addressline2",  
"ZipCode": "XX21 5XX",  
"City": "Cambridge",  
"StateProvince": null,  
"PhoneNumber1": "+44 9999 000000",
```

Adding the backend script

Include the backend script that creates and handles the Payment Intent created earlier (for information see [Creating a payment intent](#)) at the top of the checkout page within the header of the page. Here's an example to show how to create a HTML page using PHP:

```
<?php  
    include 'create_payment_intent_cashflows.php';  
?>  
  
<!DOCTYPE html>  
  
<html>  
  
html page....  
  
body contents...
```

Example backend code (PHP)

You can find a PHP code example for the backend if you visit github:

<https://github.com/Cashflows/cashflows-clientlib-js/tree/main/examples>

Downloading the JavaScript library and creating a Cashflows object

1. Get and download the relevant JavaScript (JS) library from [Git Hub](#).
2. Add the JS library to the Checkout Page where your payment fields will be rendered. Refer to it with the following HTML:

```
<script src = "JS library link path"></script>
```

This loads in the JavaScript library that creates the Cashflows object that will be used in the steps that follow.

Creating the payment fields and payment form

Create a form with the payment fields and payment button. You can use any names for the payment fields but we suggest:

- card-number
- card-name
- card-expiration
- card-cvc

Important: Always use the same capitalisation as the examples that we provide.

Example:

```
<div class = "form-group">  
  <label class = "form label" for "card-number"> Card Number </label>  
  <input id = "card-number" type = "tel" class = "form control  
  required"/>  
</div>
```

You can use the div class and input IDs to style the payment fields within the payment form using styles from a cascading style sheet (CSS).

Initializing the Cashflows object

At the bottom of the checkout page, You need to initialise the Cashflows object with the token acquired earlier when you created the Payment Intent. For information, see [Creating a payment intent](#). This instructs our library use the payment fields that you have created. For information, see [Creating the payment fields and payment form](#).

Example:

```
<script>  
  window.onload = function() {  
    var cashflows = new Cashflows('yourpaymentintentsoken', true);
```

```
var inits = [  
    cashflows.initCard('#card-number', '#card-name', '#card-  
expiration', '#card-cvc', '#pay-with-card'),  
    cashflows.initApplePay('#apple-pay'),  
    cashflows.initGooglePay('#google-pay')  
];  
  
Promise.all(inits)  
  
.then(() => {  
    cashflows.checkout()  
  
    .then(() => {  
        var el = document.getElementById('card');  
        el.style.display = "none";  
        el = document.getElementById('alert-success');  
        el.style.display = "block";  
    })  
  
    .catch(e => {  
        var el = document.getElementById('card');  
        el.style.display = "none";  
        el = document.getElementById('alert-error');  
        el.style.display = "block";  
        el.querySelector('#message').innerHTML = e;  
    });  
})  
  
.catch(e => {  
    var el = document.getElementById('card');  
    el.style.display = "none";  
    el = document.getElementById('alert-error');  
    el.style.display = "block";  
    el.querySelector('#message').innerHTML = e;  
});
```

```
        });  
    }  
</script>
```



Apple Pay Domain Association File

To acquire the domain association file, you will first need to add your domain via the Apple Pay Domains tab within Cashflows Go, to do this please see our Cashflows Go guide – which can be found [here](#).

Once you have successfully registered your domain with Apple you will need to contact our implementation team at implementations@cashflows.com

Once you have the domain association file you will need to host it with the name [https://\[DOMAIN_NAME\]/.well-known/apple-developer-merchantid-domain-association](https://[DOMAIN_NAME]/.well-known/apple-developer-merchantid-domain-association) with “domain_name” being your domain name.



Testing and going live

To enable you to test your embedded frames before going live, we have an integration environment where you can simulate different payment scenarios:

To send a request to our Integration environment use this url: <https://gateway-int.cashflows.com>

To send a request to our Production environment use this url: <https://gateway.cashflows.com>

If you are testing card payments, you need to use a valid card number. We provide some test cards that you can use. Visit the Cashflows Help Centre and search for *test* or *test card*.

For information about testing your integration and preparing to go live, refer to [Getting started with the Cashflows Gateway API](#) (PDF) in our Cashflows Help Centre.



Further Information

For more information, refer to our integration guide, [Adding a hosted payment page to a website](#) (PDF), which is available from the Cashflows Help Centre. It describes how to:

- Handle payment status updates
- Manage capture requests for card payments
- Refund a payment
- Switch on Address Verification checking.

Cashflows



+44 (0) 330 128 9855

Cambridge
CPC1
Capital Park
Cambridge
CB21 5XE

London
20 Farringdon St
London
EC4A 4AB

The Netherlands
Noorderhof 24
5804 BV Venray