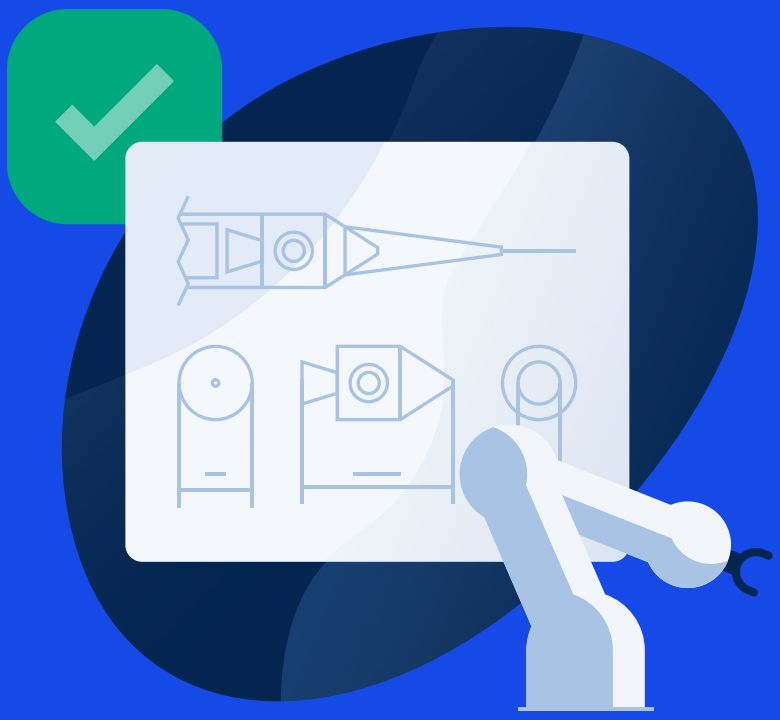
GOCARDLESS

A guide to technical certification

GoCardless Partners



Introduction

With the GoCardless API, it's easy to build a partner integration allowing your users to collect and manage their payments in a way that's seamlessly integrated with your product.

Before setting your integration live, it must be technically certified by a member of the GoCardless team. Technical certification ensures that your partner integration supports all the features and functionality necessary for providing your users with a best in class experience.

Once your partner integration has been technically certified and has gone live, it will be listed and promoted on our partner directory.

This guide lists the requirements under each section, why it is important that your integration satisfies them and provides links to more detailed information and guidance.

What are the requirements?

We've broken down our certification requirements into 10 easy to follow sections:

How to complete the process

Before getting started, we recommend reading through this guide to fully understand our requirements before testing your partner integration in our Sandbox environment.

Once you've completed testing, you can create a live GoCardless account and partner app. At this point you'll be prompted to submit your details and join our Partner Programme, gaining access to our Partner Portal.

You can complete the certification process within the Portal, submitting your answers and comments against each requirement before submitting for review. Once you've submitted your answers one of our Solutions Engineers will be in touch to provide feedback and information on next steps. You can read more information about this here.

Other useful information

Partner developer documentation

GoCardless API reference

GoCardless Partner Portal: Getting started guide

Section 1

Creating the GoCardless account and partner app

In order to create a Partner App, you need to create a GoCardless account. A payout bank must be added to this account during the verification process to ensure revenue share or app fee payouts can be paid.

1.1 Have you created a live GoCardless account?

You need to create a live GoCardless account in order to create your Partner app and register relevant payout bank account details to receive revenue share or app fee payouts.

Note - during the Onboarding flow you should select the 'Standard' package.

1.2 Have you given your Partner app a sensible public facing name and description?

Your users will see the app name and description when connecting their GoCardless accounts to your Partner app.

1.3 Have you uploaded and added your company logo to your Partner app within the GoCardless dashboard?

Your users will see this logo when connecting their GoCardless accounts to your Partner App

1.4 Have you added relevant contact details to your GoCardless account?

We need these details in case we need to contact you. You can update these details by navigating to 'Settings', 'Company Info', 'Edit Account Details', '3. Contact details'.

1.5 Have you completed the GoCardless account verification process?

You need to complete this process, including adding relevant payout bank account details, in order to receive revenue share or app fee payouts. You can find more information here.

Note - if your users are collecting payments in more than one scheme, you need to add a payout bank account per scheme. More information here.

Connecting your users via OAuth

Your integration needs to support the functionality required to securely gain access to your users' GoCardless accounts. This is done via our OAuth flow. Note we will not support non-OAuth integration partners.

2.1 **Are you using the GoCardless OAuth flow to obtain your users' access tokens and to access your users' accounts?**

OAuth is secure, fast, future-proof and enables GoCardless to payout revenue share or app fees.

Note - we do not support non-OAuth integration partners.

2.2 **Are you passing the 'initial_view' parameter when creating the OAuth link to ensure your users see the signup screen?**

This provides a best in class experience for your users when connecting to your Partner app.

2.3 **Are you pre-filling as much information on the GoCardless signup screen as possible (e.g. the email address, given and family names, company name)?**

This provides a best in class experience for your users when connecting to your Partner app.

2.4 **Are you ensuring that the process of exchanging the client secret for the permanent access token is taking place on your server(s) and avoids passing the client secret to your users' device?**

This will prevent anybody impersonating your Partner app and gaining unauthorised access to your users' accounts.

2.5 Are you storing the access token in your database in a safe and secure manner?

Storing the access token will allow you to make requests to the GoCardless API on your users' behalf. Note access tokens are analogous to passwords and need to be treated with similar care and must be encrypted.

2.6 Are you storing the GoCardless Organisation ID for each user that completes the OAuth flow?

Storing the Organisation ID will allow you to interpret webhook events.

Section 3

Helping your users to get verified

Before any of your users can receive payouts of funds they've collected from their customers via GoCardless, they need to verify their account. We collect the necessary details for this through our dedicated Onboarding flow in order to comply with anti-money laundering regulations.

Once your users have created their Gocardless account through your integration, they should be sent straight to the Onboarding flow to begin the verification process.

3.1 **Once a user has completed the OAuth flow, are you checking the user's verification status via the Creditors API?**

You can check a user's verification status (e.g. action required) through the 'verification_status' endpoint and take action accordingly (e.g. immediately send them to the Onboarding flow). Your users must have a verified account to receive payouts.

3.2 **If a user has just created a GoCardless account and is in the 'action_required' state, are you sending them to the GoCardless Onboarding flow immediately?**

Sending a user directly to the Onboarding flow once they have connected their account will allow them to begin the verification process and improve user activation rate.

3.3 **Are you displaying a user's verification status within your platform and providing an appropriate link to the GoCardless Onboarding flow?**

This provides a best in class experience for your users and will improve activation rate. It also reduces the need for your users to login to the GoCardless dashboard.

3.4 **If a user's verification status changes to 'action_required' at any other point, are you highlighting this to the user and providing them with a way to get back to the GoCardless Onboarding flow?**

A user's verification status can change even after a 'successful' event. Note, GoCardless does not generate a webhook event when a user's verification status changes - you need to check this via the creditor endpoint on a scheduled basis.

3.5 **Have you specified an appropriate 'Post-onboarding URL' that users will be redirected to once the Onboarding flow has been completed?**

Specifying an appropriate post-onboarding URL ensures that users are redirected back to your software once they have completed the GoCardless onboarding flow. Note, this must be a https URL that you trust.

Section 4

Mandates - Set up

You need to provide a way for your users to set up Direct Debit mandates with their end customers (payers). A mandate allows a user to pull money from an end customer's bank account with a simple API call.

4.1 **Are you providing your users with the ability to send end customers (payers) to the Direct Debit setup flow?**

You need to provide your users with a way to create customers and mandates within your platform rather than using the GoCardless dashboard. Typical methods include enabling users to;

- Email a link to end customers (payers)
- Generate a link to be used elsewhere (e.g. embed in website or personal email)
- Add a link to an invoice

4.2 **If mandate links can be sent by email, are your users able to send mandate requests to multiple end customers (payers) in batches?**

This provides a best in class experience for your users and avoids the need to send each mandate request link individually.

4.3 **If mandate links can be sent by email, are your users able to send reminders to end customers (payers) to complete the mandate setup?**

This provides a best in class experience for your users. When a user sends a generic link and a customer sets up a mandate, it should be tied to the existing customer record in your platform.

4.4 **If mandate links can be sent by email, and the email address and/or customer record does not already exist in your platform, are you providing mandate matching functionality?**

This provides a best in class experience for your users. When a user sends a generic link and a customer sets up a mandate, it should be tied to the existing customer record in your platform.

4.5 **Once the re-direct flow has been completed, are you storing the mandate ID in your database against the end customer (payer) record?**

This allows you to easily keep track of mandates over their whole lifecycle (ensuring the same mandate is used for future payments) and handle multiple mandates per individual end customer (payer).

Section 5

Mandates - Payment pages

There are several options for setting up Direct Debit mandates. You can choose to use the GoCardless secure, hosted payment pages (see examples here) or instead to build your own custom payment pages. In addition, individual users may choose to build their own payment pages (or request this) which you must be able to support.

Sub-section 1 - GoCardless hosted payment pages

The following requirements are only applicable if you are using GoCardless hosted payment pages

5.1.1 Are you using GoCardless hosted payment pages?

You have the option of using GoCardless hosted payment pages or building your own custom payment pages (see next section).

5.1.2 Are you pre-filling the end customer (payer) details on the GoCardless payment page?

This provides a best in class experience for end customers (payers).

5.1.3 Are you including the helpful 'description' parameter when generating the link to the redirect flow (mandate setup)?

This provides a best in class experience for end customers (payers). It ensures that they are aware of what they are creating a mandate for. More information here.

5.1.4 Are you displaying the GoCardless confirmation page, clearly telling an end customer (payer) that a Direct Debit mandate has been setup?

This provides a best in class experience for end customers (payers). Note, the GoCardless provided page is only available for 15 minutes from the moment the redirect flow is completed via the API. You must redirect end customers (payers) straight to it.

Sub-section 2 - Custom payment pages (hosted by the Partner)

The following requirements are only applicable if you are using your own custom payment pages

5.2.1 Have you followed our guidance to creating custom payment pages including the requirement to display a confirmation page and the GoCardless privacy notice?

The confirmation page must clearly tell the end customer (payer) that a Direct Debit mandate has been set up complete with the first payment date. You must also display a link to the GoCardless privacy notice. You can view the requirements here.

5.2.2 Are you using the 'Get a Single Creditor' API endpoint to understand which scheme(s) a user is using?

This provides a best in class experience for your users and will ensure mandates aren't initiated/created with the wrong currency. More information here.

5.2.3 Are you using the 'Bank Details Lookup' API endpoint to perform modulus checking?

This provides a best in class experience as it allows basic validity checking of end customer (payer) bank details which in turn improves conversion. More information here.

5.2.4 Are you using GoCardless generated mandate reference numbers?

This provides a best in class experience as it allows basic validity checking of end customer (payer) bank details which in turn improves conversion. More information here.

5.2.5 Please submit your custom payment pages for approval

If you want to use your own payment pages these need to be approved by GoCardless before we can set your integration live. You will need to provide relevant screenshots of payment pages (including the confirmation page) across each scheme you wish to use them in. You can find guides to creating custom payment pages on the GoCardless website (BACS - UK, SEPA - Eurozone, BECS - Australia, BECS NZ - New Zealand, PAD - Canada, ACH - USA).

Section 3 - Custom payment pages (individual user)

The following requirements are only applicable if you will be hosting custom payment pages on behalf of individual users.

5.3.1 Are you aware that any custom payment pages for individual users must be approved by GoCardless?

If you want to host payment pages on behalf of individual merchants these need to be approved by GoCardless before we can enable them. You will need to provide relevant screenshots of payment pages (including the confirmation page) across each scheme you wish to use them in. You can find guides to creating custom payment pages the GoCardless website (BACS - UK | SEPA - Eurozone, BECS - Australia, BECS NZ - New Zealand, PAD - Canada, ACH - USA).

5.3.2 For individual custom payment pages, are you receiving the details collected via the custom payment pages via the Partner API and using them to create customers/mandates via GoCardless?

This ensures data is created and managed in your platform first, ensuring it is synchronised correctly across different databases.

5.3.3 For individual custom payment pages, are you using the 'Bank Details Lookup' API endpoint to perform modulus checking?

This provides a best in class experience as it allows basic validity checking of end customer (payer) bank details which in turn improves conversion. More information here.

Section 6

Managing Mandates

It's important that your integration can handle mandate events to provide a best in class experience for your users and avoid the need for them to use the GoCardless dashboard.

Sub-section 1 - Handling mandate events

6.1.1 **Are you displaying real-time mandate information to your users?**

This provides a best in class experience for your users and avoids the need for them to use the GoCardless dashboard. You need to display the status and 'details [description]' to provide the user with a human description of any event ' details [cause]'. More information here.

Note - you will be required to share screenshots to demonstrate this functionality.

6.1.2 **Are you providing visibility to your users within your platform when an end customer (payer) cancels their mandate?**

You need to interpret and display the 'origin' 'cause' and 'description' to the users so they can take appropriate action. This provides a best in class experience for your users and avoids the need for them to use the GoCardless dashboard. More information here.

Note - you will be required to share screenshots to demonstrate this functionality.

6.1.3 **Are you providing visibility to your users within your platform when a mandate fails?**

You need to interpret and display the 'origin' 'cause' and 'description' to the users so they can take appropriate action. This provides a best in class experience for your users and avoids the need for them to use the GoCardless dashboard. More information here.

Note - you will be required to share screenshots to demonstrate this functionality.

6.1.4 **Are you providing visibility to your users within your platform when a mandate expires?**

You need to interpret and display the 'origin' 'cause' and 'description' to the users so they can take appropriate action. This provides a best in class experience for your users and avoids the need for them to use the GoCardless dashboard.

Note - you will be required to share screenshots to demonstrate this functionality.

6.1.5 **Are you able to handle the 'mandate_replaced' event in the event a user chooses to upgrade their GoCardless package (e.g. from Standard to Plus or Pro)?**

This functionality enables your users to upgrade their GoCardless package without the need for any manual intervention.

Sub-section 2 - Supporting mandes setup outside of the Partner platform

6.2.1 **Are users able to import mandates attached to an existing GoCardless account once they connect to your platform? Are these imports automated?**

Automatically loading and matching existing direct debit mandates makes importing your users' existing end customers (payers) simple. It also prevents the need for you users to cancel existing mandates and set them up again.

6.2.2 **Are you able to import mandates associated with a bulk change from a different Direct Debit provider?**

This provides a best in class experience for your users. It prevents the need for your users to cancel existing mandates and set them up again. More information here.

Section 7

Managing Payments

Once Direct Debit mandates have been set up with end customers, your users can start collecting payments. There are two ways to set up payments - either via the payments endpoint or the subscriptions endpoint.

You can choose which option to support depending on the kind of payments your users will want to take and how the process will be managed within your platform.

As with mandates, it is important that your integration can handle payment events to provide the best experience for your users and avoid the need for them to use the GoCardless dashboard.

Sub-section 1 - One-off payments (using the payments endpoint)

The following requirements are only applicable if you will be using the payments endpoint

7.1.1 Does your integration use the payments endpoint?

Partners can use either the subscriptions or payments endpoint when creating payments on behalf of users.

7.1.2 Are your users able to create payments of different amounts?

This provides a best in class experience for your users as well as their end customers (payers). More information here.

7.1.3 Are your users able to create payments in different currencies?

This is a requirement if your integration supports multiple different currencies. More information here.

7.1.4 **Are your users able to create payments with different payment collection dates?**

This provides a best in class experience for your users as well as their end customers (payers). More information here.

7.1.5 **Are your users able to select a 'charge date' when creating a payment?**

This provides a best in class experience for your users as well as their end customers (payers). It allows users to specify a charge date in the future. More information here.

Note - you must ensure that the date selected is no earlier than the 'next_possible_charge_date'.

7.1.6 **Are your users able to create/submit a payment at the same time as a mandate is created?**

Mandates do not have to be active before a payment can be created/submitted against it.

Note - in some schemes (e.g.SEPA) payments need to be submitted before a mandate is confirmed as active.

7.1.7 **Are your users able to cancel payments whilst in the 'pending_submission' state?**

This provides a best in class experience for your users as well as their end customers (payers). More information here.

7.1.8 **Are you using idempotency keys to avoid creating duplicate payments?**

This avoids ever billing the end customer (payer) twice should an API request time out or something goes wrong.

Note - your integration cannot be set live if you are not using idempotency keys. More information here.

7.1.9 **Are you users able to retry failed payments?**

This provides a best in class experience for your users and will help to reduce their failure rates. More information here.

Note - this should not involve the creation of a new payment but instead utilise the retry endpoint. You will be required to provide details of how a user does this within your platform.

7.1.10 **Are you controlling the advance payment and mandate setup notification emails?**

This is a requirement for subscription/membership platforms. It allows you to send one notification detailing the scheduleof all upcoming payments for an end customer, rather than having GoCardless send out an email each time a payment is reated. More information here.

Sub-section 2 - Recurring payments (using the subscriptions endpoint)

The following requirements are only applicable if you will be using the subscriptions endpoint

7.2.1 Does your integration use the subscriptions endpoint?

Partners can use either the subscriptions or payments endpoint when creating payments on behalf of users.

7.2.2 Are your users able to create subscriptions with payments of different amounts?

This provides a best in class experience for your users as well as their end customers (payers). More information here.

7.2.3 Are your users able to create subscription payments in different currencies?

This is a requirement if your integration supports multiple different currencies. More information here.

7.2.4 Are your users able to create subscription payments with different start dates?

This provides a best in class experience for your users as well as their end customers (payers). More information here.

7.2.5 Are your users able to create subscription payments with different payment/collection dates (i.e. different collection intervals)?

Your users should be able to create subscriptions with different payment/collection intervals (rather than fixed dates each week/month). This provides a best in class experience for yourusers as well as their end customers (payers). More information here.

7.2.6 Are your users able to cancel individual payments that are part of a subscription whilst still in the 'pending_ submission' state?

Users should be able to cancel individual payments that are part of a subscription whilst still in the 'pending_submission' state. This provides a best in class experience for your users as well as their end customers (payers). More information here.

7.2.7	**Are you using idempotency keys to avoid creating duplicate payments?** This avoids ever billing the end customer (payer) twice should an API request time out or something goes wrong. Note your integration cannot be set live if you are not using idempotency keys. More information here.	
7.2.8	**Are you users able to retry failed payments?** This provides a best in class experience for your users and will help to reduce their failure rates. Note this should not involve the creation of a new payment but instead utilise the retry endpoint.	Note - you will be required to provide details of how a user does this within your platform.
Sub-section 3 - Handling payment events		
7.3.1	**Are you displaying real-time payment information to your users?** This provides a best in class experience for your users and avoids the need for them to use the GoCardless dashboard. You need to display the status and 'details [description]' to provide the user with a human description of any event 'details [cause]'. More information here.	
7.3.2	**Are you providing visibility to your users within your platform when a payment fails?** You need to interpret and display the 'origin' 'cause' and 'description' to the users so they can take appropriate action. This provides a best in class experience for your users and avoids the need for them to use the GoCardless dashboard	Note- you will be required to share screenshots to demonstrate this functionality
7.3.3	**Are you providing visibility to your users within your platform when a payment is charged back?** You need to interpret and display the 'origin' 'cause' and 'description' to the users so they can take appropriate action. This provides a best in class experience for your users and avoids the need for them to use the GoCardless dashboard.	Note - you will be required to share screenshots to demonstrate this functionality.
7.3.4	**Are you providing visibility and/or alerting your users within your platform when a late payment failure occurs?** This provides a best in class experience for your users and avoids the need for them to use the GoCardless dashboard. More infomation here.	Note - you will be required to share screenshots to demonstrate this functionality.

Section 8

Managing Payouts and Reconciliation

Periodically (usually once every working day) we will pay out funds that users have collected from their end customers (payers). It is important that your integration can handle payout events and utilise the Payout Items API to reconcile these payments within your platform.

Sub-section 1 - General

8.1.1 **Does your integration handle reconciliation and use the Payout Items API to display individual transactions to your users?**

This provides a best in class experience for your users, ensuring they have full visibility of their payouts. It also avoids the need for users to login to the GoCardless dashboard.
More information here.

8.1.2 **Are you providing visibility to your users within your platform when a payment is confirmed?**

This provides a best in class experience for your users, ensuring they have full visibility of their payouts. It also avoids the need for users to use the GoCardless dashboard.
More information here.

Note - you will be required to share screenshots to demonstrate this functionality.

8.1.3 **Are you providing visibility to your users within your platform when a payment is paid out?**

This provides a best in class experience for your users, ensuring they have full visibility of their payouts. It also avoids the need for users to use the GoCardless dashboard.
More information here.

Note - you will be required to share screenshots to demonstrate this functionality.

8.1.4	**Are you displaying any app fee (if charged) to your users and marking this clearly in payouts?** Marking app fees clearly in payouts ensures your users have full visibility of their transaction costs and do not need to login to the GoCardless dashboard to obtain this information. As a Partner you can choose to either charge app fees or receive revenue share.	Note - if applicable, you will be required to share screenshots to demonstrate this functionality.

Sub-section 2 - Invoicing

The following requirements are only applicable if you are an invoicing platform

8.2.1	**Are invoices marked as paid/unpaid upon confirmation from GoCardless?** This provides a best in class experience for your users, allowing them to reconcile payments within your platform.	Note - if applicable, you will be required to share screenshots to demonstrate this functionality.
8.2.2	**Are you handling reconciliation of GoCardless fees and income at both the payment and payout level by allocating income to a designated fee account and fees to a designated nominal code?** This provides a best in class experience for your users, allowing them to reconcile payments within your platform.	Note - if applicable, you will be required to share screenshots to demonstrate this functionality.
8.2.3	**Are you allowing users to choose their own reconciliation settings for both income and fees?** This provides a best in class experience for your users, allowing them to reconcile payments within your platform.	

Section 9

Privacy and Security

Providing a lawful, stable and secure service for our shared users is essential. These are our minimum requirements for ensuring that we can protect our shared users and their end customers (payers) as well as treating personal and confidential business data appropriately.

9.1 **Does your API client use a modern version of TLS (1.2+ or newer)?**

This API client should validate certificates and avoid using weak ciphers. For more information see here.

9.2 **Are you using HTTPS (TLS 1.2+) for all interactions involving GoCardless data?**

HTTPS ensures secure communication.

9.3 **Are you ensuring that webhooks use a securely generated 30 byte or more secret that is unique?**

All webhooks must validate the secret.

Note - where a secret is not specified in the setup GoCardless generates a secure secret for you.

9.4 **Are you effectively managing access tokens and avoiding embedding them in code?**

More information available here.

9.5 **Are you ensuring that your users' data and sensitive information such as passwords and access tokens are encrypted at rest?**

This is a minimum requirement to ensure we can protect our shared users and their end customers (payers).

9.6 **Do you have controls in place to protect against the top 10 OWASP threats?**

More information available here.

9.7 **Do you have controls in place to meet the OWASP Application Security Verification Standard v4.0 Level 1?**

More information available here.

9.8 **Do you have 2FA enabled for your GoCardless account?**

This is a minimum requirement and can be enabled when you first create your account.

9.9 **Do you allow users to delete payer/user data when it expires or upon request?**

This ensures we meet our joint obligation to keep data no longer than necessary.

Section 10

Other

This final section covers our requirements for webhooks, error handling and in product discovery.

Sub-section 1 - Webhooks

10.1.1 Are you supporting webhooks to receive real-time notifications?

This provides a best in class experience for your users. Supporting webhooks enables automated actions to be taken in response to specific events e.g. payments failing, mandates expiring etc.

10.1.2 Are you processing webhooks asynchronously? Are you ensuring that webhooks use a securely generated 30 byte or more secret that is unique?

This avoids any issues with timing out if a large number of events are received at the same time.

10.1.3 Does your integration support GoCardless Rate Limiting? Are you effectively managing access tokens and avoiding?

This should be done via staging a high number of requests and handling the 'rate_limited_exceeded' error and/or tryingagain when it is reset as sent in the response header. This will help prevent any failed requests.

Note - if you don't respond to webhooks with a 2xx response within 10 seconds, the webhook will fail and we will then retry the webhook at increasing intervals e.g. 1 minute, 2 minutes, 10 minutes etc.

Sub-section 2 - In Product Discovery

10.2.1 Are users able to discover the GoCardless integration from within your platform? Are you promoting the GoCardless integration from within your platform?

This provides the best experience for your users and will help maximise uptake of your integration. More information here.

Note - you will be required to share screenshots to demonstrate this feature.

Sub-section 3 - Error handling

10.3.1 Question text: Does your integration handle all error types returned by the API (gocardless, invalid_api_usage, invalid_state, validation_failed)?

Errors should be passed on to your users and/or end customer (payer. To provide the best experience you need to be able to support users and/or end customers who have issues and monitor error rates, communicating with GoCardless where required. More information here.