

How to test Stripe 3DS2

This guide is mostly for QE but anyone can benefit from it.

Prerequisites

Have a setup of Porta/System somewhere.

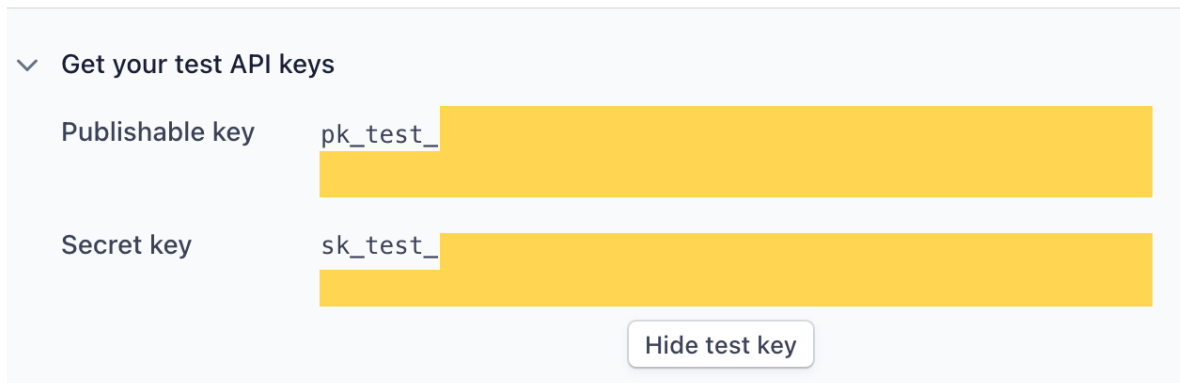
The environment must be either 'production' (deployed in an OpenShift cluster somewhere) or 'development' (locally). It won't work in 'preview'.

If the environment is 'development', it is recommended to follow [our installation guide](#), with the instructions given in [the description of this PR](#).

Steps

Setup the provider to be able to accept payments

1. Sign in to [the Stripe Dashboard](#) and get your test keys. If you do not have an account, you can create one for free and you do not even need to activate it just to get the test keys. They must look like the following image. It is important to note that they must be the test keys if you do not want to try real money being charged.



2. Configure payment settings in 3scale following [this documentation](#). Alternatively, you can quickly do it in the *rails console* like this:

```
> provider = Account.tenants.last
> provider.payment_gateway_options = {"login" => "sk_test_example",
  "publishable_key" => "pk_test_example"}
> provider.payment_gateway_type = :stripe
> provider.save!
> provider.billing_strategy.update(charging_enabled: true)
```

Create the developer account(s) that you will use to test charging them

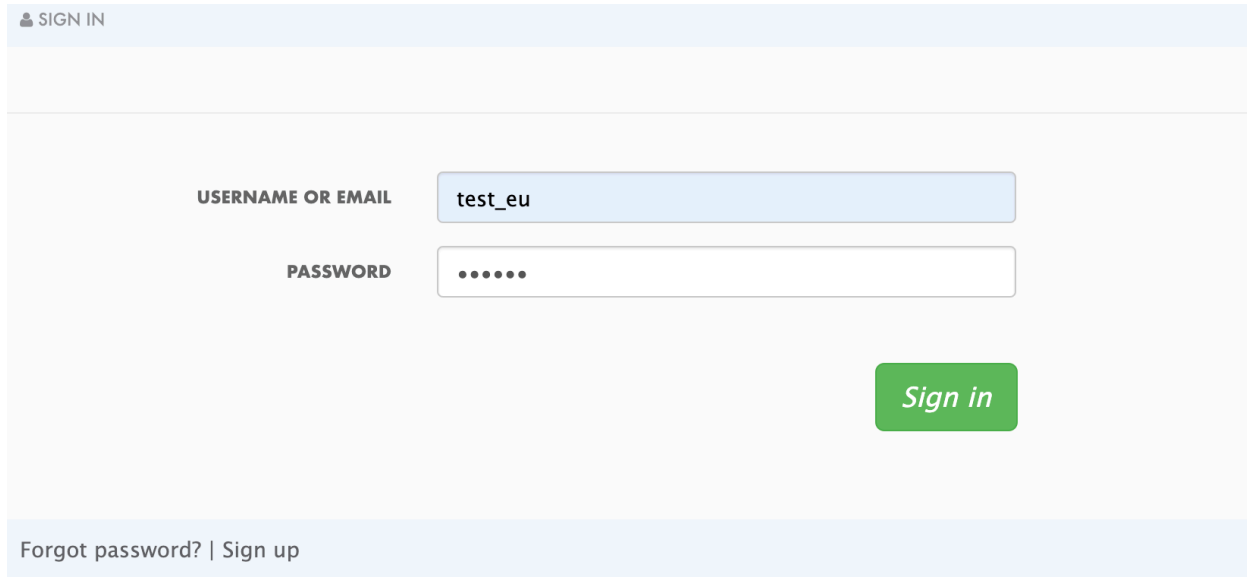
Follow [this documentation](#) to create a developer. Alternatively, you can quickly do it in the *rails console* like this:

```
> provider = Account.tenants.last
> signup_params = Signup::SignupParams.new(user_attributes: { email:
'test_eu@example.com', username: 'test_eu', first_name: 'Test', last_name: 'EU',
password: '123456' }, account_attributes: { org_name: 'test_eu', vat_rate: 33 })
> Signup::DeveloperAccountManager.new(provider).create(signup_params)
> provider.buyer_accounts.order(id: :asc).last.approve
> provider.buyer_accounts.order(id: :asc).last.admin_user.activate
```

Save the credit card data from the developer portal

This can only be done from the UI because it needs the JavaScript to interact with Stripe. Besides, this is the flow that we actually want to test so better doing it as the customer would.

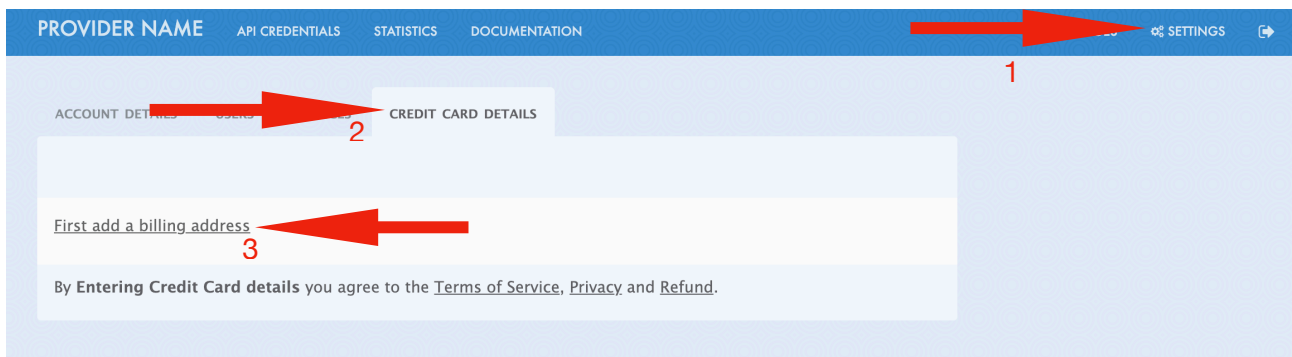
1. Login to the developer portal with the developer that you have created in the previous step.



The screenshot shows a login form titled "SIGN IN". It features two input fields: "USERNAME OR EMAIL" containing the text "test_eu" and "PASSWORD" containing six dots. A green "Sign in" button is positioned to the right of the password field. At the bottom left, there are links for "Forgot password?" and "Sign up".

Form in *yourDeveloperPortalUrl/login*

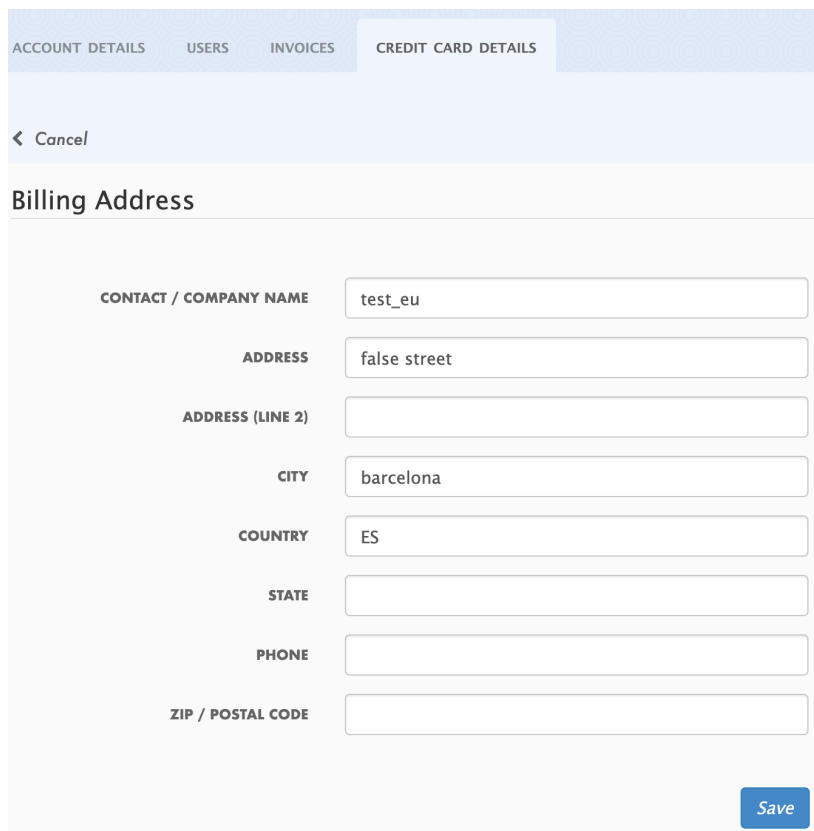
2. Navigate through the menu to insert *Credit Card Details*.



The screenshot displays the "ADMIN" interface for a provider. The top navigation bar includes "PROVIDER NAME", "API CREDENTIALS", "STATISTICS", "DOCUMENTATION", and "SETTINGS" (indicated by a red arrow labeled "1"). Below the navigation bar, there are tabs for "ACCOUNT DETAILS" and "CREDIT CARD DETAILS" (indicated by a red arrow labeled "2"). The "CREDIT CARD DETAILS" tab is active, showing a message: "First add a billing address" (indicated by a red arrow labeled "3"). Below this message, there is a note: "By Entering Credit Card details you agree to the [Terms of Service](#), [Privacy](#) and [Refund](#)."

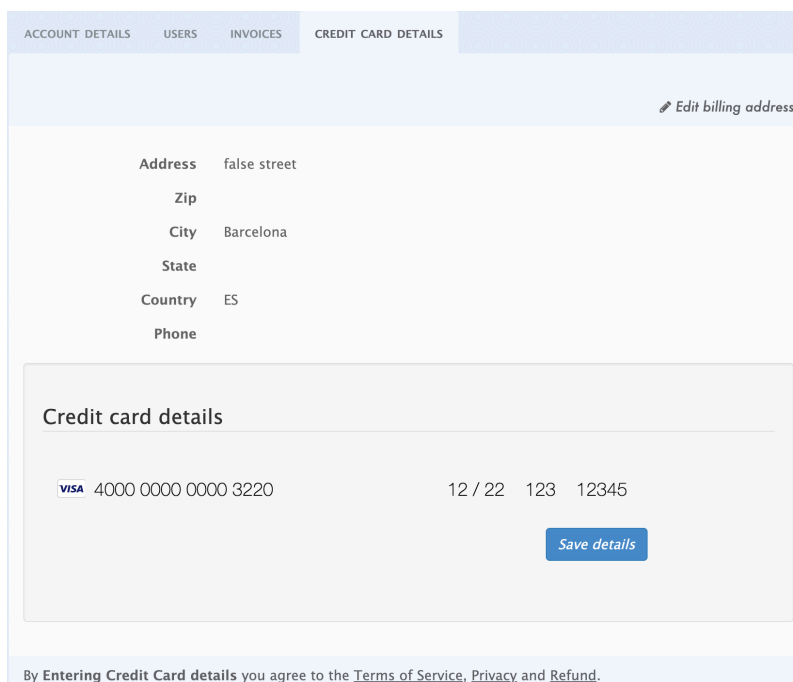
In *yourDeveloperPortalUrl/admin/account/stripe*

3. Fill the form of billing address and click on 'save'. There has been a change in the API of Stripe that allows countries only as ISO 3166-1 alpha-2 codes, while before we were sending the name of the countries directly. We have not solved this incompatibility issue yet and we will have to implement a fix to be decided yet.



The screenshot shows the 'Billing Address' form in a Stripe dashboard. The navigation tabs at the top are 'ACCOUNT DETAILS', 'USERS', 'INVOICES', and 'CREDIT CARD DETAILS'. A back arrow and 'Cancel' link are visible. The form fields are: 'CONTACT / COMPANY NAME' (test_eu), 'ADDRESS' (false street), 'ADDRESS (LINE 2)' (empty), 'CITY' (barcelona), 'COUNTRY' (ES), 'STATE' (empty), 'PHONE' (empty), and 'ZIP / POSTAL CODE' (empty). A 'Save' button is located at the bottom right.

4. Fill the form with the credit card details using a 3DS2 card from the stripe testing documentation. Then, click on 'Save details'. For my test, I am using the first one from the list, with the number 4000 0000 0000 3220.



The screenshot shows the 'Credit Card Details' form in a Stripe dashboard. The navigation tabs at the top are 'ACCOUNT DETAILS', 'USERS', 'INVOICES', and 'CREDIT CARD DETAILS'. An 'Edit billing address' link is visible. The form displays the following information: Address: false street, Zip: (empty), City: Barcelona, State: (empty), Country: ES, Phone: (empty). Below this, the 'Credit card details' section shows a VISA card with the number 4000 0000 0000 3220, an expiration date of 12 / 22, and the name 123 12345. A 'Save details' button is located at the bottom right. At the bottom of the page, there is a footer: 'By Entering Credit Card details you agree to the [Terms of Service](#), [Privacy](#) and [Refund](#).'

5. Complete de 3D Secure 2 authentication.

The image shows a dark grey background with a navigation menu at the top containing 'ACCOUNT DETAILS', 'USERS', 'INVOICES', and 'CREDIT'. On the left, there is a form for credit card details with the following fields: 'Address' (false street), 'Zip', 'City' (Barcelona), 'State', 'Country' (ES), and 'Phone'. Below this is a section titled 'Credit card details' showing a VISA card number '4000 0000 0000 3220'. A modal window is centered on the screen with a blue header that reads '3D Secure 2 Test Page'. The modal contains the text: 'This is a test 3D Secure 2 authentication for a transaction with . In live mode, customers will be asked to verify their identity with a push notification, a text message, or another method chosen by their bank.' At the bottom of the modal are two buttons: 'FAIL' and 'COMPLETE'. A red arrow points to the 'COMPLETE' button. In the top right corner of the modal, there is a 'CANCEL' link. The word 'address' is visible on the right side of the background form.

Create & Charge invoice(s)

I recommend doing it with more than 1 invoice for the same provider and buyer, with different amounts, to ensure that it keeps working for more than 1 payment with different amounts.

1. Go to *Audience* -> *Listing* in the Admin Portal and click on the developer account. Then go to *Invoices* (that will take you to `yourAdminPortalUrl/buyers/accounts/developerAccountId/invoices`) and click on 'Create Invoice'.

That will create an invoice, but you can also do it from the rails console like this:

```
> provider = Account.tenants.last
> developer = provider.buyer_accounts.last
> invoice = Invoice.create!(provider_account: provider, buyer_account:
developer, period: Month.new(2020, 12))
```

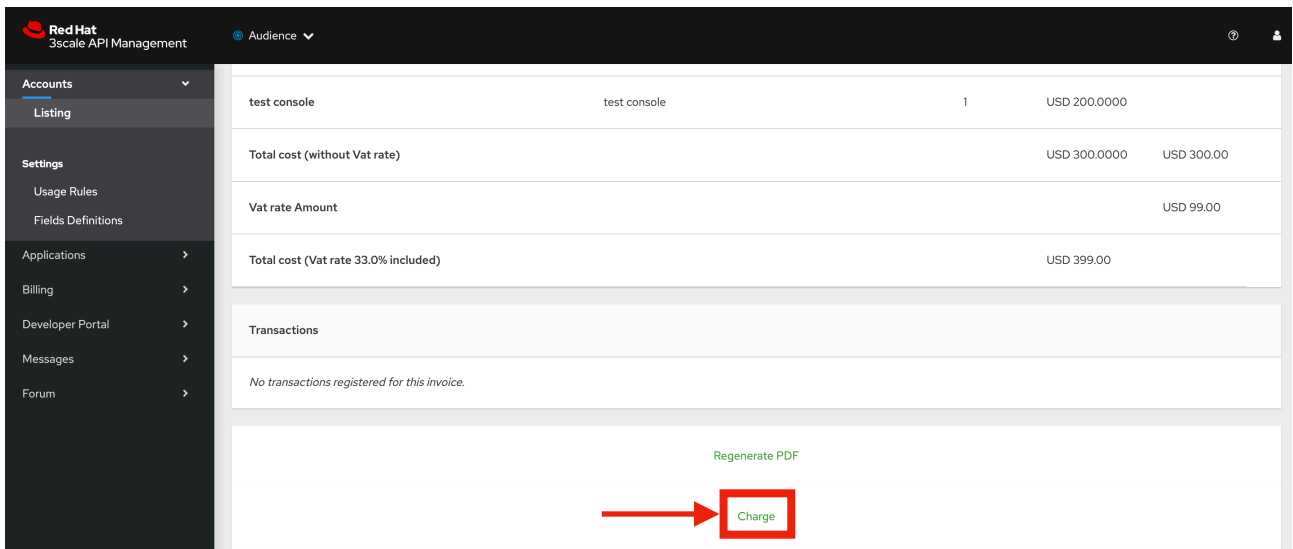
2. Create a *line item* for the invoice. With any data you want. In the UI this is done like this:

Name	Quantity	Price	Charged	➕ Add
Total cost (without Vat rate)		USD 0.0000	USD 0.00	1
Vat rate Amount			USD 0.00	
Total cost (Vat rate 33.0% included)			USD 0.00	
Transactions				
No transactions registered for this invoice				

In the *rails console* it is done this way:

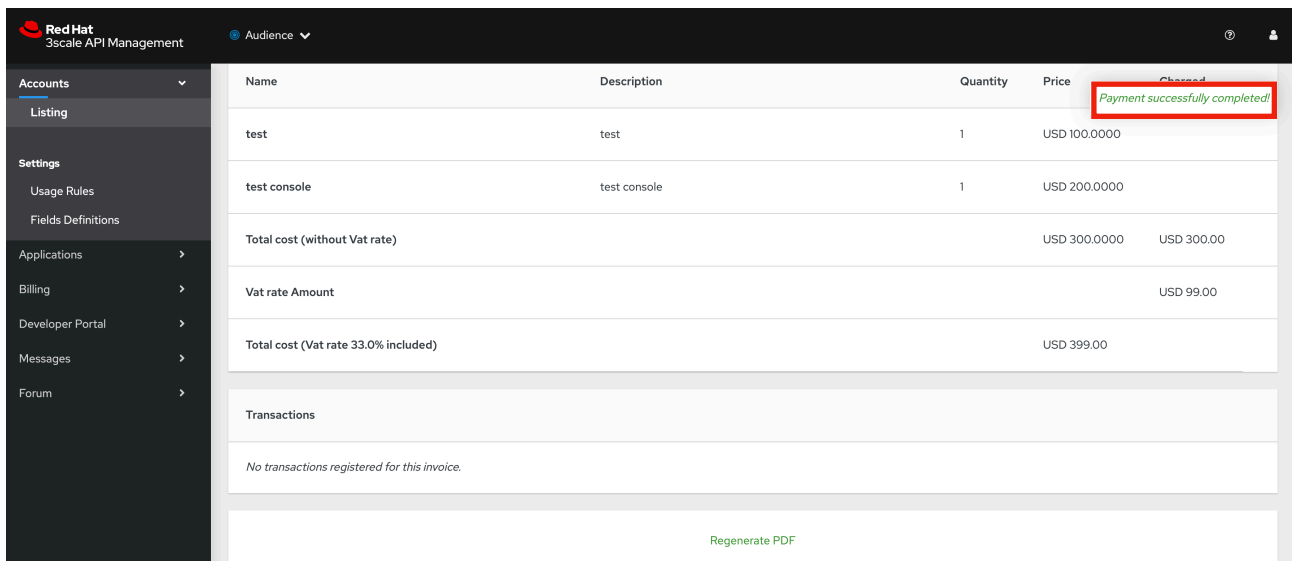
```
> invoice = Invoice.last
> invoice.line_items.create!({name: 'test console', quantity: '1', description: 'test
console', cost: '200.00'})
```

3. Change the invoice *state* to 'pending' by issuing it. If you are doing it from the UI, click to the 'invoice' that you have created, and inside the show view, click on 'Issue Invoice'. From the *rails console* the command is "`invoice.issue`".
4. Charge invoice. This can be done from the Rails console, but as this is also part of what we actually want to test here, it is better to do it from the UI as the customer would. This is how it is done (confirm any alert that pops up in the browser):



In yourAdminPortal/buyers/accounts/developer_id/invoices/invoice_id

- The next screenshots are the result you should see. However, this does not mean that the payment is done yet, as you can see in the dashboard of stripe.



Instant feedback

Transactions					
Status	Created At	Reference	Message	Amount	
Success	2020-12-17 22:55:26 UTC	pi_lHHzVNaIxGJbGz9pue9KcKwbD	Transaction approved	399.0 USD	

Transactions table after reloading the page