



Red Hat 3scale 2.2

Billing

For Use with Red Hat 3scale 2.2

Red Hat 3scale 2.2 Billing

For Use with Red Hat 3scale 2.2

Legal Notice

Copyright © 2018 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This guide documents billing management with Red Hat 3scale 2.2.

Table of Contents

1. CONFIGURE BILLING	2
2. ADDING VAT RATE ON BILLING	2
3. ADYEN INTEGRATION	3
3.1. Prerequisites	4
3.2. Step 1: Find the information in your Adyen account	4
3.2.1. Step 2: Configure your payment gateway in your 3scale account	6
3.2.2. Step 3: Enable the "alias" additional data in the Adyen API response	7
3.2.3. Step 4: Test your billing workflow	7
3.2.4. Troubleshooting	8
4. BRAINTREE INTEGRATION	8
4.1. Step 1: Get your API keys from Braintree	8
4.2. Step 2: Configure your settings in 3scale	10
4.2.1. Note	12
4.2.2. Troubleshooting	12
5. CREDIT CARD FLOW	12
5.1. Prerequisites	12
5.2. Step 1: Hide certain menu items when credit card details are missing	12
5.2.1. Editing users_menu partial	13
5.2.2. Editing submenu partial	13
5.3. Step 2: Enable Feature	15
5.4. Step 4: Ensure credit card is entered by developer when upgrading from a free to a paid plan	15
5.4.1. Editing applications/show Template	16
5.4.2. Editing applications/form partial	16
6. STRIPE INTEGRATION (RECOMMENDED)	16
6.1. Prerequisites	16
6.2. Step 1: Get your API keys from Stripe	16
6.3. Configure settings in 3scale	17
6.3.1. Note	20

1. CONFIGURE BILLING

In this tutorial you'll learn how the 3scale billing system works and how to set it up.

When setting up your billing configuration, you can choose between Prepaid and Postpaid.

- **Prepaid mode:** In prepaid mode, all fixed fees and setup fees are billed immediately at the beginning of the month or at the beginning of any pro-rated billing period. Variable costs are always calculated at the end of the month.
- **Postpaid mode:** In postpaid mode, all fixed fees as well as variable fees are billed at the end of the month.

Here is a visual example of billing for the [prepaid mode](#) and for the [postpaid mode](#).

To do so, go to **Settings > Billing**

3scale Dashboard Developers Applications Billing Analytics APIs Developer Portal **Settings**

General Developer Portal Legal Terms **Billing** Policies Fields Definitions Web Hooks Emails

Finance Settings

Mode: **POSTPAID** [Switch to PREPAID](#)

In postpaid mode, all fixed fees as well as variable fees are billed at the end of the month.

Charging enabled
Enables credit card transactions. When this setting is on, all due invoices will be charged using the selected payment gateway. If you leave this setting off, billing will take place and invoices will issued, but no real transaction will take place.

Currency* USD - American Dollar

Please make sure that the selected currency is supported by your payment gateway. If you are missing currency options, don't hesitate to contact us.

Invoice footnote

Text to show if VAT/Sales Tax is 0%

Billing periods for invoice ids* monthly

[Save](#)

All paid plans are charged monthly.

2. ADDING VAT RATE ON BILLING

After completing these steps, you'll have configured Adyen as a payment gateway for your account. This will allow your developers to enter their credit card details, and you can automatically charge them through Adyen for access to your API, according to the calculated invoices.

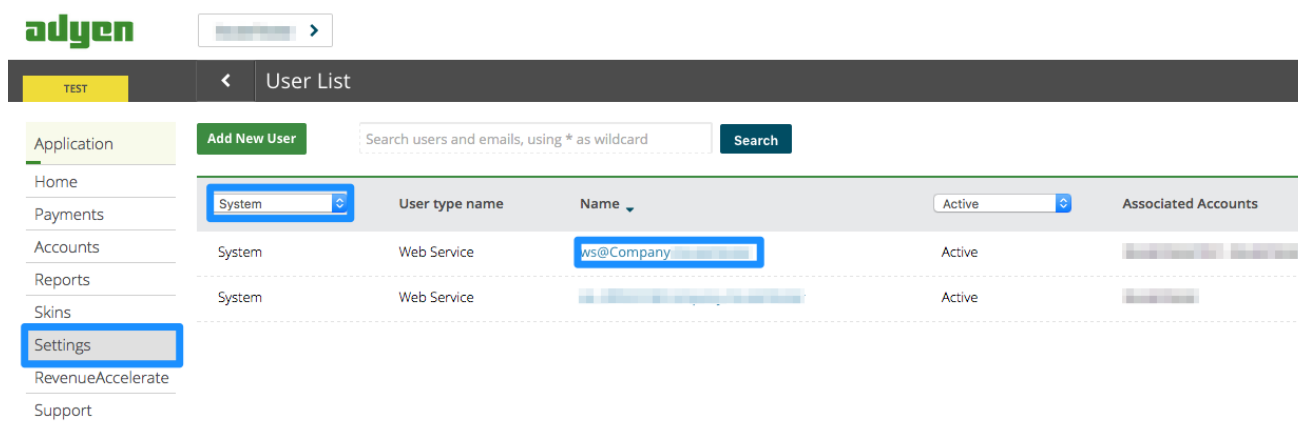
Setting up your payment gateway is a key step enabling credit card charging for use of your paid API. There are a number of alternative payment gateways you can use with your 3scale account. Here we cover the steps for Adyen.

3.1. Prerequisites

Before you start these steps, you'll need to open an account with [Adyen](#). You need a *Company* account and a *Merchant* account within it (sub-account). There are a number of requirements that must be fulfilled before you can apply for a live account with Adyen. You can see what those requirements are [here](#).

3.2. Step 1: Find the information in your Adyen account

To begin, log into your Adyen account. Then find your credentials in the **Settings** > *Users* area, and select *system* from the dropdown menu in the following view:



The screenshot shows the Adyen user management interface. On the left, a sidebar menu has 'Settings' highlighted. The main area is titled 'User List' and contains a table of users. The table has columns for 'System', 'User type name', 'Name', 'Active', and 'Associated Accounts'. The first row shows a 'System' user of type 'Web Service' with the name 'ws@Company'. The second row shows another 'System' user of type 'Web Service' with a redacted name. A search bar is located above the table.

System	User type name	Name	Active	Associated Accounts
System	Web Service	ws@Company	Active	[Redacted]
System	Web Service	[Redacted]	Active	[Redacted]

Click on the "Company" account (the account at the top of the list). You will then be taken to the Company account settings view:

The screenshot displays the 'Edit Web Service User' interface in the Adyen Admin console. The left sidebar shows the navigation menu with 'Settings' selected. The main content area is divided into four panels:

- User Account Details:** Contains fields for 'User Name', 'Last Password Change' (2016-05-05 15:39:51 CEST), 'Password', and 'Retype Password'. It includes 'Generate Password' and 'Generate POS Password' buttons. A 'Client Certificate (DN)' field is also present.
- Easy Encryption:** Features a 'Public Key' field with a preformatted string: `"10001|C9CEA1701A8BB203254B314A081C814E8...`. Below it, the 'Hosted Client Side Encryption' section has a 'Library Version' dropdown set to 'Always latest version' and a 'Library location' dropdown. A green notification box states: 'The V0_1_X CSE library is available, running library version 0_1_17.'
- Roles and Associated Accounts:** Shows a search bar and a table of roles and accounts. The table has columns for 'Role', 'Account', and a toggle switch. The '3scaleTester Comp...' role is associated with the 'MerchantAccount' account, and its toggle is turned on. A 'Merchant Id' label is visible next to the toggle.
- Edit Allowed User IP Range:** Includes an 'Add IP address' field with the value '0.0.0.1', an 'Allow' dropdown set to '32 (this IP address only)', and an 'Allow' button. A note at the bottom states 'No allowed IP address ranges configured'.

A 'Save' button is located at the bottom left of the 'Roles and Associated Accounts' panel.

Finally you have access to the **Login**, **Secret Password**, **Client Encryption Public Key**, **Merchant ID** and **Library location** that you will need for the 3scale billing settings. To be able to view the **Public Key**, you must click on "generate password" and copy this password somewhere.

User Account Details

User Name [Redacted]

Last Password Change 2016-05-05 15:39:51 CEST

Password [Redacted]

Retype Password [Redacted]

Client Certificate (DN) [Redacted]

Easy Encryption

Client Encryption Public Key [Redacted]

Preformatted for direct insertion in your code:
[Redacted]

Hosted Client Side Encryption

Library Version (changes will be applied after you click save)

The V0_1_X CSE library is available, running library version 0_1_17.

All these switches should be enabled by default but if for some reason they are not you should enable at least these 3 for successful integration

Roles and Associated Accounts

Type to search roles

API Clientside Encryption Payments role	<input checked="" type="checkbox"/>
iDeal Recurring	<input type="checkbox"/>
Merchant PAL Webservice role	<input checked="" type="checkbox"/>
Merchant Recurring role	<input checked="" type="checkbox"/>

Edit Allowed User IP Range

Add IP address . . .

Allow

No allowed IP address ranges configured

3.2.1. Step 2: Configure your payment gateway in your 3scale account

In the billing area of the Settings tab, enable charging by checking the box and click **save**. You will see all the fields that you need to setup to create the link to your Adyen gateway. Remember to select Adyen from the gateway dropdown menu and save changes.



Finance Settings

 Mode: **PREPAID** [Switch to POSTPAID](#)

In prepaid mode, all fixed fees and setup fees are billed immediately at the beginning of the month or at the beginning of any pro-rated billing period. Variable costs are always calculated at the end of the month.

Charging enabled

Enables credit card transactions. When this setting is on, all due invoices will be charged using the selected payment gateway. If you leave this setting off, billing will take place and invoices will be issued, but no real transaction will take place.

Currency*

USD - American Dollar

Please make sure that the selected currency is supported by your payment gateway. If you are missing currency options, don't hesitate to contact us.

Invoice footnote

Text to show if VAT/Sales Tax is 0%

Billing periods for invoice ids*

monthly

Save

Credit card gateway

Select your credit card gateway and configure it with the details of your merchant account.

Basics

Gateway

Adyen

Adyen Options

Login

Secret Password

Client Encryption Public Key

Merchant ID

Library location

Save changes

3.2.2. Step 3: Enable the "alias" additional data in the Adyen API response

By default when credit card authorization requests are sent from 3scale to Adyen, the returned response does not include the unique identifier of the credit card. To ensure that the correct credit card reference is saved in 3scale and the correct card is charged, this additional data needs to be enabled. In order to do this, you should ask Adyen support to enable the "alias" additional data in the response for the authorization request.

3.2.3. Step 4: Test your billing workflow

Make sure you accelerate the test cycle by enabling *Prepaid Mode* to generate the charge within a day or so. Then choose an existing test account and create an invoice with a line item charge added. Charge the account immediately. This testing approach will incur some minor costs, but it is worth it for the peace of mind that everything works fine, before you have real paying developers using your API.

The payment gateway is now set up, but your users might not be able to use it yet since it is not configured in the CMS. Go to the developer portal tab, and find the template called **Payment Gateway / Show** on the left navigation pane.

If it's not there already, add the following snippet after the block of code beginning with `{% when "stripe" %}`

```
{% when "adyen12" %}
{% if current_account.has_billing_address? %}
  {% adyen12_form %}
{% else %}
  <p><a href="{{ current_account.edit_adyen12_billing_address_url }}">First add a billing address</a></p>
{% endif %}
```



NOTE

- For accounts created before 11th May 2016 you must add the snippet above manually. After said date this will be included in the template by default.
- In order to map your data from Adyen with your data on 3scale, you can use the Adyen field called **shopperReference** which is composed of **3scale-[PROVIDER_ID] - [DEVELOPER_ACCOUNT_ID]**

3.2.4. Troubleshooting

If your account is in sandbox mode and you encounter any problems, you'll have to change it to production.

4. BRAINTREE INTEGRATION

These are the steps to set up the Braintree gateway in order to to charge for use of your API.

4.1. Step 1: Get your API keys from Braintree

You'll need to open an account with [Braintree](#). You need a Gateway and Merchant account plus Vault. As an optional extra, you can choose to allow American Express cards as a payment method.

To begin, log in to your Braintree account. Then find your API keys in the **Account > MyUser** area:

Braintree Welcome back [redacted] to the [redacted] Control Panel Basic Search Search Account Logout

Search Transactions Vault Recurring Billing

Edit User test_3scale

User Details

Name [redacted]
Email [redacted]

Change Password (Optional)

Current Password [redacted]
New Password [redacted]
Confirm New Password [redacted]

Authorization

API Keys

Notifications

Email Daily Settlement Batch Summary

Save Cancel

Questions? Shoot us an email at support@braintreepayments.com or call us at 877.434.2894. ©2011-2012 Braintree Payments Solutions, LLC. All rights reserved.

The API keys page still hides the private key, so select the option to view it:

Braintree Welcome back [redacted] to the [redacted] Control Panel Basic Search Search Account Logout

Search Transactions Vault Recurring Billing

API Keys

Public Key	Private Key	Last Used On	Actions
[redacted]	[redacted] View	[redacted]	[redacted]

Merchant ID [redacted]
Environment sandbox

Generate New

Questions? Shoot us an email at support@braintreepayments.com or call us at 877.434.2894. ©2011-2012 Braintree Payments Solutions, LLC. All rights reserved.

Finally you have the Public Key, Private Key, and Merchant ID that you'll need for the 3scale billing settings:

The screenshot shows the Braintree Control Panel interface. At the top, there is a navigation bar with the Braintree logo, a welcome message, a search bar, and account/logout options. On the left, there is a sidebar with navigation links for Search, Transactions, Vault, and Recurring Billing. The main content area is titled "Client Library Key" and displays four fields: Public Key, Private Key, Environment (set to "sandbox"), and Merchant ID. The Public Key, Private Key, and Merchant ID fields are highlighted with blue boxes. Below these fields is a "Usage" table with columns "Client Library" and "Last Used On". The table lists three entries: Braintree Ruby Gem 2.10.3, Braintree Ruby Gem 2.10.2, and Transparent Redirect, each with a corresponding "Last Used On" date. At the bottom, there is a text block explaining that the keys will be copied to the clipboard and a "Copy" button with a language dropdown set to "Java".

4.2. Step 2: Configure your settings in 3scale

You need to tell 3scale to start using these API keys. To do this, log in to your 3scale Admin Portal and go to **Settings > Billing**.

The screenshot shows the 3scale Admin Portal interface. At the top, there is a navigation bar with the 3scale logo and several menu items: Dashboard, Developers, Applications, Billing, Analytics, APIs, Developer Portal, and Settings. The Settings menu item is highlighted with a blue box. Below the navigation bar, there is a sub-navigation bar with several items: General, Developer Portal, Legal Terms, Billing, Policies, Fields Definitions, Web Hooks, and Emails. The Billing item is highlighted with a blue box. The main content area is titled "Finance Settings" and contains several configuration options. The "Mode" is set to "POSTPAID" with a "Switch to PREPAID" button. Below this, there is a text block explaining that in postpaid mode, all fixed fees as well as variable fees are billed at the end of the month. There is a checkbox for "Charging enabled" which is currently unchecked. Below this, there is a "Currency" dropdown menu set to "USD - American Dollar". There is a text input field for "Invoice footnote" and another for "Text to show if VAT/Sales Tax is 0%". At the bottom, there is a "Billing periods for invoice ids" dropdown menu set to "monthly". A "Save" button is located at the bottom right of the page.

If the Charging Enabled flag is not active, enable it and click Save.

3scale Dashboard Developers Applications Billing Analytics APIs Developer Portal **Settings**

General Developer Portal Legal Terms **Billing** Policies Fields Definitions Web Hooks Emails

Finance Settings

Mode: **POSTPAID** [Switch to PREPAID](#)

In postpaid mode, all fixed fees as well as variable fees are billed at the end of the month.

Charging enabled

Enables credit card transactions. When this setting is on, all due invoices will be charged using the selected payment gateway. If you leave this setting off, billing will take place and invoices will issued, but no real transaction will take place.

Currency*

Please make sure that the selected currency is supported by your payment gateway. If you are missing currency options, don't hesitate to contact us.

Invoice footnote

Text to show if VAT/Sales Tax is 0%

Billing periods for invoice ids*

Save

You should see a drop-down called Gateway near the bottom of the page. Change it to Braintree (Blue Platform).

Credit card gateway

Select your credit card gateway and configure it with the details of your merchant account.

Basics

Gateway

- ✓ Choose one
- Authorize.Net
- Braintree (Blue Platform)**
- Ogone
- Stripe

Save changes

The form below the drop-down should change to show two fields. Insert your Braintree keys and click Save.


Credit card gateway

Select your credit card gateway and configure it with the details of your merchant account.

Basics

Gateway Braintree (Blue Platform) ▾

Braintree (Blue Platform) Options

Public Key		
Merchant ID		
Private Key		

Save changes

You might see a couple of alerts when you change your payment gateway. This is expected. Read and accept them if they appear.

Your users should now be able to pay you using the Braintree gateway.

4.2.1. Note

In order to map your data from Braintree with your data on 3scale, you can use the Braintree field called **customer . id** which is composed of **3scale - [PROVIDER_ID] - [DEVELOPER_ACCOUNT_ID]**

4.2.2. Troubleshooting

In case your account is in the sandbox mode and you encounter any problems, you will have to change it to production.

5. CREDIT CARD FLOW

This guide will help you define a great workflow in your Developer Portal to ensure that developers submit their credit card details before they gain access to your paid API(s).

It's increasingly common to see businesses charge for access to their APIs. Sometimes the entire business model of a company is built around paid APIs, while other times it's an important revenue stream to cover costs or generate profits for the company.

5.1. Prerequisites

- 3scale finance module (billing and charging) activated and configured
- Paid plans defined (usually application plans)

After following the steps detailed below, all the existing developers who have paid plans but no credit card details entered, will be redirected to the credit card details form. They will not be allowed to access the portal until they have successfully entered their credit card details.

5.2. Step 1: Hide certain menu items when credit card details are missing

When a developer signs up to a paid plan and is required to fill in the credit card details, it is recommended to hide some menu items since it will not be possible to navigate there until the credit card is entered. In order to do so, you will need to change the following 2 menu partials:

- users_menu
- submenu

5.2.1. Editing users_menu partial

You will need to replace the whole of the partial content with the below:

```
<ul class="nav nav-tabs">
  {% unless current_account.requires_credit_card_now? %}
    <li class="{% if urls.account_overview.active? %}active{% endif %}">
      <a href="{% if urls.account_overview %}">Account Details</a>
    </li>

    {% if provider.account_management_enabled? %}
      {% include 'menu_item' with urls.users %}

      {% if provider.multiple_users_allowed? %}
        {% include 'menu_item' with urls.invitations %}
      {% endif %}
    {% endif %}

    {% if provider.account_plans.size > 1 %}
      {% include 'menu_item' with urls.account_plans %}
    {% endif %}
  {% endunless %}

  {% if provider.finance_allowed? %}
    {% unless current_account.requires_credit_card_now? %}
      {% include 'menu_item' with urls.invoices %}
    {% endunless %}

    {% if urls.payment_details %}
      {% include 'menu_item' with urls.payment_details %}
    {% endif %}
  {% endif %}
</ul>
```

If you have customised this partial already, e.g by adding some css, modifying the HTML tags, etc... you should use the new liquid tag for this purpose accordingly:

```
current_account.requires_credit_card_now?
```

5.2.2. Editing submenu partial

The developer gains access to their keys to start making calls on your API through the API credentials. The changes below ensure that you only display the links if the account does not require credit card details to be entered e.g because there are no paid plans or if the credit card is already available. If you wanted to be completely strict about this, you would edit the system page for displaying the keys to restrict the display – but this would be an advanced customization. You may also want to delete the snippet to display credentials directly on the home page after login.

You will need to replace the whole of the partial content with the below:

```
{% if current_user %}
```

```

    {% unless current_account.requires_credit_card_now? %}
    <ul class="nav navbar-nav">

      {% if provider.multiple_applications_allowed? %}
      <li class="{% if urls.applications.active? %}active{% endif
%}">
        <a href="{ { urls.applications } }">{{
urls.applications.title }}</a>
      </li>
      {% elsif current_account.applications.first == present%}
      {% assign app = current_account.applications.first %}
      <li class="{% if app.url.active? %}active{% endif %}">
        <a href="{ { app.url } }">API Credentials</a>
      </li>
      {% elsif current_user.can.create_application? %}
      <li class="{% if url.new_application.active? %}active{% endif
%}">
        <a href="{ { urls.new_application } }">Get API Credentials</a>
      </li>
      {% else %}
      <!-- You don't have any application neither can create one.
Bad luck. -->
      {% endif %}

      {% assign live_apps_present = current_account.applications |
map: 'state' | any: 'live' %}
      {% if live_apps_present %}
      <li class="{% if urls.stats.active? %}active{% endif %}">
        <a href="{ { urls.stats } }">Statistics</a>
      </li>
      {% endif %}

      {% if provider.multiple_services_allowed? %}
      <li class="{% if urls.services.active? %}active{% endif %}">
        <a href="{ { urls.services } }">{{ urls.services.title }}</a>
      </li>
      {% endif %}

      <li><a class="{% if urls.docs.active? %}active{% endif %}"
href="/docs">Documentation</a></li>

    </ul>
    {% endunless %}
    <ul id="user_widget" class="nav navbar-nav navbar-right">
      {% unless current_account.requires_credit_card_now? %}
      <li class="{% if urls.messages_inbox.active? %}active{% endif
%}">
        <a href="{ { urls.messages_inbox } }">
          Messages
          {% if current_account.unread_messages.size > 0 %}
          {{ current_account.unread_messages.size }}
          {% endif %}
        </a>
      </li>
      <li class="{% if urls.account_overview.active? %}active{% endif
%}">

```

```

        <a href="{{ urls.account_overview }}">
            Settings
        </a>
    </li>
{% endunless %}
<li>
    <a id="sign-out-button" class="pull-right sign-out"
title="Logout {{ current_user.username }}" href="{{ urls.logout }}">

    </a>
</li>
</ul>

{% else %}
<ul class="nav navbar-nav">
    <li><a href="/docs">Documentation</a></li>
    <li><a href="/#plans">Plans</a></li>
</ul>
<ul class="nav navbar-nav navbar-right">
    <li>
        <a href="{{ urls.login }}">
            Sign in
        </a>
    </li>
</ul>
{% endif %}

```

Again, if you have customised this partial already, you will need to modify your version accordingly using the same new liquid tag: `current_account.requires_credit_card_now?`

5.3. Step 2: Enable Feature

To enable this feature, you must enable the following 2 switches:

Switches can be found in *Developer portal > Feature visibility*.

- Finance
- Credit Card on Signup (only available if you are on an older 3scale account.)

5.4. Step 4: Ensure credit card is entered by developer when upgrading from a free to a paid plan

You may allow developers to select plans when changing plans. The ideal setting is Request credit card entry for paid plans. This will ensure that if a paid plan is selected, the plan change can only be done instantly if a credit card is stored. Otherwise a notification is displayed to point the developer to the location to enter their card details.

This setting is defined from the Admin Portal. Go to **API > Service > Settings > Application Plan Changing**. Here you can select the option Request credit card entry for paid plans.

To enable this behaviour, you will need to make changes to the following CMS pages:

- applications/show template

- applications/form partial

5.4.1. Editing applications/show Template

You will need to look for `plan_widget` in the page code and change it to the following:

```
{% plan_widget application, wizard: true %}
```

If you have added the `plan_widget` in any other pages, you will need to update those to match the above snippet.

5.4.2. Editing applications/form partial

Follow the same instructions above to change any instances of `plan_widget` in the applications/form partial to conform to the new snippet.

And that's it! You can now enforce the addition of credit card details before allowing usage of your API for paid plans.

6. STRIPE INTEGRATION (RECOMMENDED)

After completing these steps you will have configured Stripe as a payment gateway for your account. This will allow your developers to enter their credit card details, and you can automatically charge them through Stripe for access to your API, according to the calculated invoices.

If you enable credit card charging for your paid API, then one key step is to setup your [payment gateway](#). There are a number of alternative payment gateways which you can use with your 3scale account. Here we cover the steps for Stripe.

6.1. Prerequisites

Before you start these steps, you will need to open an account with [Stripe](#).

6.2. Step 1: Get your API keys from Stripe

Log in to your Stripe account, and get your API keys at <https://dashboard.stripe.com/account/apikeys>. You will need two keys: a "secret" one, and a "publishable" one. Use the "test" set when you're doing tests and the "live" ones when you are ready to start charging.

General Team **API Keys** Subscriptions Transfers Webhooks Connect Relay Data Emails

API version: No [version](#) yet — it'll be set when you start using the API.

Test Secret Key:

Test Publishable Key:

Live Secret Key:

Live Publishable Key:

[Learn more about API authentication](#) [Done](#)

6.3. Configure settings in 3scale

You need to tell 3scale to start using these API keys. To do this, log in to your 3scale admin portal and go to **Settings > Billing**



Finance Settings

Mode: **POSTPAID** [Switch to PREPAID](#)

In postpaid mode, all fixed fees as well as variable fees are billed at the end of the month.

Charging enabled

Enables credit card transactions. When this setting is on, all due invoices will be charged using the selected payment gateway. If you leave this setting off, billing will take place and invoices will issued, but no real transaction will take place.

Currency* USD - American Dollar

Please make sure that the selected currency is supported by your payment gateway. If you are missing currency options, don't hesitate to contact us.

Invoice footnote

Text to show if VAT/Sales Tax is 0%

Billing periods for invoice ids* monthly

Save

If the Charging Enabled flag is not active, enable it and click Save.



Finance Settings

 Mode: **POSTPAID** [Switch to PREPAID](#)

In postpaid mode, all fixed fees as well as variable fees are billed at the end of the month.

 Charging enabled

Enables credit card transactions. When this setting is on, all due invoices will be charged using the selected payment gateway. If you leave this setting off, billing will take place and invoices will issued, but no real transaction will take place.

 Currency* USD - American Dollar

Please make sure that the selected currency is supported by your payment gateway. If you are missing currency options, don't hesitate to contact us.

Invoice footnote

Text to show if VAT/Sales Tax is 0%

 Billing periods for invoice ids* monthly

Save

You should see a drop-down called Gateway near the bottom of the page. Change it to Stripe.

Credit card gateway

Select your credit card gateway and configure it with the details of your merchant account.

Basics

Gateway

- ✓ Choose one
- Authorize.Net
- Braintree (Blue Platform)
- Ogone
- Stripe**

Save changes

The form below the drop-down should change to show two fields. Insert your Stripe API keys and click Save.

Credit card gateway

Select your credit card gateway and configure it with the details of your merchant account.

Basics

Gateway

Stripe Options

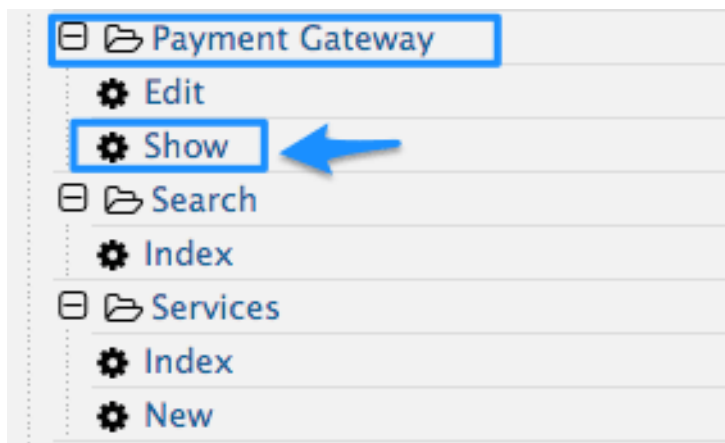
Secret Key

Publishable Key

[Save changes](#)

You might see a couple of alerts when you change your payment gateway. This is expected. Read them and accept them if they appear.

The payment gateway is now set up, but your users might not be able to use it yet since it's not configured in the CMS. Go to Developer portal, and click on the template called Payment Gateway / Show on the left navigation pane.



If it's not there already, add the following code before `{% when "braintree_blue" %}`:

```
{% when "stripe" %}
<p><a href="{{ current_account.edit_stripe_billing_address_url }}">Edit
billing address</a></p>

{% if current_account.has_billing_address? %}
  {% stripe_form "Edit Credit Card Details" %}
{% else %}
  <p>After entering billing address, the option to enter credit card will
be enabled.</p>
{% endif %}
```

Finally click Save and Publish. Your users should now be able to pay you using the Stripe gateway.

6.3.1. Note

In order to map your data from Stripe with your data on 3scale, you can use the Stripe field called `metadata.3scale_account_reference` which is composed of `3scale-[PROVIDER_ID]-[DEVELOPER_ACCOUNT_ID]`

